

japi

Reference Manual

Merten Joost

Inhaltsverzeichnis

I	Reference	5
1	Components	7
	Button	7
	Borderpanel	11
	Canvas	16
	Checkbox	21
	CheckmenuItem	24
	Choice	25
	Dialog	28
	Focuslistener	34
	Frame	35
	Helpmenu	42
	Hscrollbar	44
	Graphicbutton	47
	Graphiclabel	50
	Image	53
	KeyListener	56
	Label	57
	Led	60
	List	63
	Menu	67
	MenuItem	69
	Meter	70
	Mouselistener	73
	Panel	74
	PopupMenu	79
	Printer	81
	Progressbar	84
	Radiobutton	87
	Sevensegment	90
	ScrollPane	93
	Textarea	96
	Textfield	100
	Vscrollbar	104
	Window	107

2 Functions	113
additem	113
add	115
alertbox	116
appendtext	117
beep	118
borderpanel	119
button	120
canvas	121
checkbox	122
checkmenuItem	123
choicebox2	124
choicebox3	125
choice	126
cliprect	127
componentlistener	128
connect	129
delete	130
deselect	131
dialog	132
disable	133
dispose	134
drawarc	135
drawcircle	136
drawimagesource	137
drawimage	138
drawline	139
drawoval	140
drawpixel	141
drawpolygon	142
drawpolyline	143
drawrect	144
drawroundrect	145
drawscaledimage	146
drawstring	147
enable	148
filedialog	149
fileselect	150
fillarc	151
fillcircle	152
filloval	153
fillpolygon	154
fillrect	155
fillroundrect	156
focuslistener	157
frame	158
getaction	159

getcolumns	160
getcurpos	161
getdanger	162
getfontascent	163
getfontheight	164
getheight	165
getimagesource	166
getimage	167
getinheight	168
getinsets	169
getinwidth	171
getitemcount	172
getitem	173
getkeychar	174
getkeycode	175
getlayoutid	176
getlength	177
getmousebutton	178
getmousex	179
getmousey	180
getparentid	181
getparent	182
getrows	183
getscaledimage	184
getscreenheight	185
getscreenwidth	186
getselect	187
getselend	188
getselstart	189
getseltext	190
getstate	191
getstringwidth	192
gettext	193
getvalue	194
getviewportheight	195
getviewportwidth	196
getwidth	197
getxpos	198
getypos	199
graphicbutton	200
graphiclabel	201
hasfocus	202
helpmenu	203
hide	204
hscrollbar	205
image	206
insert	207

inserttext	208
isparent	209
isresizable	210
isselect	211
isvisible	212
keylistener	213
label	214
led	215
line	216
list	218
loadimage	219
menubar	220
menuItem	221
menu	222
messagebox	223
meter	224
mouselistener	225
multiplemode	226
nextaction	227
pack	228
panel	229
popupmenu	230
printer	231
print	232
progressbar	233
quit	234
radiobutton	235
radiogroup	236
random	237
releaseall	238
release	239
removeall	240
removeitem	241
remove	242
replacetext	243
saveimage	244
scrollpane	245
selectall	246
select	247
selecttext	248
separator	249
setalign	250
setblockinc	251
setborderlayout	252
setborderpos	253
setcolorbg	254
setcolor	255

setcolumns	256
setcurpos	257
setcursor	258
setdanger	259
setdebug	260
setechochar	261
seteditable	262
setfixlayout	263
setflowfill	264
setflowlayout	265
setfocus	266
setfontname	267
setfontsize	268
setfontstyle	269
setfont	270
setgridlayout	272
sethgap	273
seticon	274
setimage	275
setinsets	276
setmax	277
setmin	278
setnamedcolorbg	279
setnamedcolor	280
setnolayout	281
start	282
setpos	283
setradiogroup	284
setresizable	285
setrows	286
setshortcut	287
setsize	288
setslidesize	289
setstate	290
settext	291
setunitinc	292
setvalue	293
setvgap	294
setxor	295
sevensegment	296
showpopup	297
show	298
sleep	299
start	300
sync	301
textarea	302
textfield	303

translate	304
vscrollbar	305
windowlistener	306
window	307

Teil I

Reference

Kapitel 1

Components

Button

- j_button** *function j_button (obj: integer; label: PChar) : integer;*
Erzeugt einen neuen Button mit dem Textinhalt **label**
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt den Button **obj** in den Container **cont** ein.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an den Button **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled den Button **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
der Button **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled den Button **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an den Button **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Button **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*

	Liefert die totale Höhe des aktuellen Fonts des Button obj in Bildschirmpixeln zurück.
j_getheight	<i>function j_getheight (obj: integer) : integer;</i> Liefert die Höhe des Button obj zurück.
j_getlength	<i>function j_getlength (obj: integer) : integer;</i> Liefert die Länge des Textinhaltes den Button zurück.
j_getparentid	<i>function j_getparentid (obj: integer) : integer;</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>function j_getparent (obj: integer) : integer;</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getstringwidth	<i>function j_getstringwidth (obj: integer; str: PChar) : integer;</i> Liefert die Länge des str des aktuellen Fonts des Button obj in Bildschirmpixeln zurück.
j_gettext	<i>function j_gettext (obj: integer; str: PChar) : PChar;</i> Liefert den Textinhalt des Button .
j_getwidth	<i>function j_getwidth (obj: integer) : integer;</i> Liefert die Breite des Button obj zurück.
j_getxpos	<i>function j_getxpos (obj: integer) : integer;</i> Liefert die aktuelle horizontale Position des Button obj zurück.
j_getypos	<i>function j_getypos (obj: integer) : integer;</i> Liefert die aktuelle vertikale Position des Button obj zurück.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Zeigt den Button obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>function j_isparent (obj: integer; cont: integer) : integer;</i> Liefert true zurück, falls das cont die umgebende Komponente von obj ist, sonst false
j_isvisible	<i>function j_isvisible (obj: integer) : integer;</i> Liefert true zurück, falls obj sichtbar ist, sonst false
j_keylistener	<i>function j_keylistener (obj: integer) : integer;</i> Bindet an den Button obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) : integer;</i> Bindet an den Button obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>function j_popupmenu (obj: integer; label: PChar) : integer;</i>

	Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (obj: integer) ;</i> Druckt den Inhalt des Button .
j_release	<i>procedure j_release (obj: integer) ;</i> Gibt den Button obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>procedure j_setborderpos (obj: integer; pos: integer) ;</i> Positioniert den Button obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i> Setzt den Cursor des Button obj auf cursor .
j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus dem Button obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp des Button obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font des Button obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße des Button obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle des Button obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt den Button obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe des Button obj auf width and height .
j_settext	<i>procedure j_settext (obj: integer; str: PChar) ;</i> Ändert den Textinhalt des Button obj auf str

j_show

```
procedure j_show ( obj: integer ) ;  
Zeigt den Button obj auf dem Bildschirm an.
```

Borderpanel

- j_borderpanel** *function j_borderpanel (obj: integer; type: integer) : integer;*
Erzeugt ein neues Borderpanel und liefert dessen Eventnummer zurück.
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt das Borderpanel **obj** in den Container **cont** ein.
- j_borderpanel** *function j_borderpanel (obj: integer; type: integer) : integer;*
Erzeugt ein neues Borderpanel und liefert dessen Eventnummer zurück.
- j_button** *function j_button (obj: integer; label: PChar) : integer;*
Erzeugt einen neuen Button mit dem Textinhalt **label**
- j_canvas** *function j_canvas (obj: integer; width: integer; height: integer) : integer;*
Erzeugt einen neuen Canvas mit der Breite **width** Pixel und einer Höhe von **height** Pixel.
- j_checkbox** *function j_checkbox (obj: integer; label: PChar) : integer;*
Erzeugt eine Checkbox mit dem Textinhalt **label** und liefert dessen Eventnummer zurück.
- j_choice** *function j_choice (obj: integer) : integer;*
Erzeugt eine neue Choice und gibt deren Eventnummer zurück.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an das Borderpanel **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled das Borderpanel **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
das Borderpanel **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled das Borderpanel **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an das Borderpanel **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Borderpanel **obj** in Bildschirmpixeln zurück.

j_getfontheight	<i>function j_getfontheight (obj: integer) : integer;</i> Liefert die totale Höhe des aktuellen Fonts des Borderpanel obj in Bildschirm-pixeln zurück.
j_getheight	<i>function j_getheight (obj: integer) : integer;</i> Liefert die Höhe des Borderpanel obj zurück.
j_getinheight	<i>function j_getinheight (cont: integer) : integer;</i> Liefert die Höhe des Borderpanel obj des inneren Bereichs zurück.
j_getinsets	<i>function j_getinsets (obj: integer; side: integer) : integer;</i> Liefert die angeforderte Breite des inneren Randes dem Borderpanel zurück.
j_getinwidth	<i>function j_getinwidth (cont: integer) : integer;</i> Liefert die Breite des Borderpanel obj des inneren Bereichs zurück.
j_getlayoutid	<i>function j_getlayoutid (obj: integer) : integer;</i> Liefert die Eventnummer des Layoutmanagers des Containers obj zurück.
j_getparentid	<i>function j_getparentid (obj: integer) : integer;</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>function j_getparent (obj: integer) : integer;</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getstringwidth	<i>function j_getstringwidth (obj: integer; str: PChar) : integer;</i> Liefert die Länge des str des aktuellen Fonts des Borderpanel obj in Bild-schirmpixeln zurück.
j_getwidth	<i>function j_getwidth (obj: integer) : integer;</i> Liefert die Breite des Borderpanel obj zurück.
j_getxpos	<i>function j_getxpos (obj: integer) : integer;</i> Liefert die aktuelle horizontale Position des Borderpanel obj zurück.
j_getypos	<i>function j_getypos (obj: integer) : integer;</i> Liefert die aktuelle vertikale Position des Borderpanel obj zurück.
j_graphicbutton	<i>function j_graphicbutton (obj: integer; filename: PChar) : integer;</i> Erzeugt einen neuen Graphicbutton mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_graphiclabel	<i>function j_graphiclabel (obj: integer; str: PChar) : integer;</i> Erzeugt ein neues Graphiclabel mit dem Bildinhalt des Files, das durch filena-me spezifiziert wird.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Zeigt das Borderpanel obj nicht mehr auf dem Bildschirm an.
j_hscrollbar	<i>function j_hscrollbar (obj: integer) : integer;</i>

Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.

- j_isparent** *function j_isparent (obj: integer; cont: integer) : integer;*
Liefert true zurück, falls das **cont** die umgebende Komponente von **obj** ist, sonst false
- j_isvisible** *function j_isvisible (obj: integer) : integer;*
Liefert true zurück, falls **obj** sichtbar ist, sonst false
- j_keylistener** *function j_keylistener (obj: integer) : integer;*
Bindet an das Borderpanel **obj** einen neuen Keylistener, und liefert dessen Eventnummer zurück.
- j_label** *function j_label (obj: integer; label: PChar) : integer;*
Erzeugt ein neues Label Element mit dem Textinhalt **label**
- j_led** *function j_led (obj: integer; style: integer; color: integer) : integer;*
Erzeugt eine LED der Farbe **color** und der Form **style**
- j_line** *function j_line (obj: integer; orient: integer; style: integer; length: integer) : integer;*
Erzeugt ein neues Linienelement und liefert dessen Eventnummer zurück
- j_list** *function j_list (obj: integer; rows: integer) : integer;*
Erzeugt eine neue List und gibt deren Eventnummer zurück.
- j_meter** *function j_meter (obj: integer; title: PChar) : integer;*
Erzeugt ein analoges Anzeigeinstrument mit der Beschriftung **title**.
- j_mouselistener** *function j_mouselistener (obj: integer; kind: integer) : integer;*
Bindet an das Borderpanel **obj** einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_pack** *procedure j_pack (obj: integer) ;*
Veranlaßt den Layoutmanager von **obj** die minimale Größe zu berechnen.
- j_panel** *function j_panel (obj: integer) : integer;*
Erzeugt ein neues panel und liefert dessen Eventnummer zurück.
- j_popupmenu** *function j_popupmenu (obj: integer; label: PChar) : integer;*
Erzeugt ein neues Popupmenu mit dem Inhalt **label** und liefert dessen Eventnummer zurück.
- j_print** *procedure j_print (obj: integer) ;*
Druckt den Inhalt des Borderpanel .
- j_progressbar** *function j_progressbar (obj: integer; orient: integer) : integer;*
Erzeugt eine Progress Bar mit der Ausrichtung **orient**.
- j_radiogroup** *function j_radiogroup (obj: integer) : integer;*
Erzeugt eine neue Radiogroup und liefert dessen Eventnummer zurück.

j_releaseall	<i>procedure j_releaseall (obj: integer) ;</i> Gibt alle Elemente aus dem Borderpanel obj frei.
j_release	<i>procedure j_release (obj: integer) ;</i> Gibt das Borderpanel obj aus dem umgebenden Container Element frei.
j_scrollpane	<i>function j_scrollpane (obj: integer) : integer;</i> Erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück.
j_setalign	<i>procedure j_setalign (obj: integer; align: integer) ;</i> Setzt die Ausrichtung innerhalb des Borderpanel obj auf align . Ein Flowlayout Manager wird vorausgesetzt.
j_setborderlayout	<i>procedure j_setborderlayout (obj: integer) ;</i> Integriert einen Borderlayout Manager in das Borderpanel obj .
j_setborderpos	<i>procedure j_setborderpos (obj: integer; pos: integer) ;</i> Positioniert das Borderpanel obj an die angegebene Position pos . Im umgebenden Container muß einen Borderlayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i> Setzt den Cursor des Borderpanel obj auf cursor .
j_setfixlayout	<i>procedure j_setfixlayout (obj: integer) ;</i> Integriert einen Fixlayout Manager in das Borderpanel obj (voreingestellte Layoutmanager).
j_setflowfill	<i>procedure j_setflowfill (obj: integer; bool: boolean) ;</i> Setzt die Höhe (Breite) aller enthaltenen Komponenten auf die Höhe (Breite) von dem Borderpanel obj . Ein Flowlayout Manager wird vorausgesetzt.
j_setflowlayout	<i>procedure j_setflowlayout (obj: integer; align: integer) ;</i> Integriert einen Flowlayout Manager in das Borderpanel obj mit der Ausrichtung align .
j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus dem Borderpanel obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp des Borderpanel obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font des Borderpanel obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße des Borderpanel obj auf size .

j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle des Borderpanel obj auf style .
j_setgridlayout	<i>procedure j_setgridlayout (obj: integer; row: integer; col: integer) ;</i> Integriert einen Gridlayout Manager in das Borderpanel obj mit row Reihen und col Spalten.
j_sethgap	<i>procedure j_sethgap (obj: integer; hgap: integer) ;</i> Setzt den horizontalen Abstand innerhalb des Borderpanel obj auf hgap Pixel.
j_setinsets	<i>procedure j_setinsets (obj: integer; top: integer; bottom: integer; left: integer; right: integer) ;</i> Setzt den inneren Rahmen auf die angegebenen Größen.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setnolayout	<i>procedure j_setnolayout (obj: integer) ;</i> Entfernt den aktuellen Layout Manager aus dem Borderpanel obj .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt das Borderpanel obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe des Borderpanel obj auf width and height .
j_setvgap	<i>procedure j_setvgap (obj: integer; vgap: integer) ;</i> Setzt den vertikalen Abstand innerhalb des Borderpanel obj auf vgap Pixel.
j_sevensegment	<i>function j_sevensegment (obj: integer; color: integer) : integer;</i> Erzeugt eine Siebensegment Anzeige der Farbe color .
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt das Borderpanel obj auf dem Bildschirm an.
j_textarea	<i>function j_textarea (obj: integer; rows: integer; columns: integer) : integer;</i> Erzeugt eine neue Textarea und gibt dessen Eventnummer zurück.
j_textfield	<i>function j_textfield (obj: integer; columns: integer) : integer;</i> Erzeugt eine neues Textfield und gibt dessen Eventnummer zurück.
j_vscrollbar	<i>function j_vscrollbar (obj: integer) : integer;</i> Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.

Canvas

- j_canvas** *function j_canvas (obj: integer; width: integer; height: integer) : integer;*
Erzeugt einen neuen Canvas mit der Breite **width** Pixel und einer Höhe von **height** Pixel.
- j_add** *procedure j_add (obj: integer; cont: integer);*
Fügt den Canvas **obj** in den Container **cont** ein.
- j_cliprect** *procedure j_cliprect (obj: integer; x: integer; y: integer; width: integer; height: integer);*
Setzt den Clipping Bereich auf das spezifizierte Rechteck (**x**, **y**, **width**, **height**).
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an den Canvas **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer);*
disabled den Canvas **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer);*
der Canvas **obj** wird mit allen Recourcen gelöscht.
- j_drawarc** *procedure j_drawarc (obj: integer; x: integer; y: integer; rx: integer; ry: integer; arc1: integer; arc2: integer);*
Zeichnet einen Kreisbogen mit dem Mittelpunkt (**x**, **y**) und den Hauptradien **rx** und **ry**. Die Parameter **arc1** und **arc2** bestimmen den Start- bzw. den Endwinkel des Kreisbogens.
- j_drawcircle** *procedure j_drawcircle (obj: integer; x: integer; y: integer; r: integer);*
Zeichnet einen Kreis mit dem Mittelpunkt (**x,y**) und dem Radius **r**.
- j_drawimage** *procedure j_drawimage (obj: integer; image: integer; x: integer; y: integer);*
Kopiert das Image mit der Eventnummer **image** an die Position (**x**, **y**).
- j_drawimagesource** *procedure j_drawimagesource (obj: integer; x: integer; y: integer; w: integer; h: integer; r: var integer; g: var integer; b: var integer);*
Die Funktion stellt ein Bild an der Position (**x,y**) mit der Breite **w** und der Höhe **h** dar. Die Bilddaten werden als rot-, grün- und blau- Kanal in den Arrays **r,g,b** übergeben.
- j_drawline** *procedure j_drawline (obj: integer; x1: integer; y1: integer; x2: integer; y2: integer);*
Zeichnet eine Linie in der momentan gesetzten Vordergrundfarbe von (**x1,y1**) nach (**x2,y2**)

- j_drawoval** *procedure j_drawoval (obj: integer; x: integer; y: integer; rx: integer; ry: integer) ;*
Zeichnet einen Oval mit dem Mittelpunkt **(x,y)** und den Hauptradien **rx** und **ry**.
- j_drawpixel** *procedure j_drawpixel (obj: integer; x: integer; y: integer) ;*
Zeichnet einen Pixel in der momentan gesetzten Vordergrundfarbe an die Position **(x,y)**.
- j_drawpolygon** *procedure j_drawpolygon (obj: integer; len: integer; x: var integer; y: var integer) ;*
Zeichnet ein Polygon durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.
- j_drawpolyline** *procedure j_drawpolyline (obj: integer; len: integer; x: var integer; y: var integer) ;*
Zeichnet eine Polylinie durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.
- j_drawrect** *procedure j_drawrect (obj: integer; x: integer; y: integer; width: integer; height: integer) ;*
Zeichnet ein Rechteck an die Position **(x,y)** mit der Breite **width** und der Höhe **height**.
- j_drawroundrect** *procedure j_drawroundrect (obj: integer; x: integer; y: integer; width: integer; height: integer; arcx: integer; arcy: integer) ;*
Zeichnet ein Rechteck mit abgerundeten Ecken an die Position **(x,y)** mit der Breite **width** und der Höhe **height**. Die Radien der Ecken werden durch **arcx** und **arcy** bestimmt.
- j_drawscaledimage** *procedure j_drawscaledimage (obj: integer; image: integer; sx: integer; sy: integer; sw: integer; sh: integer; tx: integer; ty: integer; tw: integer; th: integer) ;*
Kopiert einen Ausschnitt des Bildes **image**, der durch die Parameter **(sx, sy, sw, sh)** bestimmt wird an die Position **(tx, ty)** und scaliert den Ausschnitt auf die Breite **tw** und die Höhe **th**
- j_drawstring** *procedure j_drawstring (obj: integer; x: integer; y: integer; str: PChar) ;*
Zeichnet den String **str** an die durch die Parameter **(x,y)** festgelegte Koordinate.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled den Canvas **obj**.
- j_fillarc** *procedure j_fillarc (obj: integer; x: integer; y: integer; rx: integer; ry: integer; arc1: integer; arc2: integer) ;*
Zeichnet einen gefüllten Kreisbogen mit dem Mittelpunkt **(x,y)** und den Hauptradien **rx** und **ry**. Die Parameter **arc1** und **arc2** bestimmen den Start- bzw. den Endwinkel des Kreisbogens.
- j_fillcircle** *procedure j_fillcircle (obj: integer; x: integer; y: integer; r: integer) ;*
Zeichnet einen gefülltes Kreis mit dem Mittelpunkt **(x,y)** und dem Radius **r**.

- j_filloval** *procedure j_filloval (obj: integer; x: integer; y: integer; rx: integer; ry: integer) ;*
Zeichnet einen gefüllten Oval mit dem Mittelpunkt **(x,y)** und den Hauptradien **rx** und **ry**.
- j_fillpolygon** *procedure j_fillpolygon (obj: integer; len: integer; x: var integer; y: var integer) ;*
Zeichnet ein gefülltes Polygon durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.
- j_fillrect** *procedure j_fillrect (obj: integer; x: integer; y: integer; width: integer; height: integer) ;*
Zeichnet ein gefülltes Rechteck an die Position **(x,y)** mit der Breite **width** und der Höhe **height**.
- j_fillroundrect** *procedure j_fillroundrect (obj: integer; x: integer; y: integer; width: integer; height: integer; arcx: integer; arcy: integer) ;*
Zeichnet ein gefülltes Rechteck mit abgerundeten Ecken an die Position **(x,y)**. Das Rechteck hat eine Breite von **width** und eine Höhe von **height** Pixeln. Die Form der runden Ecken wird durch die Parameter **arcx** und **arcy** bestimmt.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an den Canvas **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Canvas **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*
Liefert die totale Höhe des aktuellen Fonts des Canvas **obj** in Bildschirmpixeln zurück.
- j_getheight** *function j_getheight (obj: integer) : integer;*
Liefert die Höhe des Canvas **obj** zurück.
- j_getimage** *function j_getimage (obj: integer) : integer;*
Kopiert den Inhalt dem Canvas **obj** in ein Image und liefert dessen Eventnummer zurück.
- j_getimagesource** *function j_getimagesource (obj: integer; x: integer; y: integer; w: integer; h: integer; r: var integer; g: var integer; b: var integer) : integer;*
Die Funktion liefert einen Ausschnitt vom Punkt **(x, y)** der Breite **w** und der Höhe **h** zurück. Der Bildinhalt wird als rot-, grün- und blau- Kanal in den Arrays **r,g,b** abgelegt.
- j_getparentid** *function j_getparentid (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird **-1** zurückgegeben.
- j_getparent** *function j_getparent (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird **-1** zurückgegeben.

j_getscaledimage	<i>function j_getscaledimage (obj: integer; x: integer; y: integer; sw: integer; sh: integer; tw: integer; th: integer) : integer;</i> Kopiert den Inhalt des Ausschnitts von Positio (x , y) it der Breite sw und der höhe sh in ein Image und liefert dessen Eventnummer zurück. Das Image wird dabei auf eine Breite von tw und einer Höhe von th scaliert.
j_getstringwidth	<i>function j_getstringwidth (obj: integer; str: PChar) : integer;</i> Liefert die Länge des str des aktuellen Fonts des Canvas obj in Bildschirmpixeln zurück.
j_getwidth	<i>function j_getwidth (obj: integer) : integer;</i> Liefert die Breite des Canvas obj zurück.
j_getxpos	<i>function j_getxpos (obj: integer) : integer;</i> Liefert die aktuelle horizontale Position des Canvas obj zurück.
j_getypos	<i>function j_getypos (obj: integer) : integer;</i> Liefert die aktuelle vertikale Position des Canvas obj zurück.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Zeigt den Canvas obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>function j_isparent (obj: integer; cont: integer) : integer;</i> Liefert true zurück, falls das cont die umgebende Komponente von obj ist, sonst false
j_isvisible	<i>function j_isvisible (obj: integer) : integer;</i> Liefert true zurück, falls obj sichtbar ist, sonst false
j_keylistener	<i>function j_keylistener (obj: integer) : integer;</i> Bindet an den Canvas obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) : integer;</i> Bindet an den Canvas obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>function j_popupmenu (obj: integer; label: PChar) : integer;</i> Erzeugt ein neues Popuptmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (obj: integer) ;</i> Druckt den Inhalt des Canvas .
j_release	<i>procedure j_release (obj: integer) ;</i> Gibt den Canvas obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>procedure j_setborderpos (obj: integer; pos: integer) ;</i> Positioniert den Canvas obj an die angegebene Position pos . Im umgebenden Container muß einen Borderlayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (obj: integer; r: integer; g: integer; b: integer) ;</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert

j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i> Setzt den Cursor des Canvas obj auf cursor .
j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus dem Canvas obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp des Canvas obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font des Canvas obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße des Canvas obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle des Canvas obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt den Canvas obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe des Canvas obj auf width and height .
j_setxor	<i>procedure j_setxor (obj: integer; bool: boolean) ;</i> Schaltet den Painting Mode auf XOR, falls bool = true ist. Im XOR Modus heben sich zwei gleiche Graphikbefehle auf.
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt den Canvas obj auf dem Bildschirm an.
j_translate	<i>procedure j_translate (obj: integer; x: integer; y: integer) ;</i> Setzt den Ursprung des Koordinatensystems auf (x, y).

Checkbox

- j_checkbox** *function j_checkbox (obj: integer; label: PChar) : integer;*
Erzeugt eine Checkbox mit dem Textinhalt **label** und liefert dessen Eventnummer zurück.
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt die Checkbox **obj** in den Container **cont** ein.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an die Checkbox **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled die Checkbox **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
die Checkbox **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled die Checkbox **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an die Checkbox **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der Checkbox **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*
Liefert die totale Höhe des aktuellen Fonts der Checkbox **obj** in Bildschirmpixeln zurück.
- j_getheight** *function j_getheight (obj: integer) : integer;*
Liefert die Höhe der Checkbox **obj** zurück.
- j_getparentid** *function j_getparentid (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *function j_getparent (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstate** *function j_getstate (obj: integer) : integer;*
Liefert true , falls die Checkbox selektiert ist, sonst false .

j_getstringwidth	<i>function j_getstringwidth (obj: integer; str: PChar) : integer;</i> Liefert die Länge des str des aktuellen Fonts der Checkbox obj in Bildschirm-pixeln zurück.
j_gettext	<i>function j_gettext (obj: integer; str: PChar) : PChar;</i> Liefert den Textinhalt der Checkbox .
j_getwidth	<i>function j_getwidth (obj: integer) : integer;</i> Liefert die Breite der Checkbox obj zurück.
j_getxpos	<i>function j_getxpos (obj: integer) : integer;</i> Liefert die aktuelle horizontale Position der Checkbox obj zurück.
j_getypos	<i>function j_getypos (obj: integer) : integer;</i> Liefert die aktuelle vertikale Position der Checkbox obj zurück.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Zeigt die Checkbox obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>function j_isparent (obj: integer; cont: integer) : integer;</i> Liefert true zurück, falls das cont die umgebende Komponente von obj ist, sonst false
j_isvisible	<i>function j_isvisible (obj: integer) : integer;</i> Liefert true zurück, falls obj sichtbar ist, sonst false
j_keylistener	<i>function j_keylistener (obj: integer) : integer;</i> Bindet an die Checkbox obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) : integer;</i> Bindet an die Checkbox obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>function j_popupmenu (obj: integer; label: PChar) : integer;</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (obj: integer) ;</i> Druckt den Inhalt der Checkbox .
j_release	<i>procedure j_release (obj: integer) ;</i> Gibt die Checkbox obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>procedure j_setborderpos (obj: integer; pos: integer) ;</i> Positioniert die Checkbox obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;</i>

	Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i> Setzt den Cursor der Checkbox obj auf cursor .
j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus der Checkbox obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp der Checkbox obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font der Checkbox obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße der Checkbox obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle der Checkbox obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt die Checkbox obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe der Checkbox obj auf width and height .
j_setstate	<i>procedure j_setstate (obj: integer; bool: boolean) ;</i> Ist bool true so wird die Checkbox selectiert.
j_settext	<i>procedure j_settext (obj: integer; str: PChar) ;</i> Ändert den Textinhalt der Checkbox obj auf str
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt die Checkbox obj auf dem Bildschirm an.

Checkmenüitem

j_checkmenüitem	<p><i>function j_checkmenüitem (obj: integer; label: PChar) : integer;</i> Erzeugt ein Checkmenüitem mit der Beschriftung label und liefert dessen Eventnummer zurück.</p>
j_disable	<p><i>procedure j_disable (obj: integer) ;</i> disabled das Checkmenüitem obj, sodaß diese auf keine Benutzereingaben mehr reagiert.</p>
j_dispose	<p><i>procedure j_dispose (obj: integer) ;</i> das Checkmenüitem obj wird mit allen Recourcen gelöscht.</p>
j_enable	<p><i>procedure j_enable (obj: integer) ;</i> enabled das Checkmenüitem obj.</p>
j_getlength	<p><i>function j_getlength (obj: integer) : integer;</i> Liefert die Länge des Textinhaltes des Checkmenüitem zurück.</p>
j_getstate	<p><i>function j_getstate (obj: integer) : integer;</i> Liefert true , falls das Checkmenüitem selektiert ist, sonst false .</p>
j_gettext	<p><i>function j_gettext (obj: integer; str: PChar) : PChar;</i> Liefert den Textinhalt des Checkmenüitem .</p>
j_setfontname	<p><i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp des Checkmenüitem obj auf name.</p>
j_setfont	<p><i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font des Checkmenüitem obj entsprechend der Parameterliste.</p>
j_setfontsize	<p><i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße des Checkmenüitem obj auf size.</p>
j_setfontstyle	<p><i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle des Checkmenüitem obj auf style.</p>
j_setshortcut	<p><i>procedure j_setshortcut (obj: integer; chr: char) ;</i> Bestimmt das Zeichen chr als Shortcut für das Checkmenüitem .</p>
j_setstate	<p><i>procedure j_setstate (obj: integer; bool: boolean) ;</i> Ist bool true so wird das Checkmenüitem selectiert.</p>
j_settext	<p><i>procedure j_settext (obj: integer; str: PChar) ;</i> Ändert den Textinhalt des Checkmenüitem obj auf str</p>

Choice

- j_choice** *function j_choice (obj: integer) : integer;*
Erzeugt eine neue Choice und gibt deren Eventnummer zurück.
- j_additem** *procedure j_additem (obj: integer; str: PChar) ;*
Hängt ein neues Item mit dem Inhalt **str** an die Auswahlliste an.
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt die Choice **obj** in den Container **cont** ein.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an die Choice **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled die Choice **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
die Choice **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled die Choice **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an die Choice **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der Choice **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*
Liefert die totale Höhe des aktuellen Fonts der Choice **obj** in Bildschirmpixeln zurück.
- j_getheight** *function j_getheight (obj: integer) : integer;*
Liefert die Höhe der Choice **obj** zurück.
- j_getitemcount** *function j_getitemcount (obj: integer) : integer;*
Liefert die Anzahl der Items der Choice **obj** zurück.
- j_getitem** *function j_getitem (obj: integer; item: integer; str: PChar) : PChar;*
liefert den Inhalt des Items mit dem Index **item** zurück.
- j_getparentid** *function j_getparentid (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird **-1** zurückgegeben.

j_getparent	<i>function j_getparent (obj: integer) : integer;</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getselect	<i>function j_getselect (obj: integer) : integer;</i> Liefert den Index des selectierten Items der Choice .
j_getstringwidth	<i>function j_getstringwidth (obj: integer; str: PChar) : integer;</i> Liefert die Länge des str des aktuellen Fonts der Choice obj in Bildschirmpixeln zurück.
j_getwidth	<i>function j_getwidth (obj: integer) : integer;</i> Liefert die Breite der Choice obj zurück.
j_getxpos	<i>function j_getxpos (obj: integer) : integer;</i> Liefert die aktuelle horizontale Position der Choice obj zurück.
j_getypos	<i>function j_getypos (obj: integer) : integer;</i> Liefert die aktuelle vertikale Position der Choice obj zurück.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Zeigt die Choice obj nicht mehr auf dem Bildschirm an.
j_insert	<i>function j_insert (obj: integer; pos: integer; label: PChar) : integer;</i> fügt ein neues Item in an der Position pos ein. Das Item trägt den Inhalt label .
j_isparent	<i>function j_isparent (obj: integer; cont: integer) : integer;</i> Liefert true zurück, falls das cont die umgebende Komponente von obj ist, sonst false
j_isvisible	<i>function j_isvisible (obj: integer) : integer;</i> Liefert true zurück, falls obj sichtbar ist, sonst false
j_keylistener	<i>function j_keylistener (obj: integer) : integer;</i> Bindet an die Choice obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) : integer;</i> Bindet an die Choice obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>function j_popupmenu (obj: integer; label: PChar) : integer;</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (obj: integer) ;</i> Druckt den Inhalt der Choice .
j_release	<i>procedure j_release (obj: integer) ;</i> Gibt die Choice obj aus dem umgebenden Container Element frei.
j_removeall	<i>function j_removeall (obj: integer) : integer;</i> entfernt alle Items aus der Choice .

j_removeitem	<i>function j_removeitem (obj: integer; item: PChar) : integer;</i> entfernt den ersten Eintrag item aus der Choice .
j_remove	<i>function j_remove (obj: integer; item: integer) : integer;</i> entfernt das Item mit dem Index item aus der Choice .
j_select	<i>function j_select (obj: integer; item: integer) : integer;</i> selectiert das Item mit dem Index item .
j_setborderpos	<i>procedure j_setborderpos (obj: integer; pos: integer) ;</i> Positioniert die Choice obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i> Setzt den Cursor der Choice obj auf cursor .
j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus der Choice obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp der Choice obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font der Choice obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße der Choice obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle der Choice obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt die Choice obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe der Choice obj auf width and height .
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt die Choice obj auf dem Bildschirm an.

Dialog

- j_dialog** *function j_dialog (obj: integer; label: PChar) : integer;*
Erzeugt ein neues Dialogwindow mit dem Titel **label** und liefert dessen Eventnummer zurück.
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt den Dialog **obj** in den Container **cont** ein.
- j_borderpanel** *function j_borderpanel (obj: integer; type: integer) : integer;*
Erzeugt ein neues Borderpanel und liefert dessen Eventnummer zurück.
- j_button** *function j_button (obj: integer; label: PChar) : integer;*
Erzeugt einen neuen Button mit dem Textinhalt **label**
- j_canvas** *function j_canvas (obj: integer; width: integer; height: integer) : integer;*
Erzeugt einen neuen Canvas mit der Breite **width** Pixel und einer Höhe von **height** Pixel.
- j_checkbox** *function j_checkbox (obj: integer; label: PChar) : integer;*
Erzeugt eine Checkbox mit dem Textinhalt **label** und liefert dessen Eventnummer zurück.
- j_choice** *function j_choice (obj: integer) : integer;*
Erzeugt eine neue Choice und gibt deren Eventnummer zurück.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an den Dialog **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled den Dialog **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
der Dialog **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled den Dialog **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an den Dialog **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Dialog **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*

	Liefert die totale Höhe des aktuellen Fonts des Dialog obj in Bildschirmpixeln zurück.
j_getheight	<i>function j_getheight (obj: integer) : integer;</i> Liefert die Höhe des Dialog obj zurück.
j_getinheight	<i>function j_getinheight (cont: integer) : integer;</i> Liefert die Höhe des Dialog obj des inneren Bereichs zurück.
j_getinsets	<i>function j_getinsets (obj: integer; side: integer) : integer;</i> Liefert die angeforderte Breite des inneren Randes dem Dialog zurück.
j_getinwidth	<i>function j_getinwidth (cont: integer) : integer;</i> Liefert die Breite des Dialog obj des inneren Bereichs zurück.
j_getlayoutid	<i>function j_getlayoutid (obj: integer) : integer;</i> Liefert die Eventnummer des Layoutmanagers des Containers obj zurück.
j_getlength	<i>function j_getlength (obj: integer) : integer;</i> Liefert die Länge des Textinhaltes den Dialog zurück.
j_getparentid	<i>function j_getparentid (obj: integer) : integer;</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>function j_getparent (obj: integer) : integer;</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getstringwidth	<i>function j_getstringwidth (obj: integer; str: PChar) : integer;</i> Liefert die Länge des str des aktuellen Fonts des Dialog obj in Bildschirmpixeln zurück.
j_gettext	<i>function j_gettext (obj: integer; str: PChar) : PChar;</i> Liefert den Textinhalt des Dialog .
j_getwidth	<i>function j_getwidth (obj: integer) : integer;</i> Liefert die Breite des Dialog obj zurück.
j_getxpos	<i>function j_getxpos (obj: integer) : integer;</i> Liefert die aktuelle horizontale Position des Dialog obj zurück.
j_getypos	<i>function j_getypos (obj: integer) : integer;</i> Liefert die aktuelle vertikale Position des Dialog obj zurück.
j_graphicbutton	<i>function j_graphicbutton (obj: integer; filename: PChar) : integer;</i> Erzeugt einen neuen Graphicbutton mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_graphiclabel	<i>function j_graphiclabel (obj: integer; str: PChar) : integer;</i> Erzeugt ein neues Graphiclabel mit dem Bildinhalt des Files, das durch filename spezifiziert wird.

j_hide	<i>procedure j_hide (obj: integer) ;</i> Zeigt den Dialog obj nicht mehr auf dem Bildschirm an.
j_hscrollbar	<i>function j_hscrollbar (obj: integer) : integer;</i> Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.
j_isparent	<i>function j_isparent (obj: integer; cont: integer) : integer;</i> Liefert true zurück, falls das cont die umgebende Komponente von obj ist, sonst false
j_isresizable	<i>function j_isresizable (obj: integer) : boolean;</i> Liefert false zurueck, falls der Dialog eine fixe Größe besitzt, true sonst.
j_isvisible	<i>function j_isvisible (obj: integer) : integer;</i> Liefert true zurück, falls obj sichtbar ist, sonst false
j_keylistener	<i>function j_keylistener (obj: integer) : integer;</i> Bindet an den Dialog obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_label	<i>function j_label (obj: integer; label: PChar) : integer;</i> Erzeugt ein neues Label Element mit dem Textinhalt label
j_led	<i>function j_led (obj: integer; style: integer; color: integer) : integer;</i> Erzeugt eine LED der Farbe color und der Form style
j_line	<i>function j_line (obj: integer; orient: integer; style: integer; length: integer) : integer;</i> Erzeugt ein neues Linienelement und liefert dessen Eventnummer zurück
j_list	<i>function j_list (obj: integer; rows: integer) : integer;</i> Erzeugt eine neue List und gibt deren Eventnummer zurück.
j_meter	<i>function j_meter (obj: integer; title: PChar) : integer;</i> Erzeugt ein analoges Anzeigeelement mit der Beschriftung title .
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) : integer;</i> Bindet an den Dialog obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_pack	<i>procedure j_pack (obj: integer) ;</i> Veranlaßt den Layoutmanager von obj die minimale Größe zu berechnen.
j_panel	<i>function j_panel (obj: integer) : integer;</i> Erzeugt ein neues panel und liefert dessen Eventnummer zurück.
j_popupmenu	<i>function j_popupmenu (obj: integer; label: PChar) : integer;</i> Erzeugt ein neues Popuptextfeld mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (obj: integer) ;</i> Druckt den Inhalt des Dialog .

j_progressbar	<i>function j_progressbar (obj: integer; orient: integer) : integer;</i> Erzeugt eine Progress Bar mit der Ausrichtung orient .
j_radiogroup	<i>function j_radiogroup (obj: integer) : integer;</i> Erzeugt eine neue Radiogroup und liefert dessen Eventnummer zurück.
j_releaseall	<i>procedure j_releaseall (obj: integer) ;</i> Gibt alle Elemente aus dem Dialog obj frei.
j_release	<i>procedure j_release (obj: integer) ;</i> Gibt den Dialog obj aus dem umgebenden Container Element frei.
j_scrollpane	<i>function j_scrollpane (obj: integer) : integer;</i> Erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück.
j_setalign	<i>procedure j_setalign (obj: integer; align: integer) ;</i> Setzt die Ausrichtung innerhalb des Dialog obj auf align . Ein Flowlayout Manager wird vorausgesetzt.
j_setborderlayout	<i>procedure j_setborderlayout (obj: integer) ;</i> Integriert einen BorderLayout Manager in den Dialog obj .
j_setborderpos	<i>procedure j_setborderpos (obj: integer; pos: integer) ;</i> Positioniert den Dialog obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i> Setzt den Cursor des Dialog obj auf cursor .
j_setfixlayout	<i>procedure j_setfixlayout (obj: integer) ;</i> Integriert einen Fixlayout Manager in den Dialog obj (voreingestellte Layoutmanager).
j_setflowfill	<i>procedure j_setflowfill (obj: integer; bool: boolean) ;</i> Setzt die Höhe (Breite) aller enthaltenen Komponenten auf die Höhe (Breite) von dem Dialog obj . Ein Flowlayout Manager wird vorausgesetzt.
j_setflowlayout	<i>procedure j_setflowlayout (obj: integer; align: integer) ;</i> Integriert einen Flowlayout Manager in den Dialog obj mit der Ausrichtung align .
j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus dem Dialog obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp des Dialog obj auf name .

j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font des Dialog obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße des Dialog obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle des Dialog obj auf style .
j_setgridlayout	<i>procedure j_setgridlayout (obj: integer; row: integer; col: integer) ;</i> Integriert einen Gridlayout Manager in den Dialog obj mit row Reihen und col Spalten.
j_sethgap	<i>procedure j_sethgap (obj: integer; hgap: integer) ;</i> Setzt den horizontalen Abstand innerhalb des Dialog obj auf hgap Pixel.
j_setinsets	<i>procedure j_setinsets (obj: integer; top: integer; bottom: integer; left: integer; right: integer) ;</i> Setzt den inneren Rahmen auf die angegebenen Größen.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setnolayout	<i>procedure j_setnolayout (obj: integer) ;</i> Entfernt den aktuellen Layout Manager aus dem Dialog obj .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt den Dialog obj zur angegebenen Koordinate (xpos,ypos) an.
j_setresizable	<i>procedure j_setresizable (obj: integer; resizable: boolean) ;</i> Ist resizable false , kann die Größe des Dialog nicht vom Benutzer verändert werden.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe des Dialog obj auf width and height .
j_settext	<i>procedure j_settext (obj: integer; str: PChar) ;</i> Ändert den Textinhalt des Dialog obj auf str
j_setvgap	<i>procedure j_setvgap (obj: integer; vgap: integer) ;</i> Setzt den vertikalen Abstand innerhalb des Dialog obj auf vgap Pixel.
j_sevensegment	<i>function j_sevensegment (obj: integer; color: integer) : integer;</i> Erzeugt eine Siebensegment Anzeige der Farbe color .
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt den Dialog obj auf dem Bildschirm an.
j_textarea	<i>function j_textarea (obj: integer; rows: integer; columns: integer) : integer;</i>

	Erzeugt eine neue Textarea und gibt dessen Eventnummer zurück.
j_textfield	<i>function j_textfield (obj: integer; columns: integer) : integer;</i> Erzeugt eine neues Textfield und gibt dessen Eventnummer zurück.
j_vscrollbar	<i>function j_vscrollbar (obj: integer) : integer;</i> Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.
j_windowlistener	<i>function j_windowlistener (window: integer; kind: integer) : integer;</i> Bindet an den Dialog window einen neuen Windowlistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.

Focuslistener

j_focuslistener*function j_focuslistener (obj: integer) : integer;*

Bindet an den Focuslistener **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.

j_dispose*procedure j_dispose (obj: integer) ;*

der Focuslistener **obj** wird mit allen Recourcen gelöscht.

j_hasfocus*function j_hasfocus (obj: integer) : integer;*

Liefert true zurück, falls der Focuslistener den Focus besitzt, sonst false .

Frame

- j_frame** *function j_frame (label: PChar) : integer;*
Erzeugt einen Frame mit der Titelzeile **label**.
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt den Frame **obj** in den Container **cont** ein.
- j_alertbox** *procedure j_alertbox (obj: integer; title: PChar; text: PChar; button: PChar) ;*
Öffnet eine Alertbox mit der Überschrift **title**, Textinhalt **text** und einem Button.
- j_borderpanel** *function j_borderpanel (obj: integer; type: integer) : integer;*
Erzeugt ein neues Borderpanel und liefert dessen Eventnummer zurück.
- j_button** *function j_button (obj: integer; label: PChar) : integer;*
Erzeugt einen neuen Button mit dem Textinhalt **label**
- j_canvas** *function j_canvas (obj: integer; width: integer; height: integer) : integer;*
Erzeugt einen neuen Canvas mit der Breite **width** Pixel und einer Höhe von **height** Pixel.
- j_checkbox** *function j_checkbox (obj: integer; label: PChar) : integer;*
Erzeugt eine Checkbox mit dem Textinhalt **label** und liefert dessen Eventnummer zurück.
- j_choicebox2** *procedure j_choicebox2 (obj: integer; title: PChar; text: PChar; button1: PChar; button2: PChar) ;*
Öffnet eine Choicebox mit der Überschrift **title**, Textinhalt **text** und zwei Buttons.
- j_choicebox3** *procedure j_choicebox3 (obj: integer; title: PChar; text: PChar; button1: PChar; button2: PChar; button3: PChar) ;*
Öffnet eine Choicebox mit der Überschrift **title**, Textinhalt **text** und drei Buttons.
- j_choice** *function j_choice (obj: integer) : integer;*
Erzeugt eine neue Choice und gibt deren Eventnummer zurück.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an den Frame **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_dialog** *function j_dialog (obj: integer; label: PChar) : integer;*
Erzeugt ein neues Dialogwindow mit dem Titel **label** und liefert dessen Eventnummer zurück.
- j_disable** *procedure j_disable (obj: integer) ;*

disabled den Frame **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.

- j_dispose** *procedure j_dispose (obj: integer) ;*
der Frame **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled den Frame **obj**.
- j_filedialog** *function j_filedialog (frame: integer; title: PChar; directory: PChar; filename: PChar) : PChar;*
Öffnet eine Filedialogbox in dem angegebenen **directory** mit dem Title **title** und liefert den ausgewählten Dateiname in **filename** zurück.
- j_fileselect** *function j_fileselect (frame: integer; title: PChar; filter: PChar; filename: PChar) : PChar;*
Öffnet eine Fileslectorbox mit dem voreingestellten **filename** und dem Title **title** und liefert den ausgewählten Dateiname in **filename** zurück.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an den Frame **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Frame **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*
Liefert die totale Höhe des aktuellen Fonts des Frame **obj** in Bildschirmpixeln zurück.
- j_getheight** *function j_getheight (obj: integer) : integer;*
Liefert die Höhe des Frame **obj** zurück.
- j_getinheight** *function j_getinheight (cont: integer) : integer;*
Liefert die Höhe des Frame **obj** des inneren Bereichs zurück.
- j_getinsets** *function j_getinsets (obj: integer; side: integer) : integer;*
Liefert die angeforderte Breite des inneren Randes dem Frame zurück.
- j_getinwidth** *function j_getinwidth (cont: integer) : integer;*
Liefert die Breite des Frame **obj** des inneren Bereichs zurück.
- j_getlayoutid** *function j_getlayoutid (obj: integer) : integer;*
Liefert die Eventnummer des Layoutmanagers des Containers **obj** zurück.
- j_getlength** *function j_getlength (obj: integer) : integer;*
Liefert die Länge des Textinhaltes den Frame zurück.
- j_getparentid** *function j_getparentid (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.

j_getparent	<i>function j_getparent (obj: integer) : integer;</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getstringwidth	<i>function j_getstringwidth (obj: integer; str: PChar) : integer;</i> Liefert die Länge des str des aktuellen Fonts des Frame obj in Bildschirmpixeln zurück.
j_gettext	<i>function j_gettext (obj: integer; str: PChar) : PChar;</i> Liefert den Textinhalt des Frame .
j_getwidth	<i>function j_getwidth (obj: integer) : integer;</i> Liefert die Breite des Frame obj zurück.
j_getxpos	<i>function j_getxpos (obj: integer) : integer;</i> Liefert die aktuelle horizontale Position des Frame obj zurück.
j_getypos	<i>function j_getypos (obj: integer) : integer;</i> Liefert die aktuelle vertikale Position des Frame obj zurück.
j_graphicbutton	<i>function j_graphicbutton (obj: integer; filename: PChar) : integer;</i> Erzeugt einen neuen Graphicbutton mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_graphiclabel	<i>function j_graphiclabel (obj: integer; str: PChar) : integer;</i> Erzeugt ein neues Graphiclabel mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Zeigt den Frame obj nicht mehr auf dem Bildschirm an.
j_hscrollbar	<i>function j_hscrollbar (obj: integer) : integer;</i> Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.
j_isparent	<i>function j_isparent (obj: integer; cont: integer) : integer;</i> Liefert true zurück, falls das cont die umgebende Komponente von obj ist, sonst false
j_isresizable	<i>function j_isresizable (obj: integer) : boolean;</i> Liefert false zurueck, falls der Frame eine fixe Größe besitzt, true sonst.
j_isvisible	<i>function j_isvisible (obj: integer) : integer;</i> Liefert true zurück, falls obj sichtbar ist, sonst false
j_keylistener	<i>function j_keylistener (obj: integer) : integer;</i> Bindet an den Frame obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_label	<i>function j_label (obj: integer; label: PChar) : integer;</i> Erzeugt ein neues Label Element mit dem Textinhalt label
j_led	<i>function j_led (obj: integer; style: integer; color: integer) : integer;</i> Erzeugt eine LED der Farbe color und der Form style

j_line	<i>function j_line (obj: integer; orient: integer; style: integer; length: integer) : integer;</i> Erzeugt ein neues Linienelement und liefert dessen Eventnummer zurück
j_list	<i>function j_list (obj: integer; rows: integer) : integer;</i> Erzeugt eine neue List und gibt deren Eventnummer zurück.
j_menubar	<i>function j_menubar (obj: integer) : integer;</i> Erzeugt eine neue Menubar in obj und liefert dessen Eventnummer zurück.
j_messagebox	<i>procedure j_messagebox (obj: integer; title: PChar; text: PChar) ;</i> Öffnet eine MessageBox mit der Überschrift title und dem Textinhalt text
j_meter	<i>function j_meter (obj: integer; title: PChar) : integer;</i> Erzeugt ein analoges Anzeigeelement mit der Beschriftung title .
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) : integer;</i> Bindet an den Frame obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_pack	<i>procedure j_pack (obj: integer) ;</i> Veranlaßt den Layoutmanager von obj die minimale Größe zu berechnen.
j_panel	<i>function j_panel (obj: integer) : integer;</i> Erzeugt ein neues panel und liefert dessen Eventnummer zurück.
j_popupmenu	<i>function j_popupmenu (obj: integer; label: PChar) : integer;</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_printer	<i>function j_printer (frame: integer) : integer;</i> Erzeugt ein neues Objekt, das eine Papierseite des Druckes repräsentiert.
j_print	<i>procedure j_print (obj: integer) ;</i> Druckt den Inhalt des Frame .
j_progressbar	<i>function j_progressbar (obj: integer; orient: integer) : integer;</i> Erzeugt eine Progress Bar mit der Ausrichtung orient .
j_radiogroup	<i>function j_radiogroup (obj: integer) : integer;</i> Erzeugt eine neue Radiogroup und liefert dessen Eventnummer zurück.
j_releaseall	<i>procedure j_releaseall (obj: integer) ;</i> Gibt alle Elemente aus dem Frame obj frei.
j_release	<i>procedure j_release (obj: integer) ;</i> Gibt den Frame obj aus dem umgebenden Container Element frei.
j_scrollpane	<i>function j_scrollpane (obj: integer) : integer;</i> Erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück.
j_setalign	<i>procedure j_setalign (obj: integer; align: integer) ;</i>

	Setzt die Ausrichtung innerhalb des Frame obj auf align . Ein Flowlayout Manager wird vorausgesetzt.
j_setborderlayout	<i>procedure j_setborderlayout (obj: integer) ;</i> Integriert einen BorderLayout Manager in den Frame obj .
j_setborderpos	<i>procedure j_setborderpos (obj: integer; pos: integer) ;</i> Positioniert den Frame obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (obj: integer; r: integer; g.: integer; b: integer) ;</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g.: integer; b: integer) ;</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i> Setzt den Cursor des Frame obj auf cursor .
j_setfixlayout	<i>procedure j_setfixlayout (obj: integer) ;</i> Integriert einen Fixlayout Manager in den Frame obj (voreingestellte Layoutmanager).
j_setflowfill	<i>procedure j_setflowfill (obj: integer; bool: boolean) ;</i> Setzt die Höhe (Breite) aller enthaltenen Komponenten auf die Höhe (Breite) von dem Frame obj . Ein Flowlayout Manager wird vorausgesetzt.
j_setflowlayout	<i>procedure j_setflowlayout (obj: integer; align: integer) ;</i> Integriert einen Flowlayout Manager in den Frame obj mit der Ausrichtung align .
j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus dem Frame obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp des Frame obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font des Frame obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße des Frame obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle des Frame obj auf style .
j_setgridlayout	<i>procedure j_setgridlayout (obj: integer; row: integer; col: integer) ;</i> Integriert einen Gridlayout Manager in den Frame obj mit row Reihen und col Spalten.
j_sethgap	<i>procedure j_sethgap (obj: integer; hgap: integer) ;</i> Setzt den horizontalen Abstand innerhalb des Frame obj auf hgap Pixel.

j_seticon	<i>procedure j_seticon (frame: integer; icon: integer) ;</i> Ordnet dem frame das Icon icon zu. Eine iconisierte Darstellung wird nicht von allen Plattformen unterstützt.
j_setinsets	<i>procedure j_setinsets (obj: integer; top: integer; bottom: integer; left: integer; right: integer) ;</i> Setzt den inneren Rahmen auf die angegebenen Größen.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setnolayout	<i>procedure j_setnolayout (obj: integer) ;</i> Entfernt den aktuellen Layout Manager aus dem Frame obj .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt den Frame obj zur angegebenen Koordinate (xpos,ypos) an.
j_setresizable	<i>procedure j_setresizable (obj: integer; resizable: boolean) ;</i> Ist resizable false , kann die Größe des Frame nicht vom Benutzer verändert werden.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe des Frame obj auf width and height .
j_settext	<i>procedure j_settext (obj: integer; str: PChar) ;</i> Ändert den Textinhalt des Frame obj auf str
j_setvgap	<i>procedure j_setvgap (obj: integer; vgap: integer) ;</i> Setzt den vertikalen Abstand innerhalb des Frame obj auf vgap Pixel.
j_sevensegment	<i>function j_sevensegment (obj: integer; color: integer) : integer;</i> Erzeugt eine Siebensegment Anzeige der Farbe color .
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt den Frame obj auf dem Bildschirm an.
j_textarea	<i>function j_textarea (obj: integer; rows: integer; columns: integer) : integer;</i> Erzeugt eine neue Textarea und gibt dessen Eventnummer zurück.
j_textfield	<i>function j_textfield (obj: integer; columns: integer) : integer;</i> Erzeugt eine neues Textfeld und gibt dessen Eventnummer zurück.
j_vscrollbar	<i>function j_vscrollbar (obj: integer) : integer;</i> Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.
j_windowlistener	<i>function j_windowlistener (window: integer; kind: integer) : integer;</i> Bindet an den Frame window einen neuen Windowlistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.

j_window*function j_window (obj: integer) : integer;*

Erzeugt ein neues Window und liefert dessen Eventnummer zurück.

Helpmenu

j_helpmenu	<i>function j_helpmenu (obj: integer; label: PChar) : integer;</i> Erzeugt ein Hilfe-Menueintrag mit der Beschriftung label und liefert dessen Eventnummer zurück.
j_checkmenuitem	<i>function j_checkmenuitem (obj: integer; label: PChar) : integer;</i> Erzeugt ein Checkmenuitem mit der Beschriftung label und liefert dessen Eventnummer zurück.
j_disable	<i>procedure j_disable (obj: integer) ;</i> disabled das Helpmenu obj , sodaß diese auf keine Benutzereingaben mehr reagiert.
j_dispose	<i>procedure j_dispose (obj: integer) ;</i> das Helpmenu obj wird mit allen Recourcen gelöscht.
j_enable	<i>procedure j_enable (obj: integer) ;</i> enabled das Helpmenu obj .
j_getlength	<i>function j_getlength (obj: integer) : integer;</i> Liefert die Länge des Textinhaltes das Helpmenu zurück.
j_gettext	<i>function j_gettext (obj: integer; str: PChar) : PChar;</i> Liefert den Textinhalt des Helpmenu .
j_menuitem	<i>function j_menuitem (obj: integer; label: PChar) : integer;</i> Erzeugt ein MenuItem mit der Beschriftung label und liefert dessen Eventnummer zurück.
j_seperator	<i>procedure j_seperator (obj: integer) ;</i> Erzeugt eine Trennlinie in das Helpmenu .
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp des Helpmenu obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font des Helpmenu obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße des Helpmenu obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle des Helpmenu obj auf style .
j_setshortcut	<i>procedure j_setshortcut (obj: integer; chr: char) ;</i> Bestimmt das Zeichen chr als Shortcut für das Helpmenu .
j_settext	<i>procedure j_settext (obj: integer; str: PChar) ;</i>

Ändert den Textinhalt des Helpmenu **obj** auf **str**

Hscrollbar

- j_hscrollbar** *function j_hscrollbar (obj: integer) : integer;*
Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt die Hscrollbar **obj** in den Container **cont** ein.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an die Hscrollbar **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled die Hscrollbar **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
die Hscrollbar **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled die Hscrollbar **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an die Hscrollbar **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der Hscrollbar **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*
Liefert die totale Höhe des aktuellen Fonts der Hscrollbar **obj** in Bildschirmpixeln zurück.
- j_getheight** *function j_getheight (obj: integer) : integer;*
Liefert die Höhe der Hscrollbar **obj** zurück.
- j_getparentid** *function j_getparentid (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *function j_getparent (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstringwidth** *function j_getstringwidth (obj: integer; str: PChar) : integer;*
Liefert die Länge des **str** des aktuellen Fonts der Hscrollbar **obj** in Bildschirmpixeln zurück.

j_getvalue	<i>function j_getvalue (obj: integer) : integer;</i> Liefert den aktuellen Wert der Hscrollbar .
j_getwidth	<i>function j_getwidth (obj: integer) : integer;</i> Liefert die Breite der Hscrollbar obj zurück.
j_getxpos	<i>function j_getxpos (obj: integer) : integer;</i> Liefert die aktuelle horizontale Position der Hscrollbar obj zurück.
j_getypos	<i>function j_getypos (obj: integer) : integer;</i> Liefert die aktuelle vertikale Position der Hscrollbar obj zurück.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Zeigt die Hscrollbar obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>function j_isparent (obj: integer; cont: integer) : integer;</i> Liefert true zurück, falls das cont die umgebende Komponente von obj ist, sonst false
j_isvisible	<i>function j_isvisible (obj: integer) : integer;</i> Liefert true zurück, falls obj sichtbar ist, sonst false
j_keylistener	<i>function j_keylistener (obj: integer) : integer;</i> Bindet an die Hscrollbar obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) : integer;</i> Bindet an die Hscrollbar obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>function j_popupmenu (obj: integer; label: PChar) : integer;</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (obj: integer) ;</i> Druckt den Inhalt der Hscrollbar .
j_release	<i>procedure j_release (obj: integer) ;</i> Gibt die Hscrollbar obj aus dem umgebenden Container Element frei.
j_setblockinc	<i>function j_setblockinc (obj: integer; val: integer) : integer;</i> Stellt die Blockänderung von der Hscrollbar auf den Wert von val ein.
j_setborderpos	<i>procedure j_setborderpos (obj: integer; pos: integer) ;</i> Positioniert die Hscrollbar obj an die angegebene Position pos . Im umgebenden Container muß einen Borderlayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert

j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i> Setzt den Cursor der Hscrollbar obj auf cursor .
j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus der Hscrollbar obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp der Hscrollbar obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font der Hscrollbar obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße der Hscrollbar obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle der Hscrollbar obj auf style .
j_setmax	<i>function j_setmax (obj: integer; val: integer) : integer;</i> Stellt die obere Grenze der Hscrollbar auf den Wert von val ein.
j_setmin	<i>function j_setmin (obj: integer; val: integer) : integer;</i> Stellt die untere Grenze der Hscrollbar auf den Wert von val ein.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt die Hscrollbar obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe der Hscrollbar obj auf width and height .
j_setslidesize	<i>function j_setslidesize (obj: integer; val: integer) : integer;</i> Setzt die Größe des Schiebers auf den Wert von val .
j_setunitinc	<i>function j_setunitinc (obj: integer; val: integer) : integer;</i> Stellt die minimale Wertänderung von die Hscrollbar auf den Wert von val ein.
j_setvalue	<i>procedure j_setvalue (obj: integer; val: integer) ;</i> Stellt den aktuellen Wert der Hscrollbar obj auf val .
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt die Hscrollbar obj auf dem Bildschirm an.

Graphicbutton

- j_graphicbutton** *function j_graphicbutton (obj: integer; filename: PChar) : integer;*
Erzeugt einen neuen Graphicbutton mit dem Bildinhalt des Files, das durch **filename** spezifiziert wird.
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt den Graphicbutton **obj** in den Container **cont** ein.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an den Graphicbutton **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled den Graphicbutton **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
der Graphicbutton **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled den Graphicbutton **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an den Graphicbutton **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Graphicbutton **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*
Liefert die totale Höhe des aktuellen Fonts des Graphicbutton **obj** in Bildschirmpixeln zurück.
- j_getheight** *function j_getheight (obj: integer) : integer;*
Liefert die Höhe des Graphicbutton **obj** zurück.
- j_getparentid** *function j_getparentid (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *function j_getparent (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstringwidth** *function j_getstringwidth (obj: integer; str: PChar) : integer;*

	Liefert die Länge des str des aktuellen Fonts des Graphicbutton obj in Bildschirmpixeln zurück.
j_getwidth	<i>function j_getwidth (obj: integer) : integer;</i> Liefert die Breite des Graphicbutton obj zurück.
j_getxpos	<i>function j_getxpos (obj: integer) : integer;</i> Liefert die aktuelle horizontale Position des Graphicbutton obj zurück.
j_getypos	<i>function j_getypos (obj: integer) : integer;</i> Liefert die aktuelle vertikale Position des Graphicbutton obj zurück.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Zeigt den Graphicbutton obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>function j_isparent (obj: integer; cont: integer) : integer;</i> Liefert true zurück, falls das cont die umgebende Komponente von obj ist, sonst false
j_isvisible	<i>function j_isvisible (obj: integer) : integer;</i> Liefert true zurück, falls obj sichtbar ist, sonst false
j_keylistener	<i>function j_keylistener (obj: integer) : integer;</i> Bindet an den Graphicbutton obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) : integer;</i> Bindet an den Graphicbutton obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>function j_popupmenu (obj: integer; label: PChar) : integer;</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (obj: integer) ;</i> Druckt den Inhalt des Graphicbutton .
j_release	<i>procedure j_release (obj: integer) ;</i> Gibt den Graphicbutton obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>procedure j_setborderpos (obj: integer; pos: integer) ;</i> Positioniert den Graphicbutton obj an die angegebene Position pos . Im umgebenden Container muß einen Borderlayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i> Setzt den Cursor des Graphicbutton obj auf cursor .

j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus dem Graphicbutton obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp des Graphicbutton obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font des Graphicbutton obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße des Graphicbutton obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle des Graphicbutton obj auf style .
j_setimage	<i>procedure j_setimage (obj: integer; image: integer) ;</i> Ordnet dem obj das Bild image zu.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt den Graphicbutton obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe des Graphicbutton obj auf width and height .
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt den Graphicbutton obj auf dem Bildschirm an.

Graphiclabel

- j_graphiclabel** *function j_graphiclabel (obj: integer; str: PChar) : integer;*
Erzeugt ein neues Graphiclabel mit dem Bildinhalt des Files, das durch **filena-**
me spezifiziert wird.
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt das Graphiclabel **obj** in den Container **cont** ein.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an das Graphiclabel **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled das Graphiclabel **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
das Graphiclabel **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled das Graphiclabel **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an das Graphiclabel **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Graphiclabel **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*
Liefert die totale Höhe des aktuellen Fonts des Graphiclabel **obj** in Bildschirmpixeln zurück.
- j_getheight** *function j_getheight (obj: integer) : integer;*
Liefert die Höhe des Graphiclabel **obj** zurück.
- j_getparentid** *function j_getparentid (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *function j_getparent (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstringwidth** *function j_getstringwidth (obj: integer; str: PChar) : integer;*

Liefert die Länge des **str** des aktuellen Fonts des Graphiclabel **obj** in Bildschirmpixeln zurück.

- j_getwidth** *function j_getwidth (obj: integer) : integer;*
Liefert die Breite des Graphiclabel **obj** zurück.
- j_getxpos** *function j_getxpos (obj: integer) : integer;*
Liefert die aktuelle horizontale Position des Graphiclabel **obj** zurück.
- j_getypos** *function j_getypos (obj: integer) : integer;*
Liefert die aktuelle vertikale Position des Graphiclabel **obj** zurück.
- j_hide** *procedure j_hide (obj: integer) ;*
Zeigt das Graphiclabel **obj** nicht mehr auf dem Bildschirm an.
- j_isparent** *function j_isparent (obj: integer; cont: integer) : integer;*
Liefert true zurück, falls das **cont** die umgebende Komponente von **obj** ist, sonst false
- j_isvisible** *function j_isvisible (obj: integer) : integer;*
Liefert true zurück, falls **obj** sichtbar ist, sonst false
- j_keylistener** *function j_keylistener (obj: integer) : integer;*
Bindet an das Graphiclabel **obj** einen neuen Keylistener, und liefert dessen Eventnummer zurück.
- j_mouselistener** *function j_mouselistener (obj: integer; kind: integer) : integer;*
Bindet an das Graphiclabel **obj** einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_popupmenu** *function j_popupmenu (obj: integer; label: PChar) : integer;*
Erzeugt ein neues Popupmenu mit dem Inhalt **label** und liefert dessen Eventnummer zurück.
- j_print** *procedure j_print (obj: integer) ;*
Druckt den Inhalt des Graphiclabel .
- j_release** *procedure j_release (obj: integer) ;*
Gibt das Graphiclabel **obj** aus dem umgebenden Container Element frei.
- j_setborderpos** *procedure j_setborderpos (obj: integer; pos: integer) ;*
Positioniert das Graphiclabel **obj** an die angegebene Position **pos**. Im umgebenden Container muß einen Borderlayout Manager eingestellt sein.
- j_setcolorbg** *procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;*
Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
- j_setcolor** *procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;*
Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
- j_setcursor** *function j_setcursor (obj: integer; cursor: integer) : integer;*
Setzt den Cursor des Graphiclabel **obj** auf **cursor**.

j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus dem Graphiclabel obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp des Graphiclabel obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font des Graphiclabel obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße des Graphiclabel obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle des Graphiclabel obj auf style .
j_setimage	<i>procedure j_setimage (obj: integer; image: integer) ;</i> Ordnet dem obj das Bild image zu.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt das Graphiclabel obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe des Graphiclabel obj auf width and height .
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt das Graphiclabel obj auf dem Bildschirm an.

Image

- j_image** *function j_image (width: integer; height: integer) : integer;*
Erzeugt einen neues (Memory) Image mit der Breite **width** Pixel und einer Höhe von **height** Pixel.
- j_cliprect** *procedure j_cliprect (obj: integer; x: integer; y: integer; width: integer; height: integer) ;*
Setzt den Clipping Bereich auf das spezifizierte Rechteck (**x**, **y**, **width**, **height**).
- j_dispose** *procedure j_dispose (obj: integer) ;*
das Image **obj** wird mit allen Recourcen gelöscht.
- j_drawarc** *procedure j_drawarc (obj: integer; x: integer; y: integer; rx: integer; ry: integer; arc1: integer; arc2: integer) ;*
Zeichnet einen Kreisbogen mit dem Mittelpunkt (**x**, **y**) und den Hauptradien **rx** und **ry**. Die Parameter **arc1** und **arc2** bestimmen den Start- bzw. den Endwinkel des Kreisbogens.
- j_drawcircle** *procedure j_drawcircle (obj: integer; x: integer; y: integer; r: integer) ;*
Zeichnet einen Kreis mit dem Mittelpunkt (**x,y**) und dem Radius **r**.
- j_drawimage** *procedure j_drawimage (obj: integer; image: integer; x: integer; y: integer) ;*
Kopiert das Image mit der Eventnummer **image** an die Position (**x**, **y**).
- j_drawimagesource** *procedure j_drawimagesource (obj: integer; x: integer; y: integer; w: integer; h: integer; r: var integer; g: var integer; b: var integer) ;*
Die Funktion stellt ein Bild an der Position (**x,y**) mit der Breite **w** und der Höhe **h** dar. Die Bilddaten werden als rot-, grün- und blau- Kanal in den Arrays **r,g,b** übergeben.
- j_drawline** *procedure j_drawline (obj: integer; x1: integer; y1: integer; x2: integer; y2: integer) ;*
Zeichnet eine Linie in der momentan gesetzten Vordergrundfarbe von (**x1,y1**) nach (**x2,y2**)
- j_drawoval** *procedure j_drawoval (obj: integer; x: integer; y: integer; rx: integer; ry: integer) ;*
Zeichnet einen Oval mit dem Mittelpunkt (**x,y**) und den Hauptradien **rx** und **ry**.
- j_drawpixel** *procedure j_drawpixel (obj: integer; x: integer; y: integer) ;*
Zeichnet einen Pixel in der momentan gesetzten Vordergrundfarbe an die Position (**x,y**).
- j_drawpolygon** *procedure j_drawpolygon (obj: integer; len: integer; x: var integer; y: var integer) ;*

- Zeichnet ein Polygon durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.
- j_drawpolyline** *procedure j_drawpolyline (obj: integer; len: integer; x: var integer; y: var integer) ;*
Zeichnet eine Polylinie durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.
- j_drawrect** *procedure j_drawrect (obj: integer; x: integer; y: integer; width: integer; height: integer) ;*
Zeichnet ein Rechteck an die Position (**x,y**) mit der Breite **width** und der Höhe **height**.
- j_drawroundrect** *procedure j_drawroundrect (obj: integer; x: integer; y: integer; width: integer; height: integer; arcx: integer; arcy: integer) ;*
Zeichnet ein Rechteck mit abgerundeten Ecken an die Position (**x,y**) mit der Breite **width** und der Höhe **height**. Die Radien der Ecken werden durch **arcx** und **arcy** bestimmt.
- j_drawscaleddimimage** *procedure j_drawscaleddimimage (obj: integer; image: integer; sx: integer; sy: integer; sw: integer; sh: integer; tx: integer; ty: integer; tw: integer; th: integer) ;*
Kopiert einen Ausschnitt des Bildes **image**, der durch die Parameter (**sx, sy, sw, sh**) bestimmt wird an die Position (**tx, ty**) und scaliert den Ausschnitt auf die Breite **tw** und die Höhe **th**
- j_drawstring** *procedure j_drawstring (obj: integer; x: integer; y: integer; str: PChar) ;*
Zeichnet den String **str** an die durch die Parameter (**x,y**) festgelegte Koordinate.
- j_fillarc** *procedure j_fillarc (obj: integer; x: integer; y: integer; rx: integer; ry: integer; arc1: integer; arc2: integer) ;*
Zeichnet einen gefüllten Kreisbogen mit dem Mittelpunkt (**x,y**) und den Hauptradien **rx** und **ry**. Die Parameter **arc1** und **arc2** bestimmen den Start- bzw. den Endwinkel des Kreisbogens.
- j_fillcircle** *procedure j_fillcircle (obj: integer; x: integer; y: integer; r: integer) ;*
Zeichnet einen gefüllten Kreis mit dem Mittelpunkt (**x,y**) und dem Radius **r**.
- j_filloval** *procedure j_filloval (obj: integer; x: integer; y: integer; rx: integer; ry: integer) ;*
Zeichnet einen gefüllten Oval mit dem Mittelpunkt (**x,y**) und den Hauptradien **rx** und **ry**.
- j_fillpolygon** *procedure j_fillpolygon (obj: integer; len: integer; x: var integer; y: var integer) ;*
Zeichnet ein gefülltes Polygon durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.
- j_fillrect** *procedure j_fillrect (obj: integer; x: integer; y: integer; width: integer; height: integer) ;*
Zeichnet ein gefülltes Rechteck an die Position (**x,y**) mit der Breite **width** und der Höhe **height**.

- j_fillroundrect** *procedure j_fillroundrect (obj: integer; x: integer; y: integer; width: integer; height: integer; arcx: integer; arcy: integer) ;*
 Zeichnet ein gefülltes Rechteck mit abgerundeten Ecken an die Position (**x,y**). Das Rechteck hat eine Breite von **width** und eine Höhe von **height** Pixeln. Die Form der runden Ecken wird durch die Parameter **arcx** und **arcy** bestimmt.
- j_getheight** *function j_getheight (obj: integer) : integer;*
 Liefert die Höhe des Image **obj** zurück.
- j_getimage** *function j_getimage (obj: integer) : integer;*
 Kopiert den Inhalt dem Image **obj** in ein Image und liefert dessen Eventnummer zurück.
- j_getimagesource** *function j_getimagesource (obj: integer; x: integer; y: integer; w: integer; h: integer; r: var integer; g: var integer; b: var integer) : integer;*
 Die Funktion liefert einen Ausschnitt vom Punkt (**x, y**) der Breite **w** und der Höhe **h** zurck. Der Bildinhalt wird als rot-, grün- und blau- Kanal in den Arrays r,g,b abgelegt.
- j_getscaledimage** *function j_getscaledimage (obj: integer; x: integer; y: integer; sw: integer; sh: integer; tw: integer; th: integer) : integer;*
 Kopiert den Inhalt des Ausschnitts von Positio (**x, y**) it der Breite **sw** und der höhe **sh** in ein Image und liefert dessen Eventnummer zurück. Das Image wird dabei auf eine Breite von **tw** und einer Höhe von **th** skaliert.
- j_getwidth** *function j_getwidth (obj: integer) : integer;*
 Liefert die Breite des Image **obj** zurück.
- j_print** *procedure j_print (obj: integer) ;*
 Druckt den Inhalt des Image .
- j_setxor** *procedure j_setxor (obj: integer; bool: boolean) ;*
 Schaltet den Painting Mode auf XOR, falls bool = true ist. Im XOR Modus heben sich zwei gleiche Graphikbefehle auf.
- j_translate** *procedure j_translate (obj: integer; x: integer; y: integer) ;*
 Setzt den Ursprung des Koordinatensystems auf (**x, y**).

KeyListener

- j_keylistener** *function j_keylistener (obj: integer) : integer;*
Bindet an den Keylistener **obj** einen neuen Keylistener, und liefert dessen Eventnummer zurück.
- j_dispose** *procedure j_dispose (obj: integer) ;*
der Keylistener **obj** wird mit allen Recourcen gelöscht.
- j_getkeychar** *function j_getkeychar (obj: integer) : integer;*
Liefert den ASCII Code, der zuletzt gedrückten Taste zurück.
- j_getkeycode** *function j_getkeycode (obj: integer) : integer;*
Liefert den Keycode, der zuletzt gedrückten Taste zurück.

Label

- j_label** *function j_label (obj: integer; label: PChar) : integer;*
Erzeugt ein neues Label Element mit dem Textinhalt **label**
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt das Label **obj** in den Container **cont** ein.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an das Label **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled das Label **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
das Label **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled das Label **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an das Label **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Label **obj** in Bildschirm-pixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*
Liefert die totale Höhe des aktuellen Fonts des Label **obj** in Bildschirmpixeln zurück.
- j_getheight** *function j_getheight (obj: integer) : integer;*
Liefert die Höhe des Label **obj** zurück.
- j_getparentid** *function j_getparentid (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *function j_getparent (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstringwidth** *function j_getstringwidth (obj: integer; str: PChar) : integer;*
Liefert die Länge des **str** des aktuellen Fonts des Label **obj** in Bildschirmpixeln zurück.

j_gettext	<i>function j_gettext (obj: integer; str: PChar) : PChar;</i> Liefert den Textinhalt des Label .
j_getwidth	<i>function j_getwidth (obj: integer) : integer;</i> Liefert die Breite des Label obj zurück.
j_getxpos	<i>function j_getxpos (obj: integer) : integer;</i> Liefert die aktuelle horizontale Position des Label obj zurück.
j_getypos	<i>function j_getypos (obj: integer) : integer;</i> Liefert die aktuelle vertikale Position des Label obj zurück.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Zeigt das Label obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>function j_isparent (obj: integer; cont: integer) : integer;</i> Liefert true zurück, falls das cont die umgebende Komponente von obj ist, sonst false
j_isvisible	<i>function j_isvisible (obj: integer) : integer;</i> Liefert true zurück, falls obj sichtbar ist, sonst false
j_keylistener	<i>function j_keylistener (obj: integer) : integer;</i> Bindet an das Label obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) : integer;</i> Bindet an das Label obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>function j_popupmenu (obj: integer; label: PChar) : integer;</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (obj: integer) ;</i> Druckt den Inhalt des Label .
j_release	<i>procedure j_release (obj: integer) ;</i> Gibt das Label obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>procedure j_setborderpos (obj: integer; pos: integer) ;</i> Positioniert das Label obj an die angegebene Position pos . Im umgebenden Container muß einen Borderlayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i> Setzt den Cursor des Label obj auf cursor .

j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus dem Label obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp des Label obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font des Label obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße des Label obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle des Label obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt das Label obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe des Label obj auf width and height .
j_settext	<i>procedure j_settext (obj: integer; str: PChar) ;</i> Ändert den Textinhalt des Label obj auf str
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt das Label obj auf dem Bildschirm an.

Led

- j_led** *function j_led (obj: integer; style: integer; color: integer) : integer;*
Erzeugt eine LED der Farbe **color** und der Form **style**
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt die LED **obj** in den Container **cont** ein.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an die LED **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled die LED **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
die LED **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled die LED **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an die LED **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der LED **obj** in Bildschirm-pixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*
Liefert die totale Höhe des aktuellen Fonts der LED **obj** in Bildschirmpixeln zurück.
- j_getheight** *function j_getheight (obj: integer) : integer;*
Liefert die Höhe der LED **obj** zurück.
- j_getparentid** *function j_getparentid (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *function j_getparent (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstate** *function j_getstate (obj: integer) : integer;*
Liefert true , falls die LED selektiert ist, sonst false .
- j_getstringwidth** *function j_getstringwidth (obj: integer; str: PChar) : integer;*

Liefert die Länge des **str** des aktuellen Fonts der LED **obj** in Bildschirmpixeln zurück.

j_getwidth	<i>function j_getwidth (obj: integer) : integer;</i> Liefert die Breite der LED obj zurück.
j_getxpos	<i>function j_getxpos (obj: integer) : integer;</i> Liefert die aktuelle horizontale Position der LED obj zurück.
j_getypos	<i>function j_getypos (obj: integer) : integer;</i> Liefert die aktuelle vertikale Position der LED obj zurück.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Zeigt die LED obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>function j_isparent (obj: integer; cont: integer) : integer;</i> Liefert true zurück, falls das cont die umgebende Komponente von obj ist, sonst false
j_isvisible	<i>function j_isvisible (obj: integer) : integer;</i> Liefert true zurück, falls obj sichtbar ist, sonst false
j_keylistener	<i>function j_keylistener (obj: integer) : integer;</i> Bindet an die LED obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) : integer;</i> Bindet an die LED obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>function j_popupmenu (obj: integer; label: PChar) : integer;</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (obj: integer) ;</i> Druckt den Inhalt der LED .
j_release	<i>procedure j_release (obj: integer) ;</i> Gibt die LED obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>procedure j_setborderpos (obj: integer; pos: integer) ;</i> Positioniert die LED obj an die angegebene Position pos . Im umgebenden Container muß einen Borderlayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i> Setzt den Cursor der LED obj auf cursor .

j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus der LED obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp der LED obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font der LED obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße der LED obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle der LED obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt die LED obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe der LED obj auf width and height .
j_setstate	<i>procedure j_setstate (obj: integer; bool: boolean) ;</i> Ist bool true so wird die LED selectiert.
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt die LED obj auf dem Bildschirm an.

List

- j_list** *function j_list (obj: integer; rows: integer) : integer;*
Erzeugt eine neue List und gibt deren Eventnummer zurück.
- j_additem** *procedure j_additem (obj: integer; str: PChar);*
Hängt ein neues Item mit dem Inhalt **str** an die Auswahlliste an.
- j_add** *procedure j_add (obj: integer; cont: integer);*
Fügt die List **obj** in den Container **cont** ein.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an die List **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_deselect** *function j_deselect (obj: integer; item: integer) : integer;*
Deselectiert das Item mit dem Index **item**.
- j_disable** *procedure j_disable (obj: integer);*
disabled die List **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer);*
die List **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer);*
enabled die List **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an die List **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der List **obj** in Bildschirm-pixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*
Liefert die totale Höhe des aktuellen Fonts der List **obj** in Bildschirmpixeln zurück.
- j_getheight** *function j_getheight (obj: integer) : integer;*
Liefert die Höhe der List **obj** zurück.
- j_getitemcount** *function j_getitemcount (obj: integer) : integer;*
Liefert die Anzahl der Items der List **obj** zurück.
- j_getitem** *function j_getitem (obj: integer; item: integer; str: PChar) : PChar;*
liefert den Inhalt des Items mit dem Index **item** zurück.

j_getparentid	<i>function j_getparentid (obj: integer) : integer;</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>function j_getparent (obj: integer) : integer;</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getselect	<i>function j_getselect (obj: integer) : integer;</i> Liefert den Index des selectierten Items der List .
j_getstringwidth	<i>function j_getstringwidth (obj: integer; str: PChar) : integer;</i> Liefert die Länge des str des aktuellen Fonts der List obj in Bildschirmpixeln zurück.
j_getwidth	<i>function j_getwidth (obj: integer) : integer;</i> Liefert die Breite der List obj zurück.
j_getxpos	<i>function j_getxpos (obj: integer) : integer;</i> Liefert die aktuelle horizontale Position der List obj zurück.
j_getypos	<i>function j_getypos (obj: integer) : integer;</i> Liefert die aktuelle vertikale Position der List obj zurück.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Zeigt die List obj nicht mehr auf dem Bildschirm an.
j_insert	<i>function j_insert (obj: integer; pos: integer; label: PChar) : integer;</i> fügt ein neues Item in an der Position pos ein. Das Item trägt den Inhalt label .
j_isparent	<i>function j_isparent (obj: integer; cont: integer) : integer;</i> Liefert true zurück, falls das cont die umgebende Komponente von obj ist, sonst false
j_isselct	<i>function j_isselct (obj: integer; item: integer) : integer;</i> Liefert true zurück, falls das Item mit dem Index item selectiert ist, sonst false
j_isvisible	<i>function j_isvisible (obj: integer) : integer;</i> Liefert true zurück, falls obj sichtbar ist, sonst false
j_keylistener	<i>function j_keylistener (obj: integer) : integer;</i> Bindet an die List obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) : integer;</i> Bindet an die List obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_multiplemode	<i>function j_multiplemode (obj: integer; bool: boolean) : integer;</i> die List erlaubt mehrfache Selectierung, falls bool true ist.
j_popupmenu	<i>function j_popupmenu (obj: integer; label: PChar) : integer;</i>

	Erzeugt ein neues Popumenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (obj: integer) ;</i> Druckt den Inhalt der List .
j_release	<i>procedure j_release (obj: integer) ;</i> Gibt die List obj aus dem umgebenden Container Element frei.
j_removeall	<i>function j_removeall (obj: integer) : integer;</i> entfernt alle Items aus der List .
j_removeitem	<i>function j_removeitem (obj: integer; item: PChar) : integer;</i> entfernt den ersten Eintrag item aus der List .
j_remove	<i>function j_remove (obj: integer; item: integer) : integer;</i> entfernt das Item mit dem Index item aus der List .
j_select	<i>function j_select (obj: integer; item: integer) : integer;</i> selectiert das Item mit dem Index item .
j_setborderpos	<i>procedure j_setborderpos (obj: integer; pos: integer) ;</i> Positioniert die List obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i> Setzt den Cursor der List obj auf cursor .
j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus der List obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp der List obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font der List obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße der List obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle der List obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .

- j_setnamedcolor** *procedure j_setnamedcolor (obj: integer; color: integer) ;*
Setzt die Vordergrundfarbe auf eine vordefinierte Farbe **color**.
- j_setpos** *procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;*
Verschiebt die List **obj** zur angegebenen Koordinate (**xpos,ypos**) an.
- j_setsize** *procedure j_setsize (obj: integer; width: integer; height: integer) ;*
Setzt Breite und Höhe der List **obj** auf **width** and **height**.
- j_show** *procedure j_show (obj: integer) ;*
Zeigt die List **obj** auf dem Bildschirm an.

Menu

j_menu	<p><i>function j_menu (obj: integer; str: PChar) : integer;</i> Erzeugt ein Menueintrag mit der Beschriftung label und liefert dessen Eventnummer zurück.</p>
j_checkmenuitem	<p><i>function j_checkmenuitem (obj: integer; label: PChar) : integer;</i> Erzeugt ein Checkmenuitem mit der Beschriftung label und liefert dessen Eventnummer zurück.</p>
j_disable	<p><i>procedure j_disable (obj: integer) ;</i> disabled das Menu obj, sodaß diese auf keine Benutzereingaben mehr reagiert.</p>
j_dispose	<p><i>procedure j_dispose (obj: integer) ;</i> das Menu obj wird mit allen Recourcen gelöscht.</p>
j_enable	<p><i>procedure j_enable (obj: integer) ;</i> enabled das Menu obj.</p>
j_getlength	<p><i>function j_getlength (obj: integer) : integer;</i> Liefert die Länge des Textinhaltes das Menu zurück.</p>
j_gettext	<p><i>function j_gettext (obj: integer; str: PChar) : PChar;</i> Liefert den Textinhalt des Menu .</p>
j_helpmenu	<p><i>function j_helpmenu (obj: integer; label: PChar) : integer;</i> Erzeugt ein Hilfe-Menueintrag mit der Beschriftung label und liefert dessen Eventnummer zurück.</p>
j_menuitem	<p><i>function j_menuitem (obj: integer; label: PChar) : integer;</i> Erzeugt ein MenuItem mit der Beschriftung label und liefert dessen Eventnummer zurück.</p>
j_menu	<p><i>function j_menu (obj: integer; str: PChar) : integer;</i> Erzeugt ein Menueintrag mit der Beschriftung label und liefert dessen Eventnummer zurück.</p>
j_seperator	<p><i>procedure j_seperator (obj: integer) ;</i> Erzeugt eine Trennlinie in das Menu .</p>
j_setfontname	<p><i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp des Menu obj auf name.</p>
j_setfont	<p><i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font des Menu obj entsprechend der Parameterliste.</p>
j_setfontsize	<p><i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße des Menu obj auf size.</p>

j_setfontstyle *procedure j_setfontstyle (obj: integer; style: integer) ;*
Setzt den Fontstyle des Menu **obj** auf **style**.

j_setshortcut *procedure j_setshortcut (obj: integer; chr: char) ;*
Bestimmt das Zeichen **chr** als Shortcut für das Menu .

j_settext *procedure j_settext (obj: integer; str: PChar) ;*
Ändert den Textinhalt des Menu **obj** auf **str**

Menuitem

j_menuitem	<i>function j_menuitem (obj: integer; label: PChar) : integer;</i> Erzeugt ein Menuitem mit der Beschriftung label und liefert dessen Eventnummer zurück.
j_disable	<i>procedure j_disable (obj: integer) ;</i> disabled das Menuitem obj , sodaß diese auf keine Benutzereingaben mehr reagiert.
j_dispose	<i>procedure j_dispose (obj: integer) ;</i> das Menuitem obj wird mit allen Recourcen gelöscht.
j_enable	<i>procedure j_enable (obj: integer) ;</i> enabled das Menuitem obj .
j_getlength	<i>function j_getlength (obj: integer) : integer;</i> Liefert die Länge des Textinhaltes das Menuitem zurück.
j_gettext	<i>function j_gettext (obj: integer; str: PChar) : PChar;</i> Liefert den Textinhalt des Menuitem .
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp des Menuitem obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font des Menuitem obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße des Menuitem obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle des Menuitem obj auf style .
j_setshortcut	<i>procedure j_setshortcut (obj: integer; chr: char) ;</i> Bestimmt das Zeichen chr als Shortcut für das Menuitem .
j_settext	<i>procedure j_settext (obj: integer; str: PChar) ;</i> Ändert den Textinhalt des Menuitem obj auf str

Meter

- j_meter** *function j_meter (obj: integer; title: PChar) : integer;*
Erzeugt ein analoges Anzeigeinstrument mit der Beschriftung **title**.
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt das Anzeigeinstrument **obj** in den Container **cont** ein.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an das Anzeigeinstrument **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled das Anzeigeinstrument **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
das Anzeigeinstrument **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled das Anzeigeinstrument **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an das Anzeigeinstrument **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getdanger** *procedure j_getdanger (obj: integer) ;*
Liefert den Warnlevel des Anzeigeinstrument **obj** zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Anzeigeinstrument **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*
Liefert die totale Höhe des aktuellen Fonts des Anzeigeinstrument **obj** in Bildschirmpixeln zurück.
- j_getheight** *function j_getheight (obj: integer) : integer;*
Liefert die Höhe des Anzeigeinstrument **obj** zurück.
- j_getparentid** *function j_getparentid (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *function j_getparent (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.

j_getstringwidth	<i>function j_getstringwidth (obj: integer; str: PChar) : integer;</i> Liefert die Länge des str des aktuellen Fonts des Anzeigeinstrument obj in Bildschirmpixeln zurück.
j_getwidth	<i>function j_getwidth (obj: integer) : integer;</i> Liefert die Breite des Anzeigeinstrument obj zurück.
j_getxpos	<i>function j_getxpos (obj: integer) : integer;</i> Liefert die aktuelle horizontale Position des Anzeigeinstrument obj zurück.
j_getypos	<i>function j_getypos (obj: integer) : integer;</i> Liefert die aktuelle vertikale Position des Anzeigeinstrument obj zurück.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Zeigt das Anzeigeinstrument obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>function j_isparent (obj: integer; cont: integer) : integer;</i> Liefert true zurück, falls das cont die umgebende Komponente von obj ist, sonst false
j_isvisible	<i>function j_isvisible (obj: integer) : integer;</i> Liefert true zurück, falls obj sichtbar ist, sonst false
j_keylistener	<i>function j_keylistener (obj: integer) : integer;</i> Bindet an das Anzeigeinstrument obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) : integer;</i> Bindet an das Anzeigeinstrument obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>function j_popupmenu (obj: integer; label: PChar) : integer;</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (obj: integer) ;</i> Druckt den Inhalt des Anzeigeinstrument .
j_release	<i>procedure j_release (obj: integer) ;</i> Gibt das Anzeigeinstrument obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>procedure j_setborderpos (obj: integer; pos: integer) ;</i> Positioniert das Anzeigeinstrument obj an die angegebene Position pos . Im umgebenden Container muß einen Borderlayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i>

	Setzt den Cursor des Anzeigeeinstrument obj auf cursor .
j_setdanger	<i>procedure j_setdanger (obj: integer; val: integer) ;</i> Stellt den Warnlevel dem Anzeigeeinstrument obj auf val .
j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus dem Anzeigeeinstrument obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp des Anzeigeeinstrument obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font des Anzeigeeinstrument obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße des Anzeigeeinstrument obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle des Anzeigeeinstrument obj auf style .
j_setmax	<i>function j_setmax (obj: integer; val: integer) : integer;</i> Stellt die obere Grenze dem Anzeigeeinstrument auf den Wert von val ein.
j_setmin	<i>function j_setmin (obj: integer; val: integer) : integer;</i> Stellt die untere Grenze dem Anzeigeeinstrument auf den Wert von val ein.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt das Anzeigeeinstrument obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe des Anzeigeeinstrument obj auf width and height .
j_setvalue	<i>procedure j_setvalue (obj: integer; val: integer) ;</i> Stellt den aktuellen Wert dem Anzeigeeinstrument obj auf val .
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt das Anzeigeeinstrument obj auf dem Bildschirm an.

Mouselistener

- j_mouselistener** *function j_mouselistener (obj: integer; kind: integer) : integer;*
Bindet an den Mouselistener **obj** einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_dispose** *procedure j_dispose (obj: integer) ;*
der Mouselistener **obj** wird mit allen Recourcen gelöscht.
- j_getmousebutton** *function j_getmousebutton (mouselistener: integer) : integer;*
Liefert die letzte benutzte Mousetaste zurück.
- j_getmousex** *function j_getmousex (mouselistener: integer) : integer;*
Liefert die aktuelle horizontale Position der Mouse zurück.
- j_getmousey** *function j_getmousey (mouselistener: integer) : integer;*
Liefert die aktuelle vertikale Position der Mouse zurück.

Panel

- j_panel** *function j_panel (obj: integer) : integer;*
Erzeugt ein neues panel und liefert dessen Eventnummer zurück.
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt das Panel **obj** in den Container **cont** ein.
- j_borderpanel** *function j_borderpanel (obj: integer; type: integer) : integer;*
Erzeugt ein neues Borderpanel und liefert dessen Eventnummer zurück.
- j_button** *function j_button (obj: integer; label: PChar) : integer;*
Erzeugt einen neuen Button mit dem Textinhalt **label**
- j_canvas** *function j_canvas (obj: integer; width: integer; height: integer) : integer;*
Erzeugt einen neuen Canvas mit der Breite **width** Pixel und einer Höhe von **height** Pixel.
- j_checkbox** *function j_checkbox (obj: integer; label: PChar) : integer;*
Erzeugt eine Checkbox mit dem Textinhalt **label** und liefert dessen Eventnummer zurück.
- j_choice** *function j_choice (obj: integer) : integer;*
Erzeugt eine neue Choice und gibt deren Eventnummer zurück.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an das Panel **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled das Panel **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
das Panel **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled das Panel **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an das Panel **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Panel **obj** in Bildschirm-pixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*

	Liefert die totale Höhe des aktuellen Fonts des Panel obj in Bildschirmpixeln zurück.
j_getheight	<i>function j_getheight (obj: integer) : integer;</i> Liefert die Höhe des Panel obj zurück.
j_getinheight	<i>function j_getinheight (cont: integer) : integer;</i> Liefert die Höhe des Panel obj des inneren Bereichs zurück.
j_getinsets	<i>function j_getinsets (obj: integer; side: integer) : integer;</i> Liefert die angeforderte Breite des inneren Randes dem Panel zurück.
j_getinwidth	<i>function j_getinwidth (cont: integer) : integer;</i> Liefert die Breite des Panel obj des inneren Bereichs zurück.
j_getlayoutid	<i>function j_getlayoutid (obj: integer) : integer;</i> Liefert die Eventnummer des Layoutmanagers des Containers obj zurück.
j_getparentid	<i>function j_getparentid (obj: integer) : integer;</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>function j_getparent (obj: integer) : integer;</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getstringwidth	<i>function j_getstringwidth (obj: integer; str: PChar) : integer;</i> Liefert die Länge des str des aktuellen Fonts des Panel obj in Bildschirmpixeln zurück.
j_getwidth	<i>function j_getwidth (obj: integer) : integer;</i> Liefert die Breite des Panel obj zurück.
j_getxpos	<i>function j_getxpos (obj: integer) : integer;</i> Liefert die aktuelle horizontale Position des Panel obj zurück.
j_getypos	<i>function j_getypos (obj: integer) : integer;</i> Liefert die aktuelle vertikale Position des Panel obj zurück.
j_graphicbutton	<i>function j_graphicbutton (obj: integer; filename: PChar) : integer;</i> Erzeugt einen neuen Graphicbutton mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_graphiclabel	<i>function j_graphiclabel (obj: integer; str: PChar) : integer;</i> Erzeugt ein neues Graphiclabel mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Zeigt das Panel obj nicht mehr auf dem Bildschirm an.
j_hscrollbar	<i>function j_hscrollbar (obj: integer) : integer;</i> Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.

j_isparent	<i>function j_isparent (obj: integer; cont: integer) : integer;</i> Liefert true zurück, falls das cont die umgebende Komponente von obj ist, sonst false
j_isvisible	<i>function j_isvisible (obj: integer) : integer;</i> Liefert true zurück, falls obj sichtbar ist, sonst false
j_keylistener	<i>function j_keylistener (obj: integer) : integer;</i> Bindet an das Panel obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_label	<i>function j_label (obj: integer; label: PChar) : integer;</i> Erzeugt ein neues Label Element mit dem Textinhalt label
j_led	<i>function j_led (obj: integer; style: integer; color: integer) : integer;</i> Erzeugt eine LED der Farbe color und der Form style
j_line	<i>function j_line (obj: integer; orient: integer; style: integer; length: integer) : integer;</i> Erzeugt ein neues Linienelement und liefert dessen Eventnummer zurück
j_list	<i>function j_list (obj: integer; rows: integer) : integer;</i> Erzeugt eine neue List und gibt deren Eventnummer zurück.
j_meter	<i>function j_meter (obj: integer; title: PChar) : integer;</i> Erzeugt ein analoges Anzeigeinstrument mit der Beschriftung title .
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) : integer;</i> Bindet an das Panel obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_pack	<i>procedure j_pack (obj: integer) ;</i> Veranlaßt den Layoutmanager von obj die minimale Größe zu berechnen.
j_panel	<i>function j_panel (obj: integer) : integer;</i> Erzeugt ein neues panel und liefert dessen Eventnummer zurück.
j_popupmenu	<i>function j_popupmenu (obj: integer; label: PChar) : integer;</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (obj: integer) ;</i> Druckt den Inhalt des Panel .
j_progressbar	<i>function j_progressbar (obj: integer; orient: integer) : integer;</i> Erzeugt eine Progress Bar mit der Ausrichtung orient .
j_radiogroup	<i>function j_radiogroup (obj: integer) : integer;</i> Erzeugt eine neue Radiogroup und liefert dessen Eventnummer zurück.
j_releaseall	<i>procedure j_releaseall (obj: integer) ;</i> Gibt alle Elemente aus dem Panel obj frei.

j_release	<i>procedure j_release (obj: integer) ;</i> Gibt das Panel obj aus dem umgebenden Container Element frei.
j_scrollpane	<i>function j_scrollpane (obj: integer) : integer;</i> Erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück.
j_setalign	<i>procedure j_setalign (obj: integer; align: integer) ;</i> Setzt die Ausrichtung innerhalb des Panel obj auf align . Ein Flowlayout Manager wird vorausgesetzt.
j_setborderlayout	<i>procedure j_setborderlayout (obj: integer) ;</i> Integriert einen BorderLayout Manager in das Panel obj .
j_setborderpos	<i>procedure j_setborderpos (obj: integer; pos: integer) ;</i> Positioniert das Panel obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i> Setzt den Cursor des Panel obj auf cursor .
j_setfixlayout	<i>procedure j_setfixlayout (obj: integer) ;</i> Integriert einen Fixlayout Manager in das Panel obj (voreingestellte Layoutmanager).
j_setflowfill	<i>procedure j_setflowfill (obj: integer; bool: boolean) ;</i> Setzt die Höhe (Breite) aller enthaltenen Komponenten auf die Höhe (Breite) von dem Panel obj . Ein Flowlayout Manager wird vorausgesetzt.
j_setflowlayout	<i>procedure j_setflowlayout (obj: integer; align: integer) ;</i> Integriert einen Flowlayout Manager in das Panel obj mit der Ausrichtung align .
j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus dem Panel obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp des Panel obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font des Panel obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße des Panel obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle des Panel obj auf style .

j_setgridlayout	<i>procedure j_setgridlayout (obj: integer; row: integer; col: integer) ;</i> Integriert einen Gridlayout Manager in das Panel obj mit row Reihen und col Spalten.
j_sethgap	<i>procedure j_sethgap (obj: integer; hgap: integer) ;</i> Setzt den horizontalen Abstand innerhalb des Panel obj auf hgap Pixel.
j_setinsets	<i>procedure j_setinsets (obj: integer; top: integer; bottom: integer; left: integer; right: integer) ;</i> Setzt den inneren Rahmen auf die angegebenen Größen.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setnolayout	<i>procedure j_setnolayout (obj: integer) ;</i> Entfernt den aktuellen Layout Manager aus dem Panel obj .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt das Panel obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe des Panel obj auf width and height .
j_setvgap	<i>procedure j_setvgap (obj: integer; vgap: integer) ;</i> Setzt den vertikalen Abstand innerhalb des Panel obj auf vgap Pixel.
j_sevensegment	<i>function j_sevensegment (obj: integer; color: integer) : integer;</i> Erzeugt eine Siebensegment Anzeige der Farbe color .
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt das Panel obj auf dem Bildschirm an.
j_textarea	<i>function j_textarea (obj: integer; rows: integer; columns: integer) : integer;</i> Erzeugt eine neue Textarea und gibt dessen Eventnummer zurück.
j_textfield	<i>function j_textfield (obj: integer; columns: integer) : integer;</i> Erzeugt eine neues Textfield und gibt dessen Eventnummer zurück.
j_vscrollbar	<i>function j_vscrollbar (obj: integer) : integer;</i> Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.

<h2>PopupMenu</h2>

j_popupmenu	<p><i>function j_popupmenu (obj: integer; label: PChar) : integer;</i> Erzeugt ein neues Popumenu mit dem Inhalt label und liefert dessen Eventnummer zurück.</p>
j_checkmenuitem	<p><i>function j_checkmenuitem (obj: integer; label: PChar) : integer;</i> Erzeugt ein Checkmenuitem mit der Beschriftung label und liefert dessen Eventnummer zurück.</p>
j_disable	<p><i>procedure j_disable (obj: integer) ;</i> disabled das Popumenu obj, sodaß diese auf keine Benutzereingaben mehr reagiert.</p>
j_dispose	<p><i>procedure j_dispose (obj: integer) ;</i> das Popumenu obj wird mit allen Recourcen gelöscht.</p>
j_enable	<p><i>procedure j_enable (obj: integer) ;</i> enabled das Popumenu obj.</p>
j_getlength	<p><i>function j_getlength (obj: integer) : integer;</i> Liefert die Länge des Textinhaltes das Popumenu zurück.</p>
j_gettext	<p><i>function j_gettext (obj: integer; str: PChar) : PChar;</i> Liefert den Textinhalt des Popumenu .</p>
j_menuitem	<p><i>function j_menuitem (obj: integer; label: PChar) : integer;</i> Erzeugt ein Menuitem mit der Beschriftung label und liefert dessen Eventnummer zurück.</p>
j_seperator	<p><i>procedure j_seperator (obj: integer) ;</i> Erzeugt eine Trennlinie in das Popumenu .</p>
j_setfontname	<p><i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp des Popumenu obj auf name.</p>
j_setfont	<p><i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font des Popumenu obj entsprechend der Parameterliste.</p>
j_setfontsize	<p><i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße des Popumenu obj auf size.</p>
j_setfontstyle	<p><i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle des Popumenu obj auf style.</p>
j_setshortcut	<p><i>procedure j_setshortcut (obj: integer; chr: char) ;</i> Bestimmt das Zeichen chr als Shortcut für das Popumenu .</p>
j_settext	<p><i>procedure j_settext (obj: integer; str: PChar) ;</i></p>

Ändert den Textinhalt des Popupmenu **obj** auf **str**

j_showpopup

procedure j_showpopup (obj: integer; xpos: integer; ypos: integer) ;
Zeigt das Popupmenu an der Koordinate (**xpos,ypos**) an.

Printer

- j_printer** *function j_printer (frame: integer) : integer;*
Erzeugt ein neues Objekt, das eine Papierseite des Druckes repräsentiert.
- j_cliprect** *procedure j_cliprect (obj: integer; x: integer; y: integer; width: integer; height: integer) ;*
Setzt den Clipping Bereich auf das spezifizierte Rechteck (**x**, **y**, **width**, **height**).
- j_dispose** *procedure j_dispose (obj: integer) ;*
der Printer **obj** wird mit allen Recourcen gelöscht.
- j_drawarc** *procedure j_drawarc (obj: integer; x: integer; y: integer; rx: integer; ry: integer; arc1: integer; arc2: integer) ;*
Zeichnet einen Kreisbogen mit dem Mittelpunkt (**x**, **y**) und den Hauptradien **rx** und **ry**. Die Parameter **arc1** und **arc2** bestimmen den Start- bzw. den Endwinkel des Kreisbogens.
- j_drawcircle** *procedure j_drawcircle (obj: integer; x: integer; y: integer; r: integer) ;*
Zeichnet einen Kreis mit dem Mittelpunkt (**x,y**) und dem Radius **r**.
- j_drawimage** *procedure j_drawimage (obj: integer; image: integer; x: integer; y: integer) ;*
Kopiert das Image mit der Eventnummer **image** an die Position (**x**, **y**).
- j_drawimagesource** *procedure j_drawimagesource (obj: integer; x: integer; y: integer; w: integer; h: integer; r: var integer; g: var integer; b: var integer) ;*
Die Funktion stellt ein Bild an der Position (**x,y**) mit der Breite **w** und der Höhe **h** dar. Die Bilddaten werden als rot-, grün- und blau- Kanal in den Arrays **r,g,b** übergeben.
- j_drawline** *procedure j_drawline (obj: integer; x1: integer; y1: integer; x2: integer; y2: integer) ;*
Zeichnet eine Linie in der momentan gesetzten Vordergrundfarbe von (**x1,y1**) nach (**x2,y2**)
- j_drawoval** *procedure j_drawoval (obj: integer; x: integer; y: integer; rx: integer; ry: integer) ;*
Zeichnet einen Oval mit dem Mittelpunkt (**x,y**) und den Hauptradien **rx** und **ry**.
- j_drawpixel** *procedure j_drawpixel (obj: integer; x: integer; y: integer) ;*
Zeichnet einen Pixel in der momentan gesetzten Vordergrundfarbe an die Position (**x,y**).
- j_drawpolygon** *procedure j_drawpolygon (obj: integer; len: integer; x: var integer; y: var integer) ;*

- Zeichnet ein Polygon durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.
- j_drawpolyline** *procedure j_drawpolyline (obj: integer; len: integer; x: var integer; y: var integer) ;*
Zeichnet eine Polylinie durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.
- j_drawrect** *procedure j_drawrect (obj: integer; x: integer; y: integer; width: integer; height: integer) ;*
Zeichnet ein Rechteck an die Position (**x,y**) mit der Breite **width** und der Höhe **height**.
- j_drawroundrect** *procedure j_drawroundrect (obj: integer; x: integer; y: integer; width: integer; height: integer; arcx: integer; arcy: integer) ;*
Zeichnet ein Rechteck mit abgerundeten Ecken an die Position (**x,y**) mit der Breite **width** und der Höhe **height**. Die Radien der Ecken werden durch **arcx** und **arcy** bestimmt.
- j_drawscaleddimimage** *procedure j_drawscaleddimimage (obj: integer; image: integer; sx: integer; sy: integer; sw: integer; sh: integer; tx: integer; ty: integer; tw: integer; th: integer) ;*
Kopiert einen Ausschnitt des Bildes **image**, der durch die Parameter (**sx, sy, sw, sh**) bestimmt wird an die Position (**tx, ty**) und scaliert den Ausschnitt auf die Breite **tw** und die Höhe **th**
- j_drawstring** *procedure j_drawstring (obj: integer; x: integer; y: integer; str: PChar) ;*
Zeichnet den String **str** an die durch die Parameter (**x,y**) festgelegte Koordinate.
- j_fillarc** *procedure j_fillarc (obj: integer; x: integer; y: integer; rx: integer; ry: integer; arc1: integer; arc2: integer) ;*
Zeichnet einen gefüllten Kreisbogen mit dem Mittelpunkt (**x,y**) und den Hauptradien **rx** und **ry**. Die Parameter **arc1** und **arc2** bestimmen den Start- bzw. den Endwinkel des Kreisbogens.
- j_fillcircle** *procedure j_fillcircle (obj: integer; x: integer; y: integer; r: integer) ;*
Zeichnet einen gefüllten Kreis mit dem Mittelpunkt (**x,y**) und dem Radius **r**.
- j_filloval** *procedure j_filloval (obj: integer; x: integer; y: integer; rx: integer; ry: integer) ;*
Zeichnet einen gefüllten Oval mit dem Mittelpunkt (**x,y**) und den Hauptradien **rx** und **ry**.
- j_fillpolygon** *procedure j_fillpolygon (obj: integer; len: integer; x: var integer; y: var integer) ;*
Zeichnet ein gefülltes Polygon durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.
- j_fillrect** *procedure j_fillrect (obj: integer; x: integer; y: integer; width: integer; height: integer) ;*
Zeichnet ein gefülltes Rechteck an die Position (**x,y**) mit der Breite **width** und der Höhe **height**.

- j_fillroundrect** *procedure j_fillroundrect (obj: integer; x: integer; y: integer; width: integer; height: integer; arcx: integer; arcy: integer) ;*
Zeichnet ein gefülltes Rechteck mit abgerundeten Ecken an die Position **(x,y)**. Das Rechteck hat eine Breite von **width** und eine Höhe von **height** Pixeln. Die Form der runden Ecken wird durch die Parameter **arcx** und **arcy** bestimmt.
- j_print** *procedure j_print (obj: integer) ;*
Druckt den Inhalt des Printer .
- j_setxor** *procedure j_setxor (obj: integer; bool: boolean) ;*
Schaltet den Painting Mode auf XOR, falls bool = true ist. Im XOR Modus heben sich zwei gleiche Graphikbefehle auf.
- j_translate** *procedure j_translate (obj: integer; x: integer; y: integer) ;*
Setzt den Ursprung des Koordinatensystems auf **(x, y)**.

Progressbar

- j_progressbar** *function j_progressbar (obj: integer; orient: integer) : integer;*
Erzeugt eine Progress Bar mit der Ausrichtung **orient**.
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt die Progressbar **obj** in den Container **cont** ein.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an die Progressbar **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled die Progressbar **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
die Progressbar **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled die Progressbar **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an die Progressbar **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der Progressbar **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*
Liefert die totale Höhe des aktuellen Fonts der Progressbar **obj** in Bildschirmpixeln zurück.
- j_getheight** *function j_getheight (obj: integer) : integer;*
Liefert die Höhe der Progressbar **obj** zurück.
- j_getparentid** *function j_getparentid (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *function j_getparent (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstringwidth** *function j_getstringwidth (obj: integer; str: PChar) : integer;*

Liefert die Länge des **str** des aktuellen Fonts der Progressbar **obj** in Bildschirmpixeln zurück.

- j_getwidth** *function j_getwidth (obj: integer) : integer;*
Liefert die Breite der Progressbar **obj** zurück.
- j_getxpos** *function j_getxpos (obj: integer) : integer;*
Liefert die aktuelle horizontale Position der Progressbar **obj** zurück.
- j_getypos** *function j_getypos (obj: integer) : integer;*
Liefert die aktuelle vertikale Position der Progressbar **obj** zurück.
- j_hide** *procedure j_hide (obj: integer) ;*
Zeigt die Progressbar **obj** nicht mehr auf dem Bildschirm an.
- j_isparent** *function j_isparent (obj: integer; cont: integer) : integer;*
Liefert true zurück, falls das **cont** die umgebende Komponente von **obj** ist, sonst false
- j_isvisible** *function j_isvisible (obj: integer) : integer;*
Liefert true zurück, falls **obj** sichtbar ist, sonst false
- j_keylistener** *function j_keylistener (obj: integer) : integer;*
Bindet an die Progressbar **obj** einen neuen Keylistener, und liefert dessen Eventnummer zurück.
- j_mouselistener** *function j_mouselistener (obj: integer; kind: integer) : integer;*
Bindet an die Progressbar **obj** einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_popupmenu** *function j_popupmenu (obj: integer; label: PChar) : integer;*
Erzeugt ein neues Popupmenu mit dem Inhalt **label** und liefert dessen Eventnummer zurück.
- j_print** *procedure j_print (obj: integer) ;*
Druckt den Inhalt der Progressbar .
- j_release** *procedure j_release (obj: integer) ;*
Gibt die Progressbar **obj** aus dem umgebenden Container Element frei.
- j_setborderpos** *procedure j_setborderpos (obj: integer; pos: integer) ;*
Positioniert die Progressbar **obj** an die angegebene Position **pos**. Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
- j_setcolorbg** *procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;*
Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
- j_setcolor** *procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;*
Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
- j_setcursor** *function j_setcursor (obj: integer; cursor: integer) : integer;*
Setzt den Cursor der Progressbar **obj** auf **cursor**.

j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus der Progressbar obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp der Progressbar obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font der Progressbar obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße der Progressbar obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle der Progressbar obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt die Progressbar obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe der Progressbar obj auf width and height .
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt die Progressbar obj auf dem Bildschirm an.

Radiobutton

- j_radiobutton** *function j_radiobutton (obj: integer; label: PChar) : integer;*
Erzeugt einen Radiobutton mit dem Textinhalt **label** und liefert dessen Eventnummer zurück.
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt den Radiobutton **obj** in den Container **cont** ein.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an den Radiobutton **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled den Radiobutton **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
der Radiobutton **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled den Radiobutton **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an den Radiobutton **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Radiobutton **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*
Liefert die totale Höhe des aktuellen Fonts des Radiobutton **obj** in Bildschirmpixeln zurück.
- j_getheight** *function j_getheight (obj: integer) : integer;*
Liefert die Höhe des Radiobutton **obj** zurück.
- j_getparentid** *function j_getparentid (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *function j_getparent (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstate** *function j_getstate (obj: integer) : integer;*

	Liefert true , falls der Radiobutton selektiert ist, sonst false .
j_getstringwidth	<i>function j_getstringwidth (obj: integer; str: PChar) : integer;</i> Liefert die Länge des str des aktuellen Fonts des Radiobutton obj in Bildschirmpixeln zurück.
j_gettext	<i>function j_gettext (obj: integer; str: PChar) : PChar;</i> Liefert den Textinhalt des Radiobutton .
j_getwidth	<i>function j_getwidth (obj: integer) : integer;</i> Liefert die Breite des Radiobutton obj zurück.
j_getxpos	<i>function j_getxpos (obj: integer) : integer;</i> Liefert die aktuelle horizontale Position des Radiobutton obj zurück.
j_getypos	<i>function j_getypos (obj: integer) : integer;</i> Liefert die aktuelle vertikale Position des Radiobutton obj zurück.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Zeigt den Radiobutton obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>function j_isparent (obj: integer; cont: integer) : integer;</i> Liefert true zurück, falls das cont die umgebende Komponente von obj ist, sonst false
j_isvisible	<i>function j_isvisible (obj: integer) : integer;</i> Liefert true zurück, falls obj sichtbar ist, sonst false
j_keylistener	<i>function j_keylistener (obj: integer) : integer;</i> Bindet an den Radiobutton obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) : integer;</i> Bindet an den Radiobutton obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>function j_popupmenu (obj: integer; label: PChar) : integer;</i> Erzeugt ein neues Popumenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (obj: integer) ;</i> Druckt den Inhalt des Radiobutton .
j_release	<i>procedure j_release (obj: integer) ;</i> Gibt den Radiobutton obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>procedure j_setborderpos (obj: integer; pos: integer) ;</i> Positioniert den Radiobutton obj an die angegebene Position pos . Im umgebenden Container muß einen Borderlayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert

j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i> Setzt den Cursor des Radiobutton obj auf cursor .
j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus dem Radiobutton obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp des Radiobutton obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font des Radiobutton obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße des Radiobutton obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle des Radiobutton obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt den Radiobutton obj zur angegebenen Koordinate (xpos,ypos) an.
j_setradiogroup	<i>function j_setradiogroup (rbutton,: integer; rgroup: integer) : integer;</i> Weist dem Radiobutton rbutton die Radiogroup rgroup zu. Der Radiobutton wird aus der alten Radiogroup entfernt.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe des Radiobutton obj auf width and height .
j_setstate	<i>procedure j_setstate (obj: integer; bool: boolean) ;</i> Ist bool true so wird den Radiobutton selectiert.
j_settext	<i>procedure j_settext (obj: integer; str: PChar) ;</i> Ändert den Textinhalt des Radiobutton obj auf str
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt den Radiobutton obj auf dem Bildschirm an.

Sevenssegment

- j_sevenssegment** *function j_sevenssegment (obj: integer; color: integer) : integer;*
Erzeugt eine Siebensegment Anzeige der Farbe **color**.
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt die Siebensegment-Anzeige **obj** in den Container **cont** ein.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an die Siebensegment-Anzeige **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled die Siebensegment-Anzeige **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
die Siebensegment-Anzeige **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled die Siebensegment-Anzeige **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an die Siebensegment-Anzeige **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der Siebensegment-Anzeige **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*
Liefert die totale Höhe des aktuellen Fonts der Siebensegment-Anzeige **obj** in Bildschirmpixeln zurück.
- j_getheight** *function j_getheight (obj: integer) : integer;*
Liefert die Höhe der Siebensegment-Anzeige **obj** zurück.
- j_getparentid** *function j_getparentid (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird **-1** zurückgegeben.
- j_getparent** *function j_getparent (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird **-1** zurückgegeben.
- j_getstringwidth** *function j_getstringwidth (obj: integer; str: PChar) : integer;*

Liefert die Länge des **str** des aktuellen Fonts der Siebensegment-Anzeige **obj** in Bildschirmpixeln zurück.

- j_getwidth** *function j_getwidth (obj: integer) : integer;*
Liefert die Breite der Siebensegment-Anzeige **obj** zurück.
- j_getxpos** *function j_getxpos (obj: integer) : integer;*
Liefert die aktuelle horizontale Position der Siebensegment-Anzeige **obj** zurück.
- j_getypos** *function j_getypos (obj: integer) : integer;*
Liefert die aktuelle vertikale Position der Siebensegment-Anzeige **obj** zurück.
- j_hide** *procedure j_hide (obj: integer) ;*
Zeigt die Siebensegment-Anzeige **obj** nicht mehr auf dem Bildschirm an.
- j_isparent** *function j_isparent (obj: integer; cont: integer) : integer;*
Liefert true zurück, falls das **cont** die umgebende Komponente von **obj** ist, sonst false
- j_isvisible** *function j_isvisible (obj: integer) : integer;*
Liefert true zurück, falls **obj** sichtbar ist, sonst false
- j_keylistener** *function j_keylistener (obj: integer) : integer;*
Bindet an die Siebensegment-Anzeige **obj** einen neuen Keylistener, und liefert dessen Eventnummer zurück.
- j_mouselistener** *function j_mouselistener (obj: integer; kind: integer) : integer;*
Bindet an die Siebensegment-Anzeige **obj** einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_popupmenu** *function j_popupmenu (obj: integer; label: PChar) : integer;*
Erzeugt ein neues Popupmenu mit dem Inhalt **label** und liefert dessen Eventnummer zurück.
- j_print** *procedure j_print (obj: integer) ;*
Druckt den Inhalt der Siebensegment-Anzeige .
- j_release** *procedure j_release (obj: integer) ;*
Gibt die Siebensegment-Anzeige **obj** aus dem umgebenden Container Element frei.
- j_setborderpos** *procedure j_setborderpos (obj: integer; pos: integer) ;*
Positioniert die Siebensegment-Anzeige **obj** an die angegebene Position **pos**. Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
- j_setcolorbg** *procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;*
Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
- j_setcolor** *procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;*
Setzt die Vordergrundfarbe auf den übergebenen RGB Wert

j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i> Setzt den Cursor der Siebensegment-Anzeige obj auf cursor .
j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus der Siebensegment-Anzeige obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp der Siebensegment-Anzeige obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font der Siebensegment-Anzeige obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße der Siebensegment-Anzeige obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle der Siebensegment-Anzeige obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt die Siebensegment-Anzeige obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe der Siebensegment-Anzeige obj auf width and height .
j_setvalue	<i>procedure j_setvalue (obj: integer; val: integer) ;</i> Stellt den aktuellen Wert der Siebensegment-Anzeige obj auf val .
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt die Siebensegment-Anzeige obj auf dem Bildschirm an.

Scrollpane

- j_scrollpane** *function j_scrollpane (obj: integer) : integer;*
Erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück.
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt die Scrollpane **obj** in den Container **cont** ein.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an die Scrollpane **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled die Scrollpane **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
die Scrollpane **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled die Scrollpane **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an die Scrollpane **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der Scrollpane **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*
Liefert die totale Höhe des aktuellen Fonts der Scrollpane **obj** in Bildschirmpixeln zurück.
- j_getheight** *function j_getheight (obj: integer) : integer;*
Liefert die Höhe der Scrollpane **obj** zurück.
- j_getparentid** *function j_getparentid (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *function j_getparent (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstringwidth** *function j_getstringwidth (obj: integer; str: PChar) : integer;*
Liefert die Länge des **str** des aktuellen Fonts der Scrollpane **obj** in Bildschirmpixeln zurück.

- j_getviewportheight** *function j_getviewportheight (obj: integer) : integer;*
Gibt die Höhe des sichtbaren Teils der Scrollpane **obj** zurück.
- j_getviewportwidth** *function j_getviewportwidth (obj: integer) : integer;*
Gibt die Breite des sichtbaren Teils der Scrollpane **obj** zurück.
- j_getwidth** *function j_getwidth (obj: integer) : integer;*
Liefert die Breite der Scrollpane **obj** zurück.
- j_getxpos** *function j_getxpos (obj: integer) : integer;*
Liefert die aktuelle horizontale Position der Scrollpane **obj** zurück.
- j_getypos** *function j_getypos (obj: integer) : integer;*
Liefert die aktuelle vertikale Position der Scrollpane **obj** zurück.
- j_hide** *procedure j_hide (obj: integer) ;*
Zeigt die Scrollpane **obj** nicht mehr auf dem Bildschirm an.
- j_hscrollbar** *function j_hscrollbar (obj: integer) : integer;*
Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.
- j_isparent** *function j_isparent (obj: integer; cont: integer) : integer;*
Liefert true zurück, falls das **cont** die umgebende Komponente von **obj** ist, sonst false
- j_isvisible** *function j_isvisible (obj: integer) : integer;*
Liefert true zurück, falls **obj** sichtbar ist, sonst false
- j_keylistener** *function j_keylistener (obj: integer) : integer;*
Bindet an die Scrollpane **obj** einen neuen Keylistener, und liefert dessen Eventnummer zurück.
- j_mouselistener** *function j_mouselistener (obj: integer; kind: integer) : integer;*
Bindet an die Scrollpane **obj** einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_popupmenu** *function j_popupmenu (obj: integer; label: PChar) : integer;*
Erzeugt ein neues Popupmenu mit dem Inhalt **label** und liefert dessen Eventnummer zurück.
- j_print** *procedure j_print (obj: integer) ;*
Druckt den Inhalt der Scrollpane .
- j_release** *procedure j_release (obj: integer) ;*
Gibt die Scrollpane **obj** aus dem umgebenden Container Element frei.
- j_setborderpos** *procedure j_setborderpos (obj: integer; pos: integer) ;*
Positioniert die Scrollpane **obj** an die angegebene Position **pos**. Im umgebenden Container muß ein BorderLayout Manager eingestellt sein.
- j_setcolorbg** *procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;*
Setzt die Hintergrundfarbe auf den übergebenen RGB Wert

j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i> Setzt den Cursor der Scrollpane obj auf cursor .
j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus der Scrollpane obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp der Scrollpane obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font der Scrollpane obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße der Scrollpane obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle der Scrollpane obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt die Scrollpane obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe der Scrollpane obj auf width and height .
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt die Scrollpane obj auf dem Bildschirm an.
j_vscrollbar	<i>function j_vscrollbar (obj: integer) : integer;</i> Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.

Textarea

- j_textarea** *function j_textarea (obj: integer; rows: integer; columns: integer) : integer;*
Erzeugt eine neue Textarea und gibt dessen Eventnummer zurück.
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt die Textarea **obj** in den Container **cont** ein.
- j_appendtext** *procedure j_appendtext (obj: integer; text: PChar) ;*
Hängt den Text **text** am Ende von **obj** an.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an die Textarea **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_delete** *procedure j_delete (obj: integer; start: integer; end: integer) ;*
löscht den Text von Position **start** bis Position **end**.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled die Textarea **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
die Textarea **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled die Textarea **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an die Textarea **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getcolumns** *procedure j_getcolumns (obj: integer) ;*
Gibt die Anzahl der Spalten von **obj** zurück.
- j_getcurpos** *function j_getcurpos (obj: integer) : integer;*
Liefert die Position des Text Cursors zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der Textarea **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*
Liefert die totale Höhe des aktuellen Fonts der Textarea **obj** in Bildschirmpixeln zurück.
- j_getheight** *function j_getheight (obj: integer) : integer;*

	Liefert die Höhe der Textarea obj zurück.
j_getlength	<i>function j_getlength (obj: integer) : integer;</i> Liefert die Länge des Textinhaltes die Textarea zurück.
j_getparentid	<i>function j_getparentid (obj: integer) : integer;</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>function j_getparent (obj: integer) : integer;</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getrows	<i>procedure j_getrows (obj: integer) ;</i> Gibt die Anzahl der Reihen von obj zurück.
j_getselend	<i>function j_getselend (obj: integer) : integer;</i> Liefert die Endposition des selectierten Textes.
j_getselstart	<i>function j_getselstart (obj: integer) : integer;</i> Liefert die Anfangsposition des selectierten Textes.
j_getseltext	<i>function j_getseltext (obj: integer; text: PChar) : PChar;</i> Liefert den selectierten Text die Textarea obj zurück.
j_getstringwidth	<i>function j_getstringwidth (obj: integer; str: PChar) : integer;</i> Liefert die Länge des str des aktuellen Fonts der Textarea obj in Bildschirm-pixeln zurück.
j_gettext	<i>function j_gettext (obj: integer; str: PChar) : PChar;</i> Liefert den Textinhalt der Textarea .
j_getwidth	<i>function j_getwidth (obj: integer) : integer;</i> Liefert die Breite der Textarea obj zurück.
j_getxpos	<i>function j_getxpos (obj: integer) : integer;</i> Liefert die aktuelle horizontale Position der Textarea obj zurück.
j_getypos	<i>function j_getypos (obj: integer) : integer;</i> Liefert die aktuelle vertikale Position der Textarea obj zurück.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Zeigt die Textarea obj nicht mehr auf dem Bildschirm an.
j_inserttext	<i>procedure j_inserttext (obj: integer; text: PChar; pos: integer) ;</i> Fügt zusätzlichen Text text an der Position pos ein.
j_isparent	<i>function j_isparent (obj: integer; cont: integer) : integer;</i> Liefert true zurück, falls das cont die umgebende Komponente von obj ist, sonst false
j_isvisible	<i>function j_isvisible (obj: integer) : integer;</i>

Liefert true zurück, falls **obj** sichtbar ist, sonst false

- j_keylistener** *function j_keylistener (obj: integer) : integer;*
Bindet an die Textarea **obj** einen neuen Keylistener, und liefert dessen Eventnummer zurück.
- j_mouselistener** *function j_mouselistener (obj: integer; kind: integer) : integer;*
Bindet an die Textarea **obj** einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_popupmenu** *function j_popupmenu (obj: integer; label: PChar) : integer;*
Erzeugt ein neues Popumenu mit dem Inhalt **label** und liefert dessen Eventnummer zurück.
- j_print** *procedure j_print (obj: integer) ;*
Druckt den Inhalt der Textarea .
- j_release** *procedure j_release (obj: integer) ;*
Gibt die Textarea **obj** aus dem umgebenden Container Element frei.
- j_replacetext** *procedure j_replacetext (obj: integer; text: PChar; start: integer; end: integer) ;*
Ersetzt den Text zwischen Position **start** bis **end** durch **text**.
- j_selectall** *procedure j_selectall (obj: integer) ;*
Selectiert den gesamten Textinhalt der Textarea .
- j_selecttext** *procedure j_selecttext (obj: integer; start: integer; end: integer) ;*
Selectiert den Text von Position **start** bis **end**.
- j_setborderpos** *procedure j_setborderpos (obj: integer; pos: integer) ;*
Positioniert die Textarea **obj** an die angegebene Position **pos**. Im umgebenden Container muß einen Borderlayout Manager eingestellt sein.
- j_setcolorbg** *procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;*
Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
- j_setcolor** *procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;*
Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
- j_setcolumns** *procedure j_setcolumns (obj: integer; columns: integer) ;*
Setzt die Anzahl Spalten von **obj** auf **columns**.
- j_setcurpos** *procedure j_setcurpos (obj: integer; pos: integer) ;*
Setzt den Text Cursor auf die Position **pos**.
- j_setcursor** *function j_setcursor (obj: integer; cursor: integer) : integer;*
Setzt den Cursor der Textarea **obj** auf **cursor**.
- j_seteditable** *procedure j_seteditable (obj: integer; bool: boolean) ;*
Ist **bool** true , so ist die Textarea editierbar, ansonsten ist die Textarea nur lesbar für den Benutzer.

j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus der Textarea obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp der Textarea obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font der Textarea obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße der Textarea obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle der Textarea obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt die Textarea obj zur angegebenen Koordinate (xpos,ypos) an.
j_setrows	<i>procedure j_setrows (obj: integer; rows: integer) ;</i> Setzt die Anzahl Reihen von obj auf rows .
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe der Textarea obj auf width and height .
j_settext	<i>procedure j_settext (obj: integer; str: PChar) ;</i> Ändert den Textinhalt der Textarea obj auf str
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt die Textarea obj auf dem Bildschirm an.

Textfield

- j_textfield** *function j_textfield (obj: integer; columns: integer) : integer;*
Erzeugt eine neues Textfield und gibt dessen Eventnummer zurück.
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt das Textfield **obj** in den Container **cont** ein.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an das Textfield **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled das Textfield **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
das Textfield **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled das Textfield **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an das Textfield **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getcolumns** *procedure j_getcolumns (obj: integer) ;*
Gibt die Anzahl der Spalten von **obj** zurück.
- j_getcurpos** *function j_getcurpos (obj: integer) : integer;*
Liefert die Position des Text Cursors zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Textfield **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*
Liefert die totale Höhe des aktuellen Fonts des Textfield **obj** in Bildschirmpixeln zurück.
- j_getheight** *function j_getheight (obj: integer) : integer;*
Liefert die Höhe des Textfield **obj** zurück.
- j_getlength** *function j_getlength (obj: integer) : integer;*
Liefert die Länge des Textinhaltes das Textfield zurück.
- j_getparentid** *function j_getparentid (obj: integer) : integer;*

	Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>function j_getparent (obj: integer) : integer;</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getselend	<i>function j_getselend (obj: integer) : integer;</i> Liefert die Endposition des selectierten Textes.
j_getselstart	<i>function j_getselstart (obj: integer) : integer;</i> Liefert die Anfangsposition des selectierten Textes.
j_getseltext	<i>function j_getseltext (obj: integer; text: PChar) : PChar;</i> Liefert den selectierten Text das Textfield obj zurück.
j_getstringwidth	<i>function j_getstringwidth (obj: integer; str: PChar) : integer;</i> Liefert die Länge des str des aktuellen Fonts des Textfield obj in Bildschirm-pixeln zurück.
j_gettext	<i>function j_gettext (obj: integer; str: PChar) : PChar;</i> Liefert den Textinhalt des Textfield .
j_getwidth	<i>function j_getwidth (obj: integer) : integer;</i> Liefert die Breite des Textfield obj zurück.
j_getxpos	<i>function j_getxpos (obj: integer) : integer;</i> Liefert die aktuelle horizontale Position des Textfield obj zurück.
j_getypos	<i>function j_getypos (obj: integer) : integer;</i> Liefert die aktuelle vertikale Position des Textfield obj zurück.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Zeigt das Textfield obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>function j_isparent (obj: integer; cont: integer) : integer;</i> Liefert true zurück, falls das cont die umgebende Komponente von obj ist, sonst false
j_isvisible	<i>function j_isvisible (obj: integer) : integer;</i> Liefert true zurück, falls obj sichtbar ist, sonst false
j_keylistener	<i>function j_keylistener (obj: integer) : integer;</i> Bindet an das Textfield obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) : integer;</i> Bindet an das Textfield obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>function j_popupmenu (obj: integer; label: PChar) : integer;</i>

	Erzeugt ein neues Popumenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (obj: integer) ;</i> Druckt den Inhalt des Textfield .
j_release	<i>procedure j_release (obj: integer) ;</i> Gibt das Textfield obj aus dem umgebenden Container Element frei.
j_selectall	<i>procedure j_selectall (obj: integer) ;</i> Selectiert den gesamten Textinhalt des Textfield .
j_selecttext	<i>procedure j_selecttext (obj: integer; start: integer; end: integer) ;</i> Selectiert den Text von Position start bis end .
j_setborderpos	<i>procedure j_setborderpos (obj: integer; pos: integer) ;</i> Positioniert das Textfield obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (obj: integer; r: integer; g: integer; b: integer) ;</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g: integer; b: integer) ;</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcolumns	<i>procedure j_setcolumns (obj: integer; columns: integer) ;</i> Setzt die Anzahl Spalten von obj auf columns .
j_setcurpos	<i>procedure j_setcurpos (obj: integer; pos: integer) ;</i> Setzt den Text Cursor auf die Position pos .
j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i> Setzt den Cursor des Textfield obj auf cursor .
j_setechochar	<i>procedure j_setechochar (obj: integer; chr: char) ;</i> Bestimmt das Zeichen chr als Echo Zeichen für alle Benutzereingaben in das Textfield .
j_seteditable	<i>procedure j_seteditable (obj: integer; bool: boolean) ;</i> Ist bool true , so ist das Textfield editierbar, ansonsten ist das Textfield nur lesbar für den Benutzer.
j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus dem Textfield obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp des Textfield obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font des Textfield obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße des Textfield obj auf size .

j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle des Textfield obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt das Textfield obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe des Textfield obj auf width and height .
j_settext	<i>procedure j_settext (obj: integer; str: PChar) ;</i> Ändert den Textinhalt des Textfield obj auf str
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt das Textfield obj auf dem Bildschirm an.

Vscrollbar

- j_vscrollbar** *function j_vscrollbar (obj: integer) : integer;*
Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt die Vscrollbar **obj** in den Container **cont** ein.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an die Vscrollbar **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled die Vscrollbar **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
die Vscrollbar **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled die Vscrollbar **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an die Vscrollbar **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der Vscrollbar **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*
Liefert die totale Höhe des aktuellen Fonts der Vscrollbar **obj** in Bildschirmpixeln zurück.
- j_getheight** *function j_getheight (obj: integer) : integer;*
Liefert die Höhe der Vscrollbar **obj** zurück.
- j_getparentid** *function j_getparentid (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *function j_getparent (obj: integer) : integer;*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstringwidth** *function j_getstringwidth (obj: integer; str: PChar) : integer;*
Liefert die Länge des **str** des aktuellen Fonts der Vscrollbar **obj** in Bildschirmpixeln zurück.

j_getvalue	<i>function j_getvalue (obj: integer) : integer;</i> Liefert den aktuellen Wert der Vscrollbar .
j_getwidth	<i>function j_getwidth (obj: integer) : integer;</i> Liefert die Breite der Vscrollbar obj zurück.
j_getxpos	<i>function j_getxpos (obj: integer) : integer;</i> Liefert die aktuelle horizontale Position der Vscrollbar obj zurück.
j_getypos	<i>function j_getypos (obj: integer) : integer;</i> Liefert die aktuelle vertikale Position der Vscrollbar obj zurück.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Zeigt die Vscrollbar obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>function j_isparent (obj: integer; cont: integer) : integer;</i> Liefert true zurück, falls das cont die umgebende Komponente von obj ist, sonst false
j_isvisible	<i>function j_isvisible (obj: integer) : integer;</i> Liefert true zurück, falls obj sichtbar ist, sonst false
j_keylistener	<i>function j_keylistener (obj: integer) : integer;</i> Bindet an die Vscrollbar obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) : integer;</i> Bindet an die Vscrollbar obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>function j_popupmenu (obj: integer; label: PChar) : integer;</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (obj: integer) ;</i> Druckt den Inhalt der Vscrollbar .
j_release	<i>procedure j_release (obj: integer) ;</i> Gibt die Vscrollbar obj aus dem umgebenden Container Element frei.
j_setblockinc	<i>function j_setblockinc (obj: integer; val: integer) : integer;</i> Stellt die Blockänderung von der Vscrollbar auf den Wert von val ein.
j_setborderpos	<i>procedure j_setborderpos (obj: integer; pos: integer) ;</i> Positioniert die Vscrollbar obj an die angegebene Position pos . Im umgebenden Container muß einen Borderlayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert

j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i> Setzt den Cursor der Vscrollbar obj auf cursor .
j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus der Vscrollbar obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp der Vscrollbar obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font der Vscrollbar obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße der Vscrollbar obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle der Vscrollbar obj auf style .
j_setmax	<i>function j_setmax (obj: integer; val: integer) : integer;</i> Stellt die obere Grenze der Vscrollbar auf den Wert von val ein.
j_setmin	<i>function j_setmin (obj: integer; val: integer) : integer;</i> Stellt die untere Grenze der Vscrollbar auf den Wert von val ein.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt die Vscrollbar obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe der Vscrollbar obj auf width and height .
j_setslidesize	<i>function j_setslidesize (obj: integer; val: integer) : integer;</i> Setzt die Größe des Schiebers auf den Wert von val .
j_setunitinc	<i>function j_setunitinc (obj: integer; val: integer) : integer;</i> Stellt die minimale Wertänderung von die Vscrollbar auf den Wert von val ein.
j_setvalue	<i>procedure j_setvalue (obj: integer; val: integer) ;</i> Stellt den aktuellen Wert der Vscrollbar obj auf val .
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt die Vscrollbar obj auf dem Bildschirm an.

Window

- j_window** *function j_window (obj: integer) : integer;*
Erzeugt ein neues Window und liefert dessen Eventnummer zurück.
- j_add** *procedure j_add (obj: integer; cont: integer) ;*
Fügt das Window **obj** in den Container **cont** ein.
- j_borderpanel** *function j_borderpanel (obj: integer; type: integer) : integer;*
Erzeugt ein neues Borderpanel und liefert dessen Eventnummer zurück.
- j_button** *function j_button (obj: integer; label: PChar) : integer;*
Erzeugt einen neuen Button mit dem Textinhalt **label**
- j_canvas** *function j_canvas (obj: integer; width: integer; height: integer) : integer;*
Erzeugt einen neuen Canvas mit der Breite **width** Pixel und einer Höhe von **height** Pixel.
- j_checkbox** *function j_checkbox (obj: integer; label: PChar) : integer;*
Erzeugt eine Checkbox mit dem Textinhalt **label** und liefert dessen Eventnummer zurück.
- j_choice** *function j_choice (obj: integer) : integer;*
Erzeugt eine neue Choice und gibt deren Eventnummer zurück.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) : integer;*
Bindet an das Window **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (obj: integer) ;*
disabled das Window **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (obj: integer) ;*
das Window **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (obj: integer) ;*
enabled das Window **obj**.
- j_focuslistener** *function j_focuslistener (obj: integer) : integer;*
Bindet an das Window **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent (obj: integer) : integer;*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Window **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *function j_getfontheight (obj: integer) : integer;*

	Liefert die totale Höhe des aktuellen Fonts des Window obj in Bildschirmpixeln zurück.
j_getheight	<i>function j_getheight (obj: integer) : integer;</i> Liefert die Höhe des Window obj zurück.
j_getinheight	<i>function j_getinheight (cont: integer) : integer;</i> Liefert die Höhe des Window obj des inneren Bereichs zurück.
j_getinsets	<i>function j_getinsets (obj: integer; side: integer) : integer;</i> Liefert die angeforderte Breite des inneren Randes dem Window zurück.
j_getinwidth	<i>function j_getinwidth (cont: integer) : integer;</i> Liefert die Breite des Window obj des inneren Bereichs zurück.
j_getlayoutid	<i>function j_getlayoutid (obj: integer) : integer;</i> Liefert die Eventnummer des Layoutmanagers des Containers obj zurück.
j_getparentid	<i>function j_getparentid (obj: integer) : integer;</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>function j_getparent (obj: integer) : integer;</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getstringwidth	<i>function j_getstringwidth (obj: integer; str: PChar) : integer;</i> Liefert die Länge des str des aktuellen Fonts des Window obj in Bildschirmpixeln zurück.
j_getwidth	<i>function j_getwidth (obj: integer) : integer;</i> Liefert die Breite des Window obj zurück.
j_getxpos	<i>function j_getxpos (obj: integer) : integer;</i> Liefert die aktuelle horizontale Position des Window obj zurück.
j_getypos	<i>function j_getypos (obj: integer) : integer;</i> Liefert die aktuelle vertikale Position des Window obj zurück.
j_graphicbutton	<i>function j_graphicbutton (obj: integer; filename: PChar) : integer;</i> Erzeugt einen neuen Graphicbutton mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_graphiclabel	<i>function j_graphiclabel (obj: integer; str: PChar) : integer;</i> Erzeugt ein neues Graphiclabel mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Zeigt das Window obj nicht mehr auf dem Bildschirm an.
j_hscrollbar	<i>function j_hscrollbar (obj: integer) : integer;</i> Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.

j_isparent	<i>function j_isparent (obj: integer; cont: integer) : integer;</i> Liefert true zurück, falls das cont die umgebende Komponente von obj ist, sonst false
j_isvisible	<i>function j_isvisible (obj: integer) : integer;</i> Liefert true zurück, falls obj sichtbar ist, sonst false
j_keylistener	<i>function j_keylistener (obj: integer) : integer;</i> Bindet an das Window obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_label	<i>function j_label (obj: integer; label: PChar) : integer;</i> Erzeugt ein neues Label Element mit dem Textinhalt label
j_led	<i>function j_led (obj: integer; style: integer; color: integer) : integer;</i> Erzeugt eine LED der Farbe color und der Form style
j_line	<i>function j_line (obj: integer; orient: integer; style: integer; length: integer) : integer;</i> Erzeugt ein neues Linienelement und liefert dessen Eventnummer zurück
j_list	<i>function j_list (obj: integer; rows: integer) : integer;</i> Erzeugt eine neue List und gibt deren Eventnummer zurück.
j_meter	<i>function j_meter (obj: integer; title: PChar) : integer;</i> Erzeugt ein analoges Anzeigeinstrument mit der Beschriftung title .
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) : integer;</i> Bindet an das Window obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_pack	<i>procedure j_pack (obj: integer) ;</i> Veranlaßt den Layoutmanager von obj die minimale Größe zu berechnen.
j_panel	<i>function j_panel (obj: integer) : integer;</i> Erzeugt ein neues panel und liefert dessen Eventnummer zurück.
j_popupmenu	<i>function j_popupmenu (obj: integer; label: PChar) : integer;</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (obj: integer) ;</i> Druckt den Inhalt des Window .
j_progressbar	<i>function j_progressbar (obj: integer; orient: integer) : integer;</i> Erzeugt eine Progress Bar mit der Ausrichtung orient .
j_radiogroup	<i>function j_radiogroup (obj: integer) : integer;</i> Erzeugt eine neue Radiogroup und liefert dessen Eventnummer zurück.
j_releaseall	<i>procedure j_releaseall (obj: integer) ;</i> Gibt alle Elemente aus dem Window obj frei.

j_release	<i>procedure j_release (obj: integer) ;</i> Gibt das Window obj aus dem umgebenden Container Element frei.
j_scrollpane	<i>function j_scrollpane (obj: integer) : integer;</i> Erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück.
j_setalign	<i>procedure j_setalign (obj: integer; align: integer) ;</i> Setzt die Ausrichtung innerhalb des Window obj auf align . Ein Flowlayout Manager wird vorausgesetzt.
j_setborderlayout	<i>procedure j_setborderlayout (obj: integer) ;</i> Integriert einen BorderLayout Manager in das Window obj .
j_setborderpos	<i>procedure j_setborderpos (obj: integer; pos: integer) ;</i> Positioniert das Window obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) : integer;</i> Setzt den Cursor des Window obj auf cursor .
j_setfixlayout	<i>procedure j_setfixlayout (obj: integer) ;</i> Integriert einen Fixlayout Manager in das Window obj (voreingestellte Layoutmanager).
j_setflowfill	<i>procedure j_setflowfill (obj: integer; bool: boolean) ;</i> Setzt die Höhe (Breite) aller enthaltenen Komponenten auf die Höhe (Breite) von dem Window obj . Ein Flowlayout Manager wird vorausgesetzt.
j_setflowlayout	<i>procedure j_setflowlayout (obj: integer; align: integer) ;</i> Integriert einen Flowlayout Manager in das Window obj mit der Ausrichtung align .
j_setfocus	<i>function j_setfocus (obj: integer) : integer;</i> Weist den Input Focus dem Window obj zu.
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Setzt den Fonttyp des Window obj auf name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Setzt den Font des Window obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Setzt den Fontgröße des Window obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Setzt den Fontstyle des Window obj auf style .

j_setgridlayout	<i>procedure j_setgridlayout (obj: integer; row: integer; col: integer) ;</i> Integriert einen Gridlayout Manager in das Window obj mit row Reihen und col Spalten.
j_sethgap	<i>procedure j_sethgap (obj: integer; hgap: integer) ;</i> Setzt den horizontalen Abstand innerhalb des Window obj auf hgap Pixel.
j_setinsets	<i>procedure j_setinsets (obj: integer; top: integer; bottom: integer; left: integer; right: integer) ;</i> Setzt den inneren Rahmen auf die angegebenen Größen.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setnolayout	<i>procedure j_setnolayout (obj: integer) ;</i> Entfernt den aktuellen Layout Manager aus dem Window obj .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Verschiebt das Window obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Setzt Breite und Höhe des Window obj auf width and height .
j_setvgap	<i>procedure j_setvgap (obj: integer; vgap: integer) ;</i> Setzt den vertikalen Abstand innerhalb des Window obj auf vgap Pixel.
j_sevensegment	<i>function j_sevensegment (obj: integer; color: integer) : integer;</i> Erzeugt eine Siebensegment Anzeige der Farbe color .
j_show	<i>procedure j_show (obj: integer) ;</i> Zeigt das Window obj auf dem Bildschirm an.
j_textarea	<i>function j_textarea (obj: integer; rows: integer; columns: integer) : integer;</i> Erzeugt eine neue Textarea und gibt dessen Eventnummer zurück.
j_textfield	<i>function j_textfield (obj: integer; columns: integer) : integer;</i> Erzeugt eine neues Textfield und gibt dessen Eventnummer zurück.
j_vscrollbar	<i>function j_vscrollbar (obj: integer) : integer;</i> Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.
j_windowlistener	<i>function j_windowlistener (window: integer; kind: integer) : integer;</i> Bindet an das Window window einen neuen Windowlistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.

Kapitel 2

Functions

additem

Synopsis procedure **j_additem** (obj: integer; str: PChar) ;

Argumente obj integer
 str PChar

Beschreibung Diese Prozedur hängt ein neues Item mit dem Inhalt **str** an die Auswahlliste an. Es wird keine Eventnummer vergeben, statt dessen werden alle Item intern durchnummeriert, beginnend mit 0. Die Zielkomponente erzeugt bei einer Auswahl einen Event. Welches Item ausgewählt wurde, kann mit der Funktion *j_getselect()* ermittelt werden.

Zielkomponenten List, Choice

Beispiel

```
:  
list := j_list(frame,3);  
j_additem(list,'Eintrag 1');  
j_additem(list,'Eintrag 2');  
:
```



add

Synopsis	procedure j_add (obj: integer; cont: integer) ;
Argumente	obj integer cont integer
Beschreibung	Fügt die Komponente obj in den Container cont ein.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment

alertbox

Synopsis `procedure j_alertbox (obj: integer; title: PChar; text: PChar;
 button: PChar) ;`

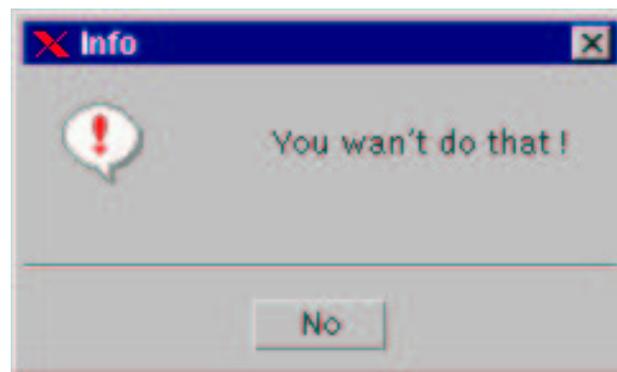
Argumente `obj integer
 title PChar
 text PChar
 button PChar`

Beschreibung Öffnet eine Alertbox mit der Überschrift **title**, Textinhalt **text** und einem Button. Eine Alertbox ist modal, dh. die Applikation wird solange blockiert, bis der Button oder das Closeicon angeklickt wird. Der Rückgabewert der Funktion beträgt 0 falls das Closeicon geklickt wird, wird der Button geklickt, so wird 1 zurückgegeben.

Zielkomponenten `Frame`

Beispiel

```
:  
retval := j_alertbox(frame,'Info','You wan't do that !',' No  ');  
:
```



appendtext

Synopsis procedure **j_appendtext** (obj: integer; text: PChar) ;

Argumente obj integer
 text PChar

Beschreibung Hängt den Text **text** am Ende von **obj** an.

Zielkomponenten Textarea

beep

Synopsis `procedure j_beep ;`

Beschreibung Es ertönt ein Systembeep.

borderpanel

Synopsis function **j_borderpanel** (obj: integer; type: integer) : integer;

Argumente obj integer
 type integer

Beschreibung Diese Funktion erzeugt ein neues Borderpanel und liefert dessen Eventnummer zurück. Im Fehlerfall wird -1 zurückgegeben. Das Panel wird der Komponente zugeordnet, die durch **obj** referenziert wird. Das Borderpanel ist funktionell identisch mit einem normalen Panel, es enthält lediglich einen sichtbaren Rahmen, der die enthaltenen Komponenten auch optisch gruppiert. Der Stil des Rahmens kann aus vier Stiltypen gewählt werden:

- J_LINEDOWN : Der Rahmen besteht aus einer Linie, die optisch abgesenkt erscheint.
- J_LINEUP : Der Rahmen besteht aus einer Linie, die optisch hervorzutreten scheint.
- J_AREADOWN : Das Panel scheint optisch abgesenkt.
- J_AREAUP : Das Panel scheint optisch hervorzutreten.

Ein Borderpanel liefert bei einer Größenänderung einen Event.

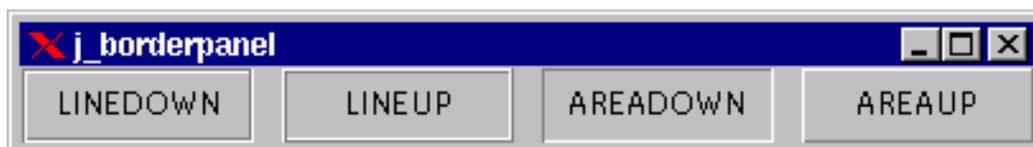
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
j_setgridlayout(frame,1,4);
p1 := j_borderpanel(frame,J_LINEDOWN);
p2 := j_borderpanel(frame,J_LINEUP);
p3 := j_borderpanel(frame,J_AREADOWN);
p4 := j_borderpanel(frame,J_AREAUP);
:

```



button

Synopsis function **j_button** (obj: integer; label: PChar) : integer;

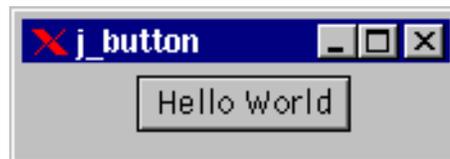
Argumente obj integer
 label PChar

Beschreibung Die Funktion erzeugt einen neuen Button mit dem Textinhalt **label**. Die Funktion liefert die Eventnummer des Buttons zurück, oder im Fehlerfall -1 . Ein Event wird erzeugt, wenn der Benutzer mit der Maus den Button betätigt.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:  
frame := j_frame('j_button');  
button := j_button(frame, 'Hello World');  
:
```



canvas

Synopsis function **j_canvas** (obj: integer; width: integer; height: integer) : integer;

Argumente obj integer
 width integer
 height integer

Beschreibung Diese Funktion erzeugt einen neuen Canvas mit der Breite **width** Pixel und einer Höhe von **height** Pixel. Ein Canvas (Leinwand) stellt eine universelle Zeichenfläche zur Verfügung. Die Funktion liefert die Eventnummer des Canvas zurück, oder im Fehlerfall -1 . Ein Event wird erzeugt, wenn sich die Größe des Canvas verändert.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
canvas := j_canvas(frame,200,50);
j_setnamedcolorbg(canvas,J_RED);
:

```



checkbox

Synopsis `function j_checkbox (obj: integer; label: PChar) : integer;`

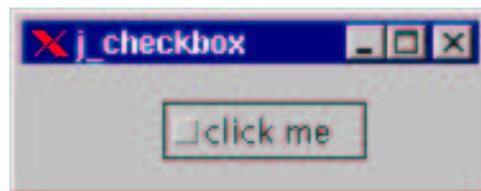
Argumente `obj` `integer`
 `label` `PChar`

Beschreibung Diese Funktion erzeugt eine Checkbox mit dem Textinhalt **label** und liefert dessen Eventnummer zurück. Im Fehlerfall wird `-1` zurückgegeben. Eine Checkbox ist ein graphisches Element, das zwei Zustände annehmen kann. Sie ist entweder gesetzt oder nicht gesetzt. Checkboxes bestehen intern aus zwei Elementen, einem Label, und einem kleinen graphischen Element, das den Zustand der Checkbox anzeigt. Eine Checkbox liefert immer dann einen Event, wenn sich ihr Zustand ändert.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:  
frame    := j_frame('j_checkbox');  
checkbox  := j_checkbox(frame,'click me');  
:
```



checkmenuItem

Synopsis function **j_checkmenuItem** (obj: integer; label: PChar) :
integer;

Argumente obj integer
 label PChar

Beschreibung Die Funktion erzeugt ein CheckmenuItem mit der Beschriftung **label** in der Komponente, die durch **obj** referenziert wird, und liefert dessen Eventnummer zurück. Eine CheckmenuItem erzeugt einen Event, wenn es durch einen Mausklick angewählt wurde. Ein CheckmenuItem kann zwei Zustände annehmen (selected / not selected), die durch ein kleine Icons dargestellt werden. Der Zustand eines Checkmenüitems kann mit der Funktion **j_getstate()** abgefragt werden. Als Zielkomponenten sind nur Menu, Popumenu und Helpmenu zulässig.

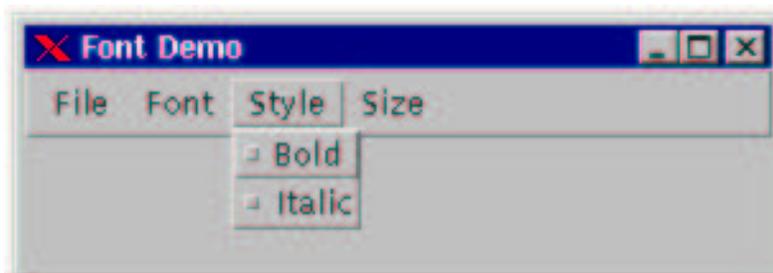
Zielkomponenten Menu, Popumenu, Helpmenu

Beispiel

```

:
menubar := j_menubar(frame)
:
style := j_menu(menubar, 'Style');
bold  := j_checkmenuItem(style, 'Bold');
italic:= j_checkmenuItem(style, 'Italic');
:

```



checkbox2

Synopsis `procedure j_checkbox2 (obj: integer; title: PChar; text: PChar; button1: PChar; button2: PChar);`

Argumente `obj integer`
 `title PChar`
 `text PChar`
 `button1 PChar`
 `button2 PChar`

Beschreibung Öffnet eine Choicebox mit der Überschrift **title**, Textinhalt **text** und zwei Buttons. Eine Choicebox ist modal, dh. die Applikation wird solange blockiert, bis ein Button oder das Closeicon angeklickt wird. Der Focus ist auf den ersten Button gesetzt. Der Rückgabewert der Funktion beträgt 0 falls das Closeicon geklickt wird, 1 wird Button1 geklickt und 2 bei Button2.

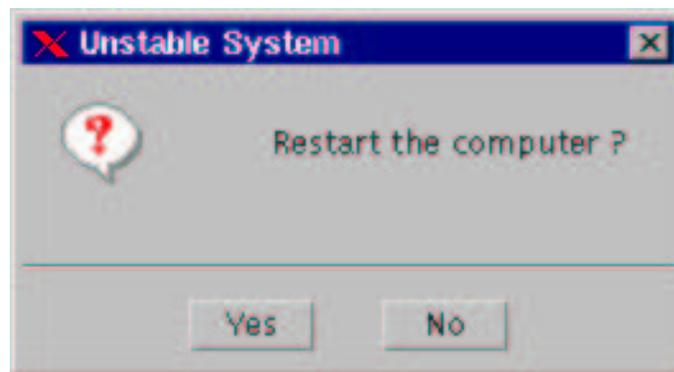
Zielkomponenten `Frame`

Beispiel

```

:
retval := j_checkbox2(frame,'Unstable System','Restart the computer ?',
                        'Yes','No');
:

```



checkbox3

Synopsis `procedure j_checkbox3 (obj: integer; title: PChar; text: PChar; button1: PChar; button2: PChar; button3: PChar) ;`

Argumente

obj	integer
title	PChar
text	PChar
button1	PChar
button2	PChar
button3	PChar

Beschreibung

Öffnet eine Choicebox mit der Überschrift **title**, Textinhalt **text** und drei Buttons. Eine Choicebox ist modal, dh. die Applikation wird solange blockiert, bis ein Button oder das Closeicon angeklickt wird. Der Focus ist auf den ersten Button gesetzt. Der Rückgabewert der Funktion beträgt 0 falls das Closeicon geklickt wird, 1 bei Button1, 2 bei Button2 und 3 bei Button3.

Zielkomponenten Frame

Beispiel

```

:
retval := j_checkbox2(frame,'Viruswarning ??','I love you',
                    'Ups','Arrrg','Cancel');
:

```



choice

Synopsis `function j_choice (obj: integer) : integer;`

Argumente `obj` `integer`

Beschreibung Diese Funktion erzeugt eine neue Choice und gibt deren Eventnummer zurück. Im Fehlerfall wird `-1` zurückgegeben. Eine Choice bietet dem Benutzer die Auswahl zwischen mehreren vordefinierten Einträgen. Dabei befindet sich auf der Oberfläche zunächst ein Element das einem Button ähnlich sieht. Wird es angeklickt, so klappt ein Menu heraus, aus dem man ein Element auswählen kann. Wird ein Element ausgewählt, so erzeugt die Choice einen Event.

Zielkomponenten `Panel, Borderpanel, Window, Dialog, Frame`

Beispiel

```
:  
choice := j_choice(frame);  
j_additem(choice, 'Eintrag 1');  
j_additem(choice, 'Eintrag 2');  
:
```



cliprect

Synopsis	procedure j_cliprect (obj: integer; x: integer; y: integer; width: integer; height: integer) ;
Argumente	obj integer x integer y integer width integer height integer
Beschreibung	Setzt den Clipping Bereich auf das spezifizierte Rechteck (x , y , width , height).
Zielkomponenten	Canvas, Image, Printer

componentlistener

Synopsis	function j_componentlistener (obj: integer; kind: integer) : integer;
Argumente	obj integer kind integer
Beschreibung	Bindet an die Komponente obj einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird. Zulässige Werte für kind sind: <ul style="list-style-type: none">• J_RESIZED : Der Componentlistener liefert einen Event, wenn die Komponente obj die Größe ändert.• J_HIDDEN : Der Componentlistener liefert einen Event, wenn die Komponente obj verdeckt wird.• J_SHOWN : Der Componentlistener liefert einen Event, wenn die Komponente obj sichtbar wird.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment

connect

Synopsis `function j_connect (hostname: PChar) : boolean;`

Argumente `hostname PChar`

Beschreibung Diese Funktion stellt eine Verbindung zu einem laufenden JAPI Kernel her. Der Kernel wird auf dem Rechner **hostname** gesucht. **hostname** kann einen Namen oder eine IP Adresse beinhalten. Auf diese Weise kann eine Ausgabeumlenkung erfolgen, dh. Applikation und Oberfläche laufen auf verschiedenen Hosts. Schlägt die Verbindung fehl, so wird als Fehlercode `false` zurückgegeben

Beispiel

```
:
if( not j_connect('atan.japi.de') ) then

    or

if( not j_connect('127.0.0.1')) then
:
```

delete

Synopsis `procedure j_delete (obj: integer; start: integer; end: integer) ;`

Argumente `obj integer`
 `start integer`
 `end integer`

Beschreibung löscht den Text von Position **start** bis Position **end**.

Zielkomponenten Textarea

deselect

Synopsis function **j_deselect** (obj: integer; item: integer) : integer;

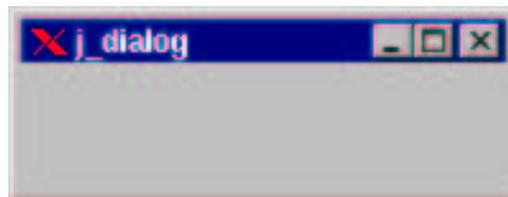
Argumente obj integer
 item integer

Beschreibung Deselectiert das Item mit dem Index **item**.

Zielkomponenten List

dialog

Synopsis	function j_dialog (obj: integer; label: PChar) : integer;
Argumente	obj integer label PChar
Beschreibung	Dies Funktion erzeugt ein neues Dialogwindow mit dem Titel label und liefert dessen Eventnummer zurück. Im Fehlerfall wird -1 zurückgeliefert. Ein Dialogfenster ist ein eigenständiges Fenster, ebenso wie ein Frame. Allerdings kann ein Dialogfenster keine Menüleiste besitzen. Als Zielobjekt dieser Funktion ist nur ein Frame zulässig. Ein Dialog wird erst angezeigt, wenn ein entsprechendes j_show() Kommando ausgeführt wurde. Ein Dialog erzeugt einen Event, wenn das Close Icon im Fensterrahmen angeklickt wird.
Zielkomponenten	Frame
Beispiel	: dialog := j_dialog(frame,'j_dialog'); j_setsize(dialog,200,80); j_show(dialog); :



disable

Synopsis	procedure j_disable (obj: integer) ;
Argumente	obj integer
Beschreibung	disabled die Komponente obj , sodaß diese auf keine Benutzereingaben mehr reagiert.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu

dispose

Synopsis	procedure j_dispose (obj: integer) ;
Argumente	obj integer
Beschreibung	die Komponente obj wird mit allen Recourcen gelöscht.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, Canvas, Image, Printer, Keylistener, Focuslistener, Mousetlistener

drawarc

Synopsis `procedure j_drawarc (obj: integer; x: integer; y: integer; rx: integer; ry: integer; arc1: integer; arc2: integer) ;`

Argumente

<code>obj</code>	<code>integer</code>
<code>x</code>	<code>integer</code>
<code>y</code>	<code>integer</code>
<code>rx</code>	<code>integer</code>
<code>ry</code>	<code>integer</code>
<code>arc1</code>	<code>integer</code>
<code>arc2</code>	<code>integer</code>

Beschreibung Zeichnet einen Kreisbogen mit dem Mittelpunkt **(x,y)** und den Hauptradien **rx** und **ry**. Die Parameter **arc1** und **arc2** bestimmen den Start- bzw. den Endwinkel des Kreisbogens. Bezugswert ist die Horizontale, von der die Winkel entgegen des Uhrzeigersinns abgetragen werden. Die Winkel sind im Bogenmaß anzugeben. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

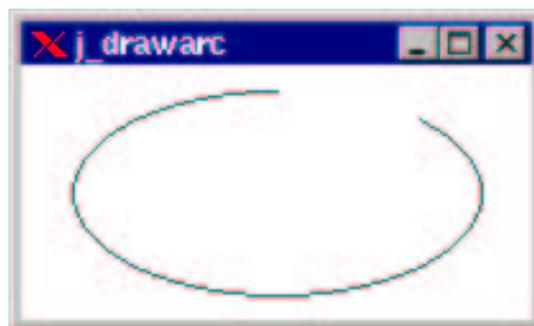
Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
canvas := j_canvas(frame,200,100);
j_drawarc(canvas,100,50,80,40,45,-270);
:

```



drawcircle

Synopsis procedure **j_drawcircle** (obj: integer; x: integer; y: integer; r:
integer) ;

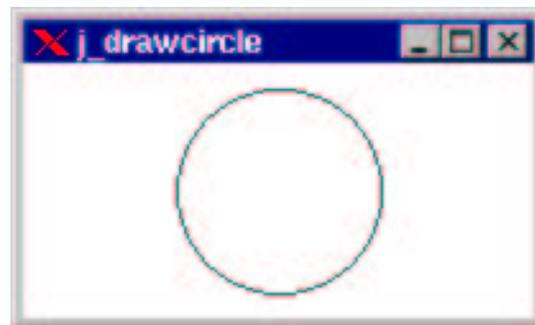
Argumente obj integer
 x integer
 y integer
 r integer

Beschreibung Zeichnet einen Kreis mit dem Mittelpunkt (**x,y**) und dem Radius
r. Nullpunkt des Koordinatensystems ist die linke obere Ecke der
Komponenten, die durch **obj** referenziert wird.

Zielkomponenten Canvas, Image, Printer

Beispiel

```
:  
canvas := j_canvas(frame,200,100);  
j_drawcircle(canvas,100,50,40);  
:
```



drawimagesource

Synopsis	<pre>procedure j_drawimagesource (obj: integer; x: integer; y: integer; w: integer; h: integer; r: var integer; g: var integer; b: var integer) ;</pre>																
Argumente	<table> <tr><td>obj</td><td>integer</td></tr> <tr><td>x</td><td>integer</td></tr> <tr><td>y</td><td>integer</td></tr> <tr><td>w</td><td>integer</td></tr> <tr><td>h</td><td>integer</td></tr> <tr><td>r</td><td>var integer (call by reference of the 1st array element)</td></tr> <tr><td>g</td><td>var integer (call by reference of the 1st array element)</td></tr> <tr><td>b</td><td>var integer (call by reference of the 1st array element)</td></tr> </table>	obj	integer	x	integer	y	integer	w	integer	h	integer	r	var integer (call by reference of the 1st array element)	g	var integer (call by reference of the 1st array element)	b	var integer (call by reference of the 1st array element)
obj	integer																
x	integer																
y	integer																
w	integer																
h	integer																
r	var integer (call by reference of the 1st array element)																
g	var integer (call by reference of the 1st array element)																
b	var integer (call by reference of the 1st array element)																
Beschreibung	<p>Die Funktion stellt ein Bild an der Position (x,y) mit der Breite w und der Höhe h dar. Die Bilddaten werden als rot-, grün- und blau- Kanal in den Arrays r,g,b übergeben.</p>																
Zielkomponenten	Canvas, Image, Printer																

drawimage

Synopsis	procedure j_drawimage (obj: integer; image: integer; x: integer; y: integer) ;
Argumente	obj integer image integer x integer y integer
Beschreibung	Kopiert das Image mit der Eventnummer image an die Position (x , y).
Zielkomponenten	Canvas, Image, Printer

drawline

Synopsis `procedure j_drawline (obj: integer; x1: integer; y1: integer; x2: integer; y2: integer) ;`

Argumente

<code>obj</code>	<code>integer</code>
<code>x1</code>	<code>integer</code>
<code>y1</code>	<code>integer</code>
<code>x2</code>	<code>integer</code>
<code>y2</code>	<code>integer</code>

Beschreibung Zeichnet eine Linie in der momentan gesetzten Vordergrundfarbe von Position **(x1,y1)** nach Position **(x2,y2)**. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
canvas := j_canvas(frame,256,50);
j_drawline(canvas,0,0,256,50);
:

```



drawoval

Synopsis procedure **j_drawoval** (obj: integer; x: integer; y: integer; rx:
integer; ry: integer) ;

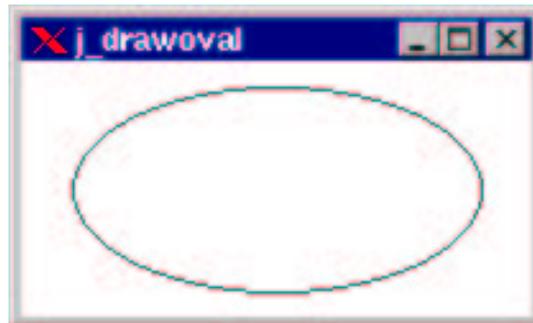
Argumente obj integer
 x integer
 y integer
 rx integer
 ry integer

Beschreibung Zeichnet einen Oval mit dem Mittelpunkt (**x,y**) und den Haupttra-
dien **rx** und **ry** in der momentan gesetzten Vordergrundfarbe.
Nullpunkt des Koordinatensystems ist die linke obere Ecke der
Komponenten, die durch **obj** referenziert wird.

Zielkomponenten Canvas, Image, Printer

Beispiel

```
:  
canvas := j_canvas(frame,200,100);  
j_drawoval(canvas,100,50,80,40);  
:
```



drawpixel

Synopsis `procedure j_drawpixel (obj: integer; x: integer; y: integer) ;`

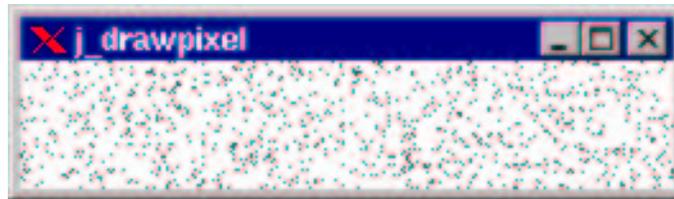
Argumente `obj integer`
 `x integer`
 `y integer`

Beschreibung Zeichnet einen Pixel in der momentan gesetzten Vordergrundfarbe an die Position **(x,y)**. Bezugspunkt ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

Zielkomponenten Canvas, Image, Printer

Beispiel

```
:  
canvas := j_canvas(frame,256,50);  
for i:=0 to 1000 do  
    j_drawpixel(canvas,j_random mod 256,j_random mod 50);  
:
```



drawpolygon

Synopsis procedure **j_drawpolygon** (obj: integer; len: integer; x: var integer; y: var integer) ;

Argumente obj integer
 len integer
 x var integer (call by reference of the 1st array element)
 y var integer (call by reference of the 1st array element)

Beschreibung Zeichnet eine Polylinie in der momentan gesetzten Vordergrundfarbe durch die Punkte, die durch die Arrays **x** und **y** gegeben sind. Beide Arrays müssen die mindestens **len** Elemente besitzen. Das Polygon wird automatisch geschlossen. Daher müssen die erste und letzte Koordinate nicht übereinstimmen. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

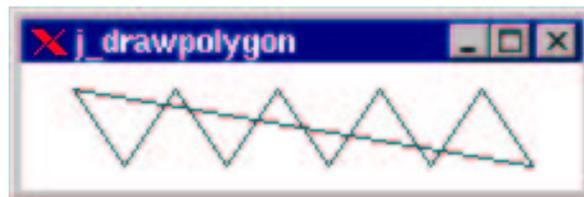
Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
x : array [0..9] of int =
    (20,40,60,80,100,120,140,160,180,200);
y : array [0..9] of int =
    (10,40,10,40,10,40,10,40,10,40);
canvas := j_canvas(frame,256,50);
j_drawpolygon(canvas,10,x[0],y[0]);
:

```



drawpolyline

Synopsis procedure **j_drawpolyline** (obj: integer; len: integer; x: var integer; y: var integer) ;

Argumente

obj	integer
len	integer
x	var integer (call by reference of the 1st array element)
y	var integer (call by reference of the 1st array element)

Beschreibung Zeichnet eine Polylinie in der momentan gesetzten Vordergrundfarbe durch die Punkte, die durch die Arrays **x** und **y** gegeben sind. Beide Arrays müssen die mindestens **len** Elemente besitzen. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

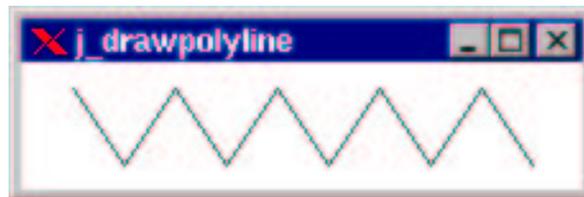
Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
x : array [0..9] of int =
    (20,40,60,80,100,120,140,160,180,200);
y : array [0..9] of int =
    (10,40,10,40,10,40,10,40,10,40);
canvas := j_canvas(frame,256,50);
j_drawpolyline(canvas,10,x[0],y[0]);
:

```



drawrect

Synopsis procedure **j_drawrect** (obj: integer; x: integer; y: integer;
 width: integer; height: integer);

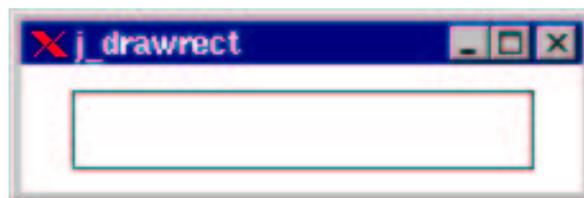
Argumente obj integer
 x integer
 y integer
 width integer
 height integer

Beschreibung Zeichnet eine Rechteck in der momentan gesetzten Vordergrund-
 farbe an die Position (**x,y**) mit der Breite **width** und der Höhe
 height. Bezugspunkt des Rechtecks ist die linke obere Ecke. Null-
 punkt des Koordinatensystems ist die linke obere Ecke der Kom-
 ponenten, die durch **obj** referenziert wird.

Zielkomponenten Canvas, Image, Printer

Beispiel

```
:  
canvas := j_canvas(frame,220,50);  
j_drawrect(canvas,20,10,180,30);  
:
```



drawroundrect

Synopsis `procedure j_drawroundrect (obj: integer; x: integer; y: integer; width: integer; height: integer; arcx: integer; arcy: integer)`
`;`

Argumente

<code>obj</code>	<code>integer</code>
<code>x</code>	<code>integer</code>
<code>y</code>	<code>integer</code>
<code>width</code>	<code>integer</code>
<code>height</code>	<code>integer</code>
<code>arcx</code>	<code>integer</code>
<code>arcy</code>	<code>integer</code>

Beschreibung

Zeichnet ein Rechteck mit abgerundeten Ecken an die Position **(x,y)**. Das Rechteck hat eine Breite von **width** und eine Höhe von **height** Pixeln. Die Form der runden Ecken wird durch die Parameter **arcx** und **arcy** bestimmt. Diese Parameter bezeichnen die Radien der Rundungen in horizontaler bzw. vertikaler Richtung.

Bezugspunkt des Rechtecks ist die linke obere Ecke. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

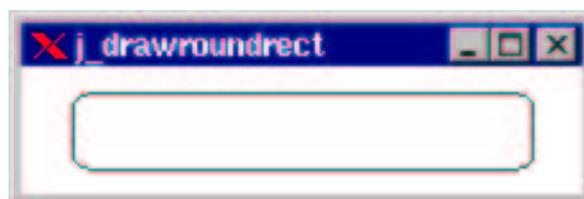
Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
canvas := j_canvas(frame,220,50);
j_drawroundrect(canvas,20,10,180,30,10,5);
:

```



drawscaledimage

Synopsis	<pre>procedure j_drawscaledimage (obj: integer; image: integer; sx: integer; sy: integer; sw: integer; sh: integer; tx: integer; ty: integer; tw: integer; th: integer) ;</pre>																				
Argumente	<table><tr><td>obj</td><td>integer</td></tr><tr><td>image</td><td>integer</td></tr><tr><td>sx</td><td>integer</td></tr><tr><td>sy</td><td>integer</td></tr><tr><td>sw</td><td>integer</td></tr><tr><td>sh</td><td>integer</td></tr><tr><td>tx</td><td>integer</td></tr><tr><td>ty</td><td>integer</td></tr><tr><td>tw</td><td>integer</td></tr><tr><td>th</td><td>integer</td></tr></table>	obj	integer	image	integer	sx	integer	sy	integer	sw	integer	sh	integer	tx	integer	ty	integer	tw	integer	th	integer
obj	integer																				
image	integer																				
sx	integer																				
sy	integer																				
sw	integer																				
sh	integer																				
tx	integer																				
ty	integer																				
tw	integer																				
th	integer																				
Beschreibung	Kopiert einen Ausschnitt des Bildes image , der durch die Parameter (sx , sy , sw , sh) bestimmt wird an die Position (tx , ty) und scaliert den Ausschnitt auf die Breite tw und die Höhe th																				
Zielkomponenten	Canvas, Image, Printer																				

drawstring

Synopsis procedure **j_drawstring** (obj: integer; x: integer; y: integer;
str: PChar) ;

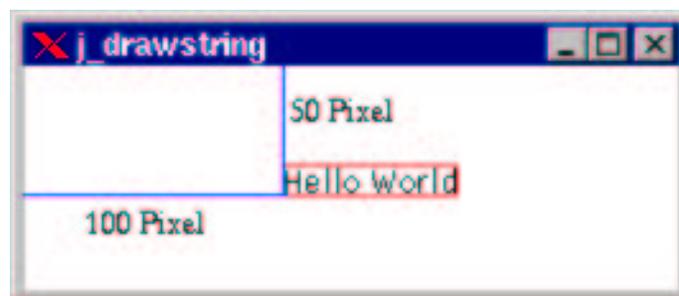
Argumente obj integer
 x integer
 y integer
 str PChar

Beschreibung Die Prozedur zeichnet einen String an die Position (**x,y**). Die-
durch **obj** referenzierte Komponente muss vom Typ Drawable sein
(siehe Zielkomponenten). Die übergebenen Koordinaten beziehen
sich auf die linke untere Ecke des Rechtecks, welches sich gedank-
lich um den gezeichneten String legen läßt.

Zielkomponenten Canvas, Image, Printer

Beispiel

```
:
j_drawstring(canvas,100,100,'Hello World');
:
```

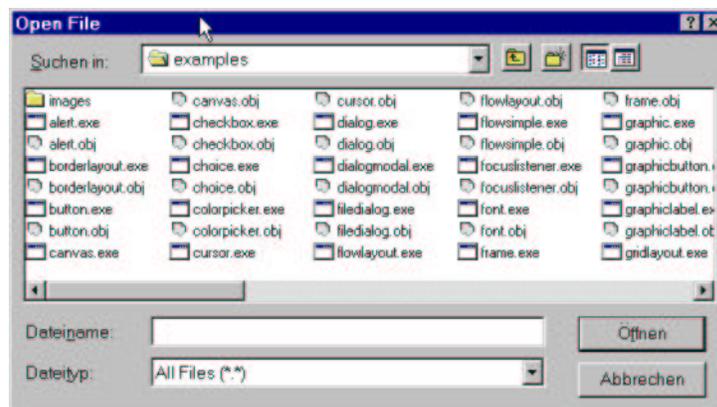


enable

Synopsis	procedure j_enable (obj: integer) ;
Argumente	obj integer
Beschreibung	enabled die Komponente obj .
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu

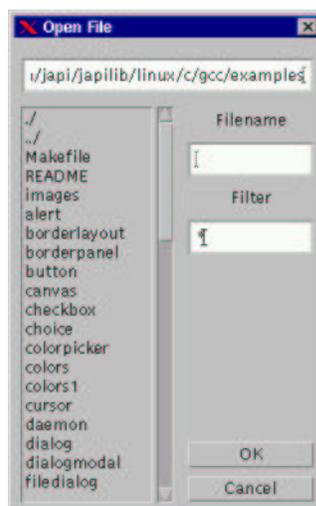
filedialog

Synopsis	function j_filedialog (frame: integer; title: PChar; directory: PChar; filename: PChar) : PChar;								
Argumente	<table border="0"> <tr><td>frame</td><td>integer</td></tr> <tr><td>title</td><td>PChar</td></tr> <tr><td>directory</td><td>PChar</td></tr> <tr><td>filename</td><td>var PChar</td></tr> </table>	frame	integer	title	PChar	directory	PChar	filename	var PChar
frame	integer								
title	PChar								
directory	PChar								
filename	var PChar								
Beschreibung	Öffnet eine Filedialogbox in dem angegebenen directory mit dem Title title und liefert den ausgewählten Dateiname in filename zurück. Ist "/S" im Titel enthalten wird die SAVE-Filedialogbox aufgerufen. Der Teilstring "/S" wird aus dem Titelstring entfernt.								
Zielkomponenten	Frame								
Beispiel	<pre> : filename := j_filedialog(frame,'Save/S File','..',filename); : </pre>								



fileselect

Synopsis	function j_fileselect (frame: integer; title: PChar; filter: PChar; filename: PChar) : PChar;								
Argumente	<table border="0" style="margin-left: 20px;"> <tr><td>frame</td><td>integer</td></tr> <tr><td>title</td><td>PChar</td></tr> <tr><td>filter</td><td>PChar</td></tr> <tr><td>filename</td><td>var PChar</td></tr> </table>	frame	integer	title	PChar	filter	PChar	filename	var PChar
frame	integer								
title	PChar								
filter	PChar								
filename	var PChar								
Beschreibung	Öffnet eine Fileselectorbox mit dem voreingestellten filename und dem Title title und liefert den ausgewählten Dateiname in filename zurück. Im Gegensatz zum Filedialog kann ein filter zur Dateiauswahl angegeben werden. Da der Fileselector auf Applikationsebene realisiert ist, kann er auch bei einer Ausgabeumlenkung verwendet werden.								
Zielkomponenten	Frame								
Beispiel	<pre> : filename := j_fileselect(frame,'Open File','*',filename); : </pre>								



fillarc

Synopsis `procedure j_fillarc (obj: integer; x: integer; y: integer; rx: integer; ry: integer; arc1: integer; arc2: integer) ;`

Argumente

<code>obj</code>	<code>integer</code>
<code>x</code>	<code>integer</code>
<code>y</code>	<code>integer</code>
<code>rx</code>	<code>integer</code>
<code>ry</code>	<code>integer</code>
<code>arc1</code>	<code>integer</code>
<code>arc2</code>	<code>integer</code>

Beschreibung

Zeichnet einen gefüllten Kreisbogen mit dem Mittelpunkt **(x,y)** und den Haupttradien **rx** und **ry**. Die Parameter **arc1** und **arc2** bestimmen den Start- bzw. den Endwinkel des Kreisbogens. Bezugswert ist die Horizontale von der die Winkel entgegen des Uhrzeigersinns abgetragen werden. Die Winkel sind im Bogenmaß anzugeben.

Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponente die durch **obj** referenziert wird.

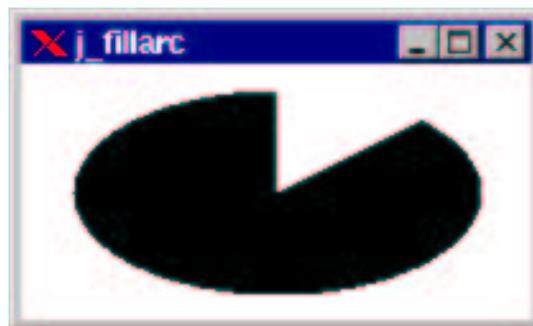
Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
canvas := j_canvas(frame,200,100);
j_fillarc(canvas,100,50,80,40,45,-270);
:

```



fillcircle

Synopsis procedure **j_fillcircle** (obj: integer; x: integer; y: integer; r:
integer) ;

Argumente obj integer
 x integer
 y integer
 r integer

Beschreibung Zeichnet einen gefülltes Kreis mit dem Mittelpunkt **(x,y)** und dem
Radius **r**. Nullpunkt des Koordinatensystems ist die linke obere
Ecke der Komponenten, die durch **obj** referenziert wird.

Zielkomponenten Canvas, Image, Printer

Beispiel

```
:  
canvas := j_canvas(frame,200,100);  
j_fillcircle(canvas,100,50,40);  
:
```



filloval

Synopsis procedure **j_filloval** (obj: integer; x: integer; y: integer; rx: integer; ry: integer) ;

Argumente

obj	integer
x	integer
y	integer
rx	integer
ry	integer

Beschreibung Zeichnet einen gefüllten Oval mit dem Mittelpunkt **(x,y)** und den Hauptradien **rx** und **ry** in der momentan gesetzten Vordergrundfarbe. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
canvas := j_canvas(frame,200,100);
j_filloval(canvas,100,50,80,40);
:

```



fillpolygon

Synopsis `procedure j_fillpolygon (obj: integer; len: integer; x: var integer; y: var integer) ;`

Argumente

obj	integer
len	integer
x	var integer (call by reference of the 1st array element)
y	var integer (call by reference of the 1st array element)

Beschreibung Zeichnet eine gefülltes Polygon in der momentan gesetzten Vordergrundfarbe durch die Punkte, die durch die Arrays **x** und **y** gegeben sind. Beide Arrays müssen die mindestens **len** Elemente besitzen. Das Polygon wird automatisch geschlossen. Daher müssen die erste und letzte Koordinate nicht übereinstimmen. Eingeschlossene Bereiche werden gefüllt dargestellt. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponente die durch **obj** referenziert wird.

Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
x : array [0..9] of int =
    (20,40,60,80,100,120,140,160,180,200);
y : array [0..9] of int =
    (10,40,10,40,10,40,10,40,10,40);
canvas := j_canvas(frame,256,50);
j_fillpolygon(canvas,10,x[0],y[0]);
:

```



fillrect

Synopsis `procedure j_fillrect (obj: integer; x: integer; y: integer; width: integer; height: integer) ;`

Argumente

<code>obj</code>	<code>integer</code>
<code>x</code>	<code>integer</code>
<code>y</code>	<code>integer</code>
<code>width</code>	<code>integer</code>
<code>height</code>	<code>integer</code>

Beschreibung Zeichnet eine gefülltes Rechteck in der momentan gesetzten Vordergrundfarbe an die Position **(x,y)** mit der Breite **width** und der Höhe **height**. Bezugspunkt des Rechtecks ist die linke obere Ecke. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
canvas := j_canvas(frame,220,50);
j_fillrect(canvas,20,10,180,30);
:

```



fillroundrect

Synopsis `procedure j_fillroundrect (obj: integer; x: integer; y: integer;
width: integer; height: integer; arcx: integer; arcy: integer) ;`

Argumente

obj	integer
x	integer
y	integer
width	integer
height	integer
arcx	integer
arcy	integer

Beschreibung Zeichnet ein gefülltes Rechteck mit abgerundeten Ecken an die Position **(x,y)**. Das Rechteck hat eine Breite von **width** und eine Höhe von **height** Pixeln. Die Form der runden Ecken wird durch die Parameter **arcx** und **arcy** bestimmt. Diese Parameter bezeichnen die Radien der Rundungen in horizontaler bzw. vertikaler Richtung.
Bezugspunkt des Rechtecks ist die linke obere Ecke. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

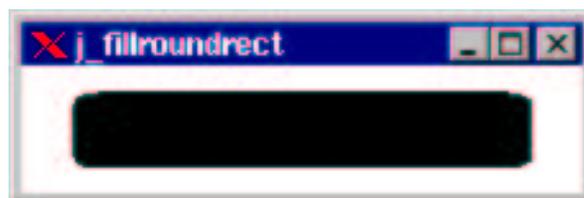
Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
canvas := j_canvas(frame,220,50);
j_fillroundrect(canvas,20,10,180,30,10,5);
:

```



focuslistener

Synopsis	function j_focuslistener (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Bindet an die Komponente obj einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

frame

Synopsis `function j_frame (label: PChar) : integer;`

Argumente `label PChar`

Beschreibung Diese Funktion erzeugt einen Frame mit der Titelzeile **label**. Ein Frame ist *das* elementare Fenster einer Applikation, an die die anderen graphischen Elemente gebunden werden. Die Funktion liefert die Eventnummer des Frames zurück. Ein Event wird erzeugt, indem der Benutzer das Close Icon im Fensterrahmen anklickt. Ein Frame wird zunächst nur erzeugt, aber nicht angezeigt. Erst nach dem Aufruf der Prozedur ***j_show(frame)*** wird der Frame sichtbar.

Beispiel

```
:  
frame := j_frame('j_frame');  
j_show(frame);  
:
```



getaction

Synopsis function **j_getaction** : integer;

Beschreibung Liefert den nächsten Benutzerevent, oder 0 falls kein weiterer Event vorliegt.

getcolumns

Synopsis `procedure j_getcolumns (obj: integer) ;`

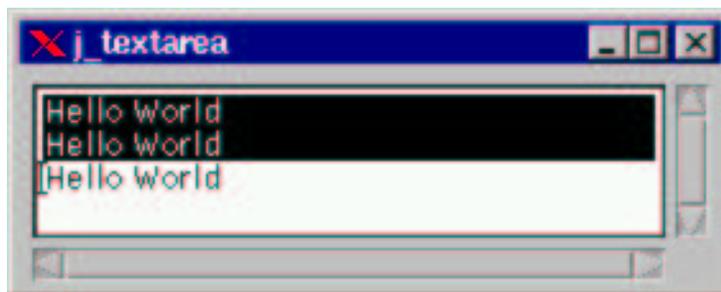
Argumente `obj` `integer`

Beschreibung Gibt die Anzahl der Spalten von **obj** zurück.

Zielkomponenten Textarea, Textfield, Gridlayout

Beispiel

```
:  
text := j_text(frame,30,4);  
j_getcolumns(text);  
:  
> 30
```



getcurpos

Synopsis `function j_getcurpos (obj: integer) : integer;`

Argumente `obj integer`

Beschreibung Liefert die Position des Text Cursors zurück.

Zielkomponenten Textarea, Textfield

getdanger

Synopsis procedure **j_getdanger** (obj: integer) ;

Argumente obj integer

Beschreibung Liefert den Warnlevel der Komponenten **obj** zurück.

Zielkomponenten Meter

getfontascent

Synopsis	function j_getfontascent (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Liefert die Oberlänge (Ascent) des aktuellen Fonts der Komponenten obj in Bildschirmpixeln zurück.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment

getfontheight

Synopsis	function j_getfontheight (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Liefert die totale Höhe des aktuellen Fonts der Komponenten obj in Bildschirmpixeln zurück.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment

getheight

Synopsis	function j_getheight (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Die Funktion liefert die Höhe der Komponenten obj zurück. Neben den graphischen Elementen kann auch die Höhe eines Image ermittelt werden. Die Maßeinheit sind Bildschirmpixel.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, Image
Beispiel	<pre>: label := j_getlabel(frame,'Hello World'); writeln(j_getheight(label)); : > 22</pre>

getimagesource

Synopsis	function j_getimagesource (obj: integer; x: integer; y: integer; w: integer; h: integer; r: var integer; g: var integer; b: var integer) : integer;																
Argumente	<table><tr><td>obj</td><td>integer</td></tr><tr><td>x</td><td>integer</td></tr><tr><td>y</td><td>integer</td></tr><tr><td>w</td><td>integer</td></tr><tr><td>h</td><td>integer</td></tr><tr><td>r</td><td>var integer (call by reference of the 1st array element)</td></tr><tr><td>g</td><td>var integer (call by reference of the 1st array element)</td></tr><tr><td>b</td><td>var integer (call by reference of the 1st array element)</td></tr></table>	obj	integer	x	integer	y	integer	w	integer	h	integer	r	var integer (call by reference of the 1st array element)	g	var integer (call by reference of the 1st array element)	b	var integer (call by reference of the 1st array element)
obj	integer																
x	integer																
y	integer																
w	integer																
h	integer																
r	var integer (call by reference of the 1st array element)																
g	var integer (call by reference of the 1st array element)																
b	var integer (call by reference of the 1st array element)																
Beschreibung	Die Funktion liefert einen Ausschnitt vom Punkt (x , y) der Breite w und der Höhe h zurck. Der Bildinhalt wird als rot-, grün- und blau- Kanal in den Arrays r,g,b abgelegt.																
Zielkomponenten	Canvas, Image																

getimage

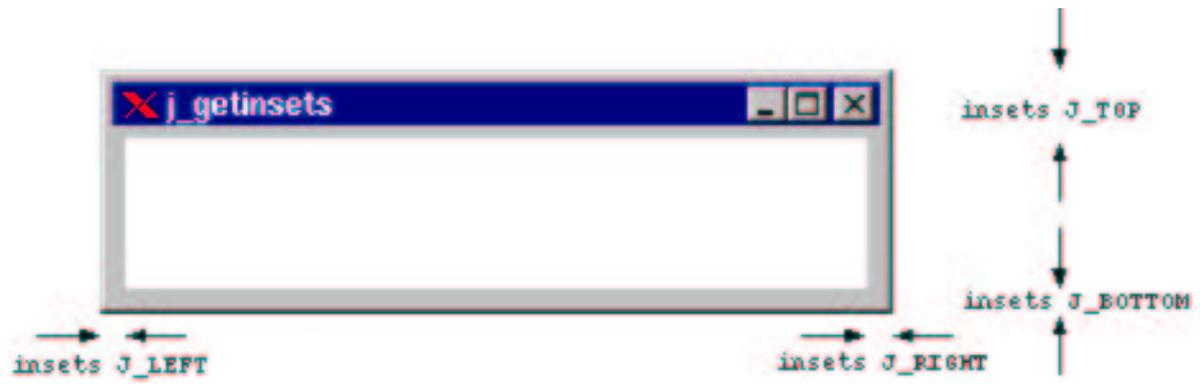
Synopsis	function j_getimage (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Kopiert den Inhalt der Komponenten obj in ein Image und liefert dessen Eventnummer zurück.
Zielkomponenten	Canvas, Image

getinheight

Synopsis	<code>function j_getinheight (cont: integer) : integer;</code>
Argumente	<code>cont integer</code>
Beschreibung	Die Funktion liefert die Höhe des inneren Bereichs der Komponenten obj zurück. Die Maßeinheit sind Bildschirmpixel.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame
Beispiel	<pre>: frame := j_frame('Hello World'); j_setsize(frame,300,400); writeln(j_getinheight(frame)); : > 370</pre>

getinsets

Synopsis	<code>function j_getinsets (obj: integer; side: integer) : integer;</code>				
Argumente	<table> <tr> <td style="padding-right: 20px;"><code>obj</code></td> <td><code>integer</code></td> </tr> <tr> <td><code>side</code></td> <td><code>integer</code></td> </tr> </table>	<code>obj</code>	<code>integer</code>	<code>side</code>	<code>integer</code>
<code>obj</code>	<code>integer</code>				
<code>side</code>	<code>integer</code>				
Beschreibung	<p>Liefert die angeforderte Breite des inneren Randes der Komponenten zurück. Der Parameter side kann folgende Werte annehmen:</p> <ul style="list-style-type: none"> • J_TOP: die Höhe des oberen Rands wird zurueckgeliefert. • J_BOTTOM: die Höhe des unteren Rands wird zurueckgeliefert. • J_LEFT: die Breite des linken Rands wird zurueckgeliefert. • J_RIGHT: die Breite des rechten Rands wird zurueckgeliefert. 				
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame				
Beispiel	<pre> : frame = j_frame('j_getinsets'); writeln(j_getinsets(frame,J_TOP),j_getinsets(frame,J_BOTTOM), j_getinsets(frame,J_LEFT),j_getinsets(frame,J_RIGHT)); : > 25 5 5 6 </pre>				



getinwidth

Synopsis	function j_getinwidth (cont: integer) : integer;
Argumente	cont integer
Beschreibung	Die Funktion liefert die Breite des inneren Bereich der Komponenten obj zurück. Die Maßeinheit sind Bildschirmpixel.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame
Beispiel	<pre>: frame := j_frame('Hello World'); j_setsize(frame,300,400); writeln(j_getinwidth(frame)); : > 289</pre>

getitemcount

Synopsis function **j_getitemcount** (obj: integer) : integer;

Argumente obj integer

Beschreibung Liefert die Anzahl der Items der Komponenten **obj** zurück.

Zielkomponenten List, Choice

getitem

Synopsis	function j_getitem (obj: integer; item: integer; str: PChar) : PChar;
Argumente	obj integer item integer str var PChar
Beschreibung	liefert den Inhalt des Items mit dem Index item zurück.
Zielkomponenten	List, Choice

getkeychar

Synopsis	function j_getkeychar (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Liefert den ASCII Code, der zuletzt gedrückten Taste zurück.
Zielkomponenten	KeyListener

getkeycode

Synopsis	function j_getkeycode (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Liefert den Keycode, der zuletzt gedrückten Taste zurück.
Zielkomponenten	KeyListener

getlayoutid

Synopsis	function j_getlayoutid (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Liefert die Eventnummer des Layoutmanagers des Containers obj zurück.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame
Beispiel	

```
:  
j_setgridlayout(frame,2,2);  
grid := j_getlayoutid(frame);  
:
```

getlength

Synopsis	function j_getlength (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Liefert die Länge des Textinhaltes die Komponente zurück.
Zielkomponenten	Textarea, Textfield, Dialog, Frame, Button, MenuItem, CheckBox-MenuItem, Menu, HelpMenu, Popupmenu

getmousebutton

Synopsis	function j_getmousebutton (mouset listener: integer) : integer;
Argumente	mouset listenerinteger
Beschreibung	Liefert die letzte gedrückte Mousetaste zurück. Der Rückgabewert ist <ul style="list-style-type: none">• J_LEFT linke Mousetaste• J_CENTER mittlere Mousetaste• J_RIGHT rechte Mousetaste
Zielkomponenten	Mouset listener

getmousex

Synopsis `function j_getmousex (mouset listener: integer) : integer;`

Argumente `mouset listenerinteger`

Beschreibung Liefert die aktuelle horizontale Position der Mouse zurück.

Zielkomponenten Mouselistener

getmousey

Synopsis `function j_getmousey (mouset listener: integer) : integer;`

Argumente `mouset listenerinteger`

Beschreibung Liefert die aktuelle vertikale Position der Mouse zurück.

Zielkomponenten Mouselistener

getparentid

Synopsis	<code>function j_getparentid (obj: integer) : integer;</code>
Argumente	<code>obj</code> <code>integer</code>
Beschreibung	Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird <code>-1</code> zurückgegeben.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, Menubar, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Radiogroup

Beispiel

```

:
radio1      := j_radiobutton(j_radiogroup(frame), 'Radiobutton 1');
radio2      := j_radiobutton(j_getparentid(radio1), 'Radiobutton 2');
:

```



getparent

Synopsis	function j_getparent (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, Menubar, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Radiogroup

Beispiel

```

:
radio1      := j_radiobutton(j_radiogroup(frame), 'Radiobutton 1');
radio2      := j_radiobutton(j_getparent(radio1), 'Radiobutton 2');
:

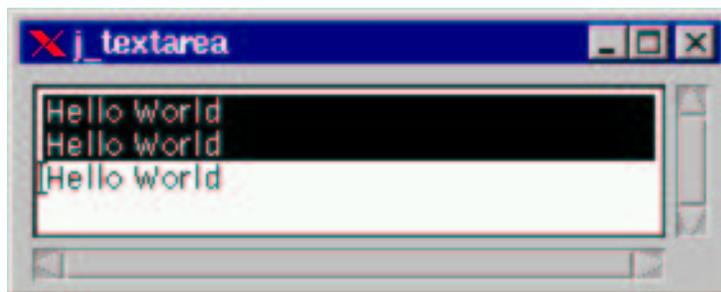
```



getrows

Synopsis	procedure j_getrows (obj: integer) ;
Argumente	obj integer
Beschreibung	Gibt die Anzahl der Reihen von obj zurck.
Zielkomponenten	Textarea, Gridlayout
Beispiel	

```
:  
text := j_text(frame,30,4);  
j_getrows(text);  
:  
> 4
```



getscaledimage

Synopsis	function j_getscaledimage (obj: integer; x: integer; y: integer; sw: integer; sh: integer; tw: integer; th: integer) : integer;
Argumente	obj integer x integer y integer sw integer sh integer tw integer th integer
Beschreibung	Kopiert den Inhalt des Ausschnitts von Positio (x , y) it der Breite sw und der höhe sh in ein Image und liefert dessen Eventnummer zurück. Das Image wird dabei auf eine Breite von tw und einer Höhe von th skaliert.
Zielkomponenten	Canvas, Image

getscreenheight

Synopsis	function j_getscreenheight : integer;
Beschreibung	Liefert die Höhe des Bildschirms in Pixeln zurück. Ist ein virtueller Bildschirm installiert, so wird dessen Höhe zurückgeliefert.
Beispiel	<pre>: writeln(j_getscreenwidth(), j_getscreenheight()); : > 1280 1024</pre>

getscreenwidth

Synopsis function **j_getscreenwidth** : integer;

Beschreibung Liefert die Breite des Bildschirms in Pixeln zurück. Ist ein virtueller Bildschirm installiert, so wird dessen Breite zurückgeliefert.

Beispiel

```
:  
writeln(j_getscreenwidth(), j_getscreenheight());  
:  
  
> 1280 1024
```

getselect

Synopsis `function j_getselect (obj: integer) : integer;`

Argumente `obj integer`

Beschreibung Liefert den Index des selectierten Items der Komponenten .

Zielkomponenten List, Choice

getselend

Synopsis `function j_getselend (obj: integer) : integer;`

Argumente `obj integer`

Beschreibung Liefert die Endposition des selectierten Textes.

Zielkomponenten Textarea, Textfield

getselstart

Synopsis `function j_getselstart (obj: integer) : integer;`

Argumente `obj integer`

Beschreibung Liefert die Anfangsposition des selectierten Textes.

Zielkomponenten Textarea, Textfield

getseltext

Synopsis `function j_getseltext (obj: integer; text: PChar) : PChar;`

Argumente `obj integer`
 `text var PChar`

Beschreibung Liefert den selectierten Text die Komponente **obj** zurück.

Zielkomponenten Textarea, Textfield

getstate

Synopsis	function j_getstate (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Liefert true , falls die Komponente selektiert ist, sonst false .
Zielkomponenten	Checkbox, Radiobutton, CheckmenuItem, Led

getstringwidth

Synopsis	function j_getstringwidth (obj: integer; str: PChar) : integer;
Argumente	obj integer str PChar
Beschreibung	Liefert die Länge des str des aktuellen Fonts der Komponenten obj in Bildschirmpixeln zurück.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

gettext

Synopsis	<code>function j_gettext (obj: integer; str: PChar) : PChar;</code>				
Argumente	<table style="border: none;"> <tr> <td style="padding-right: 20px;"><code>obj</code></td> <td><code>integer</code></td> </tr> <tr> <td><code>str</code></td> <td><code>var PChar</code></td> </tr> </table>	<code>obj</code>	<code>integer</code>	<code>str</code>	<code>var PChar</code>
<code>obj</code>	<code>integer</code>				
<code>str</code>	<code>var PChar</code>				
Beschreibung	Liefert den Textinhalt der, durch obj referenzierten, Komponenten im Parameter str zurück. In C und in Pascal wird ein Pointer auf str zurückgeliefert. Der übergebene String muß groß genug sein, um den gesamten Inhalt des Textes aufzunehmen. Ist die Komponente vom Typ Frame oder Dialog, so wird der Titelstring des Fensters zurückgegeben. Ansonsten wird der Textinhalt der Komponenten zurückgeliefert.				
Zielkomponenten	Button, Label, Checkbox, Radiobutton, Dialog, Frame, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popumenu, Textarea, Textfield				
Beispiel	<pre> str : PChar; : getmem(str,256); label := j_label(frame,'Hello World'); writeln(j_gettext(label,str)); : > Hello World </pre>				

getvalue

Synopsis `function j_getvalue (obj: integer) : integer;`

Argumente `obj integer`

Beschreibung Liefert den aktuellen Wert der Komponenten .

Zielkomponenten Scrollbar

getviewportheight

Synopsis	function j_getviewportheight (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Gibt die Höhe des sichtbaren Teils der Komponenten obj zurück.
Zielkomponenten	Scrollpane

getviewportwidth

Synopsis	function j_getviewportwidth (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Gibt die Breite des sichtbaren Teils der Komponenten obj zurück.
Zielkomponenten	Scrollpane

getwidth

Synopsis	function j_getwidth (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Die Funktion liefert die Breite der Komponenten obj zurück. Neben den graphischen Elementen kann auch die Breite eines Image ermittelt werden. Die Maßeinheit sind Bildschirmpixel.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, Image
Beispiel	<pre>: label := j_getlabel(frame,'Hello World'); writeln(j_getwidth(label)); : > 84</pre>

getxpos

Synopsis	function j_getxpos (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Liefert die aktuelle horizontale Position der Komponenten obj zurück.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

getypos

Synopsis	function j_getypos (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Liefert die aktuelle vertikale Position der Komponenten obj zurück.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

graphicbutton

Synopsis function **j_graphicbutton** (obj: integer; filename: PChar) :
integer;

Argumente obj integer
 filename PChar

Beschreibung Die Funktion erzeugt einen neuen Button mit einem Icon als Inhalt. Die Datei, die das Icon enthält, ist im Parameter **filename** anzugeben. Das Icon muß im GIF oder JPEG-Format vorliegen. Die Funktion liefert die Eventnummer des Buttons zurück, oder im Fehlerfall -1 . Ein Event wird erzeugt, wenn der Benutzer mit der Maus den Button betätigt.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:  
frame := j_frame('j_graphicbutton');  
button := j_graphicbutton(frame, 'save.gif');  
:
```



graphiclabel

Synopsis `function j_graphiclabel (obj: integer; str: PChar) : integer;`

Argumente `obj integer`
 `str PChar`

Beschreibung Die Funktion erzeugt ein neues Label mit einem Icon als Inhalt. Die Datei, die das Icon enthält, ist im Parameter **filename** anzugeben. Das Icon muß im GIF oder JPEG-Format vorliegen. Die Funktion liefert die Eventnummer des Labels zurück, oder im Fehlerfall `-1`.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
frame := j_frame('j_graphiclabel');
label := j_graphiclabel(frame, 'new.gif');
:

```



hasfocus

Synopsis	function j_hasfocus (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Liefert true zurück, falls die Komponente den Focus besitzt, sonst false .
Zielkomponenten	Focuslistener

helpmenu

Synopsis `function j_helpmenu (obj: integer; label: PChar) : integer;`

Argumente `obj integer`
 `label PChar`

Beschreibung Die Funktion erzeugt einen Hilfe-Menueintrag mit der Beschriftung **label** in der Komponenten **obj**, und liefert dessen Eventnummer zurück. An diese Menukomponente können in weiteren Schritten MenuItem gebunden werden. Eine Menukomponente selbst erzeugt keinen Event. Hilfe Menü unterscheiden sich von normalen Menüs dadurch, daß sie je nach Oberfläche auch rechtebündig erscheinen können. Zudem sind als Zielkomponenten nur Menubars zulässig.

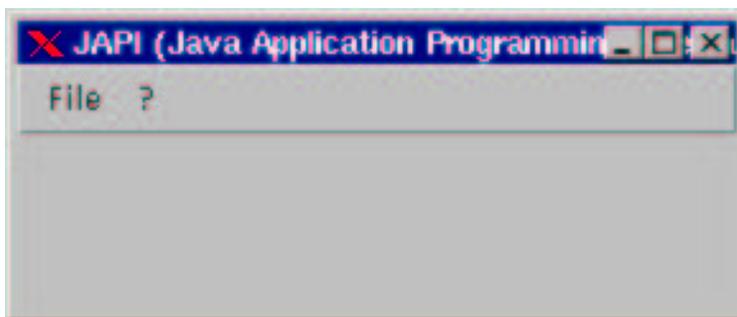
Zielkomponenten Menubar

Beispiel

```

:
frame    := j_frame('Menu Komponenten');
menubar := j_menubar(frame);
file:= j_menu(menubar,'File');
help:= j_helpmenu(menubar,'?')
:

```



hide

Synopsis	procedure j_hide (obj: integer) ;
Argumente	obj integer
Beschreibung	Zeigt die Komponente obj nicht mehr auf dem Bildschirm an.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

hscrollbar

Synopsis `function j_hscrollbar (obj: integer) : integer;`

Argumente `obj integer`

Beschreibung Diese Funktion erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück. Im Fehlerfall wird `-1` zurückgegeben. Ist die Zielkomponente **obj** vom Typ Scrollpane, so wird die Eventnummer der vorhandenen horizontalen Scrollbar zurückgegeben. Eine Scrollbar erzeugt einen Event, sobald der Schieber bewegt wird.

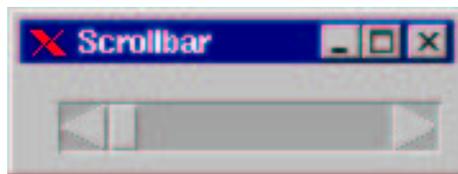
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame, Scrollpane

Beispiel

```

:
scroll:=j_hscrollbar(frame);
j_setpos(scroll,20,40);
j_setsize(scroll,150,20);
:

```



image

Synopsis `function j_image (width: integer; height: integer) : integer;`

Argumente `width integer`
 `height integer`

Beschreibung Diese Funktion erzeugt einen neues (Memory) Image mit der Breite **width** Pixel und einer Höhe von **height** Pixel. Die Funktion liefert die Eventnummer des Image zurück, oder im Fehlerfall `-1`. Ein Image erzeugt keinen Event

Beispiel

```
:  
image := j_image(200,200);  
:
```

insert

Synopsis	function j_insert (obj: integer; pos: integer; label: PChar) : integer;
Argumente	obj integer pos integer label PChar
Beschreibung	fügt ein neues Item in an der Position pos ein. Das Item trägt den Inhalt label .
Zielkomponenten	List, Choice

inserttext

Synopsis	procedure j_inserttext (obj: integer; text: PChar; pos: integer) ;
Argumente	obj integer text PChar pos integer
Beschreibung	Fügt zusätzlichen Text text an der Position pos ein.
Zielkomponenten	Textarea

isparent

Synopsis	function j_isparent (obj: integer; cont: integer) : integer;
Argumente	obj integer cont integer
Beschreibung	Liefert true zurück, falls das cont die umgebende Komponente von obj ist, sonst false
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabeled, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, Menubar, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Radiogroup

isresizable

Synopsis	function j_isresizable (obj: integer) : boolean;
Argumente	obj integer
Beschreibung	Liefert false zurueck, falls die Komponente eine fixe Größe besitzt, true sonst.
Zielkomponenten	Dialog, Frame

isselect

Synopsis	function j_isselect (obj: integer; item: integer) : integer;
Argumente	obj integer item integer
Beschreibung	Liefert true zurück, falls das Item mit dem Index item selectiert ist, sonst false
Zielkomponenten	List

isvisible

Synopsis	function j_isvisible (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Liefert true zurück, falls obj sichtbar ist, sonst false
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

keylistener

Synopsis	function j_keylistener (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Bindet an die Komponente obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

label

Synopsis `function j_label (obj: integer; label: PChar) : integer;`

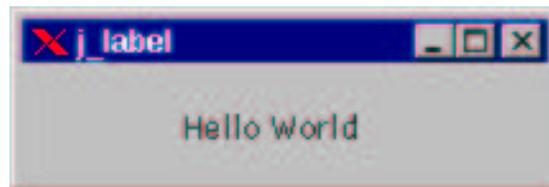
Argumente `obj integer`
 `label PChar`

Beschreibung Die Funktion erzeugt ein neues Label Element mit dem Textinhalt **label**. Ein Label stellt einen einzeiligen Text beliebiger Länge dar. Die Funktion liefert die Eventnummer des Labels zurück, oder im Fehlerfall `-1`.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

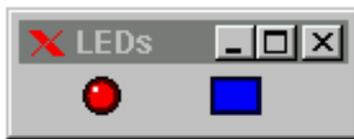
Beispiel

```
:  
frame := j_frame('j_label');  
label := j_label(frame,'Hello World');  
:
```



led

Synopsis	<code>function j_led (obj: integer; style: integer; color: integer) : integer;</code>						
Argumente	<table> <tr> <td style="padding-right: 20px;"><code>obj</code></td> <td><code>integer</code></td> </tr> <tr> <td><code>style</code></td> <td><code>integer</code></td> </tr> <tr> <td><code>color</code></td> <td><code>integer</code></td> </tr> </table>	<code>obj</code>	<code>integer</code>	<code>style</code>	<code>integer</code>	<code>color</code>	<code>integer</code>
<code>obj</code>	<code>integer</code>						
<code>style</code>	<code>integer</code>						
<code>color</code>	<code>integer</code>						
Beschreibung	Die Funktion erzeugt ein neues LED-Element und liefert die Eventnummer der Komponente zurück. Die LED kann eine runde Form besitzen <code>style = J_ROUND</code> oder eine eckige Form besitzen <code>style=J_RECT</code> . Die Farbe der LED kann eine der vordefinierten Farben sein (zB. <code>J_RED</code> , <code>J_GREEN</code>).						
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame						
Beispiel	<pre> : led1 := j_led(frame,J_ROUND,J_RED); led2 := j_led(frame,J_RECT,J_BLUE); : </pre>						



line

Synopsis function **j_line** (obj: integer; orient: integer; style: integer;
length: integer) : integer;

Argumente obj integer
 orient integer
 style integer
 length integer

Beschreibung Die Funktion erzeugt ein Linienelement und liefert dessen Eventnummer zurück. Im Fehlerfall wird -1 zurückgegeben. Das Element wird der Komponente zugeordnet, die durch **obj** referenziert wird. Die initiale Laenge der Linie wird durch den Parameter **length** bestimmt. Der Ruler dient der optischen Trennung anderer Komponenten.
Der Parameter **orient** gibt an, ob eine senkrechte oder waagerechte Linie gezogen werden soll:

- J_HORIZONTAL : waagerechte Linie
- J_VERTICAL : senkrechte Linie

Der Parameter **style** gibt den Linientyp vor:

- J_LINEDOWN : Der Linie scheint optisch abgesenkt.
- J_LINEUP : Der Linie scheint optisch hervorzutreten.

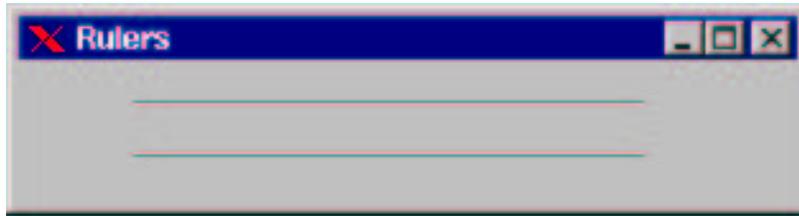
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
line1 := j_line(frame,J_HORIZONTAL,J_LINEDOWN,200);
line2 := j_line(frame,J_HORIZONTAL,J_LINEUP,200);
:

```



list

Synopsis `function j_list (obj: integer; rows: integer) : integer;`

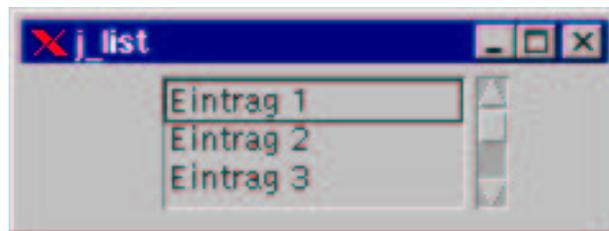
Argumente `obj integer`
 `rows integer`

Beschreibung Diese Funktion erzeugt eine neue List und gibt deren Eventnummer zurück. Im Fehlerfall wird `-1` zurückgegeben. Eine List bietet dem Benutzer die Auswahl zwischen mehreren vordefinierten Einträgen. Es sind immer **rows** Einträge sichtbar, die anderen Einträge sind über Rollbalken zu erreichen. Eine List bietet die Möglichkeit, sowohl Einfach- als auch Mehrfachselektionen zuzulassen (siehe `j_multiplemode()`). Bei Einfachselektion liefert die List bei einem Doppelklick auf einen Eintrag einen Event. Ist eine Mehrfachselektion eingestellt, löst jeder einfache Klick auf einen Eintrag einen Event aus.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:
list := j_list(frame,3);
j_additem(list,'Eintrag 1');
j_additem(list,'Eintrag 2');
:
```



loadimage

Synopsis	function j_loadimage (filename: PChar) : integer;
Argumente	filename PChar
Beschreibung	Liest das Image aus dem File filename ein und liefert dessen Eventnummer zurück. Es werden folgende Fileformate akzeptiert: <ul style="list-style-type: none">• GIF• JPEG• BMP• PPM
Beispiel	<pre>: image := j_loadimage("mandel.jpg"); :</pre>

menubar

Synopsis `function j_menubar (obj: integer) : integer;`

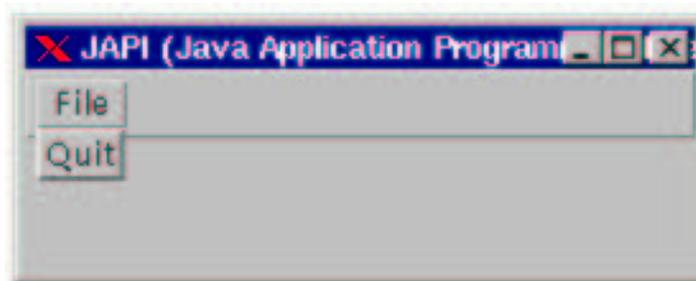
Argumente `obj integer`

Beschreibung Die Funktion erzeugt eine neue Menubar in der Komponente, die durch **obj** referenziert wird, und liefert dessen Eventnummer zurück. An die Menubar können in weiteren Schritten Menu Komponenten gebunden werden. Eine Menubar selbst erzeugt keinen Event. Als Zielkomponenten sind nur Frames zulässig.

Zielkomponenten `Frame`

Beispiel

```
:  
frame    := j_frame('Menu Komponenten');  
menubar := j_menubar(frame);  
file     := j_menu(menubar,'File');  
quit     := j_menuitem(file,'Quit');  
:
```



menuitem

Synopsis `function j_menuitem (obj: integer; label: PChar) : integer;`

Argumente `obj integer`
 `label PChar`

Beschreibung Die Funktion erzeugt ein MenuItem mit der Beschriftung **label** in der Komponente **obj**, und liefert dessen Eventnummer zurück. Eine MenuItem erzeugt einen Event, wenn es durch einen Mausklick ausgewählt wurde. Als Zielkomponenten sind nur Menu, Popupmenu und Helpmenu zulässig.

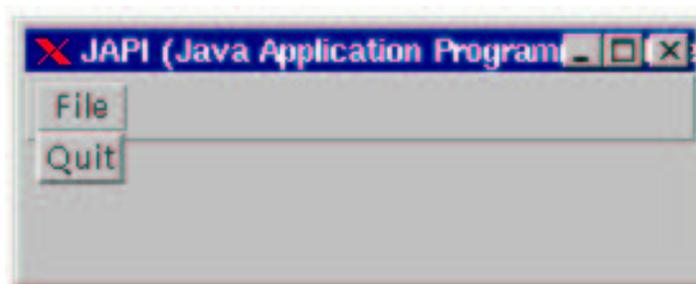
Zielkomponenten Menu, Popupmenu, Helpmenu

Beispiel

```

:
frame := j_frame('Menu Komponenten');
menubar := j_menubar(frame);
file := j_menu(menubar, 'File');
quit := j_menuitem(file, 'Quit');
:

```



menu

Synopsis function **j_menu** (obj: integer; str: PChar) : integer;

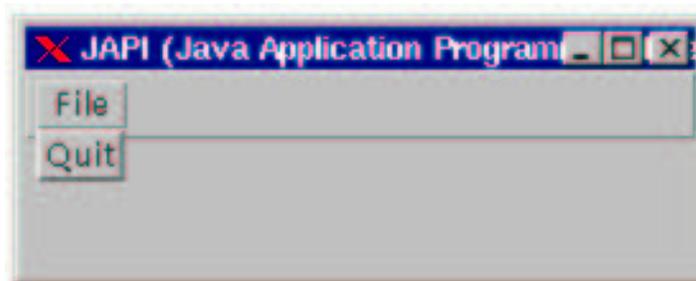
Argumente obj integer
 str PChar

Beschreibung Die Funktion erzeugt einen Menueintrag mit der Beschriftung **label** in der Komponente, die durch **obj** referenziert wird, und liefert dessen Eventnummer zurück. An diese Menukomponente können in weiteren Schritten MenuItem gebunden werden. Eine Menukomponente selbst erzeugt keinen Event. Als Zielkomponenten sind nur Menubars und andere Menukomponenten zulässig.

Zielkomponenten Menubar, Menu

Beispiel

```
:  
frame   := j_frame('Menu Komponenten');  
menubar := j_menubar(frame);  
file    := j_menu(menubar, 'File');  
quit    := j_menuitem(file, 'Quit');  
:
```



messagebox

Synopsis procedure **j_messagebox** (obj: integer; title: PChar; text: PChar);

Argumente obj integer
 title PChar
 text PChar

Beschreibung Öffnet eine MessageBox mit der Überschrift **title** und dem Textinhalt **text** und liefert dessen Eventnummer zurück. Im Fehlerfall wird -1 zurückgeliefert. Eine MessageBox erzeugt ein Event, wenn das Close Icon im Fensterrahmen angeklickt wird.

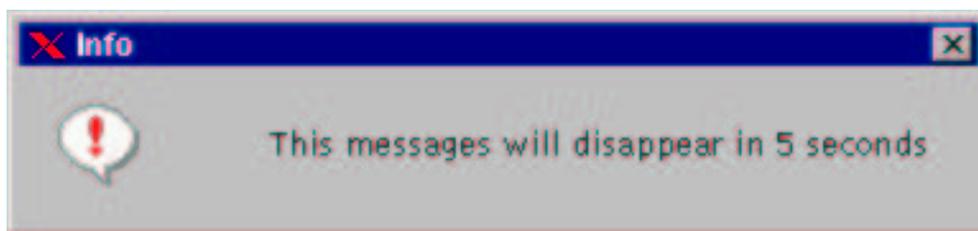
Zielkomponenten Frame

Beispiel

```

:
mbox := j_messagebox(frame,'Info','This messages will disappear in 5 seconds');
j_sleep(5000);
j_dispose(mbox);
:

```



meter

Synopsis function **j_meter** (obj: integer; title: PChar) : integer;

Argumente obj integer
 title PChar

Beschreibung Die Funktion erzeugt ein analoges Anzeigeelement und liefert die Eventnummer der Komponente zurück. Das Instrument trägt die Beschriftung **title**. Das Instrument besitzt einen vordefinierten Bereich von 0 bis 100. Die Werte können mit `j_setmin()` und `j_setmax()` verändert werden. Ein Gefahr-Bereich ist ab dem Wert 80 voreingestellt. Dies kann mit `j_setdanger()` verändert werden.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:  
meter := j_meter(frame,"Volt");  
j_setvalue(meter,40);  
:
```



mouselistener

Synopsis	function j_mouselistener (obj: integer; kind: integer) : integer;
Argumente	obj integer kind integer
Beschreibung	<p>Bindet an die Komponente obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird. Zulässige Werte für kind sind:</p> <ul style="list-style-type: none"> • J_ENTERED : Der Mouselistener liefert einen Event, wenn der Mauszeiger die Komponente obj betritt. • J_MOVED : Der Mouselistener liefert einen Event, wenn der Mauszeiger sich innerhalb der Komponenten obj bewegt. • J_EXITED : Der Mouselistener liefert einen Event, wenn der Mauszeiger die Komponente obj verläßt. • J_PRESSED : Der Mouselistener liefert beim Drücken einer Mousetaste einen Event. • J_DRAGGED : Der Mouselistener liefert einen Event, wenn der Mauszeiger sich innerhalb der Komponenten obj mit gedrückter Taste bewegt. • J_RELEASED : Der Mouselistener liefert beim Loslassen einer Mousetaste einen Event. • J_DOUBLECLICK : Der Mouselistener liefert bei einem Doppelclick einen Event.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

multiplemode

Synopsis	function j_multiplemode (obj: integer; bool: boolean) : integer;
Argumente	obj integer bool boolean
Beschreibung	die Komponente erlaubt mehrfache Selectierung, falls bool true ist.
Zielkomponenten	List

nextaction

Synopsis function **j_nextaction** : integer;

Beschreibung Wartet auf den nächsten Benutzerevent und liefert diesen zurück.

pack

Synopsis `procedure j_pack (obj: integer) ;`

Argumente `obj integer`

Beschreibung Diese Prozedur veranlaßt den Layoutmanager von **obj** die minimale Größe zu berechnen. Ist die Komponente sichtbar, so wird das neue Layout direkt angezeigt. Der Layoutmanager erfragt von allen enthaltenen Komponenten deren initiale bzw. explizit zugewiesene Größe, und berechnet daraus ein neues Layout.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:  
j_setflowlayout(jframe,J_HORIZOMTAL);  
canvas := j_canvas(frame,200,50);  
j_setnamedcolorbg(canvas,J_RED);  
j_pack(frame);  
:
```



panel

Synopsis `function j_panel (obj: integer) : integer;`

Argumente `obj integer`

Beschreibung Diese Funktion erzeugt ein neues panel und liefert dessen Eventnummer zurück. Im Fehlerfall wird `-1` zurückgegeben. Das Panel wird der Komponente zugeordnet, die durch **obj** referenziert wird. Ein Panel ist ein (unsichtbares) Behälterelement, in dem andere Elemente angeordnet werden können. Da einem Panel auch ein Layoutmanager zugewiesen werden kann, können mit Hilfe der Panels komplexe hirarchisch aufgebaute Benutzerschnittstellen erstellt werden. Ein Panel liefert bei einer Größenänderung einen Event.

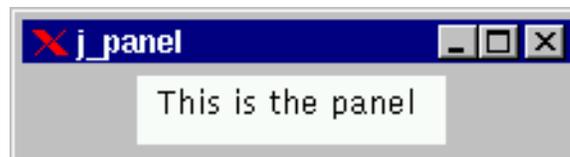
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
panel := j_panel(frame);
j_setnamedcolorbg(panel,J_WHITE);
j_setpos(panel,50,30);
label := j_label(panel,'This is the panel');
j_setpos(label,0,0);
:

```



popupmenu

Synopsis function **j_popupmenu** (obj: integer; label: PChar) : integer;

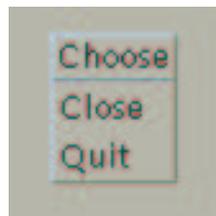
Argumente obj integer
 label PChar

Beschreibung Diese Funktion erzeugt ein neues Popupmenu und liefert dessen Eventnummer zurück. Im Fehlerfall wird -1 zurückgegeben. Ein Popupmenu ist funktional identisch mit einem Menu, es ist allerdings an keine Menuleiste gebunden. Daher kann ein Popupmenu prinzipiell an jeder Position innerhalb der Oberfläche platziert werden. Wie an Menus kann auch an ein Popupmenu MenuItem's gebunden werden, die bei Anklicken des Popupmenus sichtbar werden.

Zielkomponenten Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment

Beispiel

```
:  
choose := j_popupmenu(frame, 'Choose');  
close  := j_menuitem(choose, 'Close');  
quit   := j_menuitem(choose, 'Quit');  
j_showpopup(choose, 100, 100);  
:
```



printer

Synopsis `function j_printer (frame: integer) : integer;`

Argumente `frame integer`

Beschreibung Diese Funktion erzeugt ein neues Objekt, das eine Papierseite eines Druckes repräsentiert. Bis auf wenige Ausnahmen ist ein Printer Objekt funktional einem Canvas gleichgestellt. Die Funktion liefert die Eventnummer des Printers zurück, oder im Fehlerfall `-1`. Ein Printer erzeugt keinen Event.

Zielkomponenten `Frame`

Beispiel

```
:  
printer := j_printer(frame);  
j_drawimage(printer,image,100,100);  
:
```

print

Synopsis	procedure j_print (obj: integer) ;
Argumente	obj integer
Beschreibung	<p>Die Prozedur druckt den Inhalt der Komponente obj. Ist die Komponente von Typ Container, so wird der gesamte Inhalt des Containers gedruckt. Nach dem Aufruf dieser Funktion erscheint ein Plattformabhängiger Print Dialog, in dem weitere Optionen (beispielsweise das Speichern in eine Datei) bereitgestellt werden. Unter X-Windows besitzt der Ausdruck immer Motif Look, unabhängig von der Bildschirmdarstellung.</p> <p>Ist die Komponente von Type j_printer, dann wird die aktuelle Seite geschlossen, und eine neue Seite begonnen. Die Seiten werden noch nicht auf dem Drucker ausgegeben. Um den Druck zu starten muss j_dispose(printer) aufgerufen werden.</p>
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, Canvas, Image, Printer
Beispiel	<pre> : frame := j_frame('j_textfield'); text := j_textfield(frame,30) : j_print(frame); : </pre>



progressbar

Synopsis `function j_progressbar (obj: integer; orient: integer) : integer;`

Argumente `obj integer`
 `orient integer`

Beschreibung Die Funktion erzeugt eine Progressbar und liefert die Eventnummer der Komponente zurück. Die Ausrichtung kann `J_HORIZONTAL` oder `J_VERTICAL` sein. Das Instrument besitzt einen vordefinierten Bereich von 0 bis 100. Die Werte können mit `j_setmin()` und `j_setmax()` verändert werden.

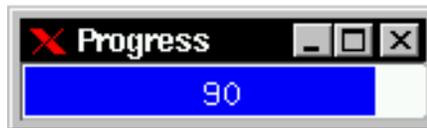
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
progress := j_progressbar(frame, J_HORIZONTAL);
j_setvalue(progress, 90);
:

```



quit

Synopsis `procedure j_quit ;`

Beschreibung Beendet die Verbindung zum JAPI Kernel.

radiobutton

Synopsis `function j_radiobutton (obj: integer; label: PChar) : integer;`

Argumente `obj integer`
 `label PChar`

Beschreibung Diese Funktion erzeugt einen Radiobutton mit dem Textinhalt **label** und liefert dessen Eventnummer zurück. Im Fehlerfall wird `-1` zurückgegeben. Ein Radiobutton muss immer in einer Radiogroup eingebettet sein. Innerhalb dieser Radiogroup kann von allen enthaltenen Radiobuttons immer nur maximal ein Button aktiviert sein. Ein Radiobutton liefert immer dann ein Event, wenn sich sein Zustand ändert.

Zielkomponenten Radiogroup

Beispiel

```

:
radiogroup := j_radiogroup(frame);
radio1     := j_radiobutton(radiogroup, 'Radiobutton 1');
radio2     := j_radiobutton(radiogroup, 'Radiobutton 2');
:

```



radiogroup

Synopsis function **j_radiogroup** (obj: integer) : integer;

Argumente obj integer

Beschreibung Diese Funktion erzeugt eine neue Radiogroup und liefert dessen Eventnummer zurück. Eine Radiogroup ist eine virtuelle Komponente, die dazu dient, Radiobutton funktionell zusammenzuschalten. Von allen Radiobutton, die in einer Radiogroup zusammengefasst sind, kann immer nur maximal ein Button aktiviert sein.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:  
radiogroup := j_radiogroup(frame);  
radio1     := j_radiobutton(radiogroup, 'Radiobutton 1');  
radio2     := j_radiobutton(radiogroup, 'Radiobutton 2');  
:
```



random

Synopsis function **j_random** : integer;

Beschreibung Liefert eine Pseudo Zufallszahl zwischen 0 und 2147483647 ($2^{31} - 1$) zurück .

releaseall

Synopsis `procedure j_releaseall (obj: integer) ;`

Argumente `obj integer`

Beschreibung Gibt alle Elemente aus der Komponenten **obj** frei.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

release

Synopsis	procedure j_release (obj: integer) ;
Argumente	obj integer
Beschreibung	Gibt die Komponente obj aus dem umgebenden Container Element frei.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

removeall

Synopsis function **j_removeall** (obj: integer) : integer;

Argumente obj integer

Beschreibung entfernt alle Items aus der Komponenten .

Zielkomponenten List, Choice

removeitem

Synopsis `function j_removeitem (obj: integer; item: PChar) : integer;`

Argumente `obj integer`
 `item PChar`

Beschreibung entfernt den ersten Eintrag **item** aus der Komponenten .

Zielkomponenten List, Choice

remove

Synopsis function **j_remove** (obj: integer; item: integer) : integer;

Argumente obj integer
 item integer

Beschreibung entfernt das Item mit dem Index **item** aus der Komponenten .

Zielkomponenten List, Choice

replacetext

Synopsis	procedure j_replacetext (obj: integer; text: PChar; start: integer; end: integer) ;
Argumente	obj integer text PChar start integer end integer
Beschreibung	Ersetzt den Text zwischen Position start bis end durch text .
Zielkomponenten	Textarea

saveimage

Synopsis function **j_saveimage** (obj: integer; filename: PChar; filetype:
integer) : boolean;

Argumente obj integer
 filename PChar
 filetype integer

Beschreibung Speichert den Inhalt der Komponenten in einem File mit dem Namen **filename**. Es werden zwei Fileformate unterstützt:

- BMP
- PPM

Beispiel

```
:  
if(j_saveimage(canvas,"mandel.bmp",J_BMP) = false)  
    writeln('Error saving Bitmap file');  
:
```

scrollpane

Synopsis `function j_scrollpane (obj: integer) : integer;`

Argumente `obj integer`

Beschreibung Diese Funktion erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück. Im Fehlerfall wird `-1` zurückgegeben. Eine Scrollpane ist eine Behälterkomponente, die ein weiteres Objekt aufnehmen kann. Dabei kann das aufgenommene Objekt größere Ausmaße haben als die Scrollpane. Über zwei automatisch angepaßte Scrollbars, kann das enthaltene Object gescrollt werden. Ein Layoutmanager kann nicht in eine Scrollpane integriert werden. Soll mehr als eine Komponente aufgenommen werden, so ist in die Scrollpane zunächst ein Panel zu integrieren. Eine Scrollpane liefert bei einer Größenänderung einen Event.

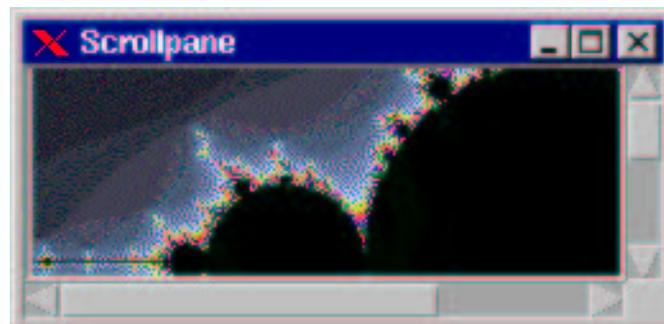
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
scrollpane := j_scrollpane(frame);
image := j_graphiclabel(scrollpane,'mandel.gif');
j_setsize(scrollpane,240,100);
:

```



selectall

Synopsis procedure **j_selectall** (obj: integer) ;

Argumente obj integer

Beschreibung Selectiert den gesamten Textinhalt der Komponenten .

Zielkomponenten Textarea, Textfield

select

Synopsis function **j_select** (obj: integer; item: integer) : integer;

Argumente obj integer
 item integer

Beschreibung selectiert das Item mit dem Index **item**.

Zielkomponenten List, Choice

selecttext

Synopsis	procedure j_selecttext (obj: integer; start: integer; end: integer) ;
Argumente	obj integer start integer end integer
Beschreibung	Selectiert den Text von Position start bis end .
Zielkomponenten	Textarea, Textfield

seperator

Synopsis	<code>procedure j_seperator (obj: integer) ;</code>
Argumente	<code>obj</code> <code>integer</code>
Beschreibung	Diese Prozedur erzeugt eine Trennlinie in einem Menu, das durch obj referenziert wird. Trennlinien dienen ausschließlich der optischen Trennung von MenuItemen, und besitzen keine weitere Funktionalität.
Zielkomponenten	Menu, HelpMenu, Popupmenu

Beispiel

```

:
file := j_menu(menubar, 'File');
open := j_menuitem(file, 'Open');
save := j_menuitem(file, 'Save');
j_seperator(file);
quit := j_menuitem(file, 'Quit');
:

```



setalign

Synopsis	procedure j_setalign (obj: integer; align: integer) ;
Argumente	obj integer align integer
Beschreibung	Setzt die Ausrichtung innerhalb der Komponenten obj auf align . Ein Flowlayout Manager wird vorausgesetzt.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

setblockinc

Synopsis	function j_setblockinc (obj: integer; val: integer) : integer;
Argumente	obj integer val integer
Beschreibung	Stellt die Blockänderung von der Komponenten auf den Wert von val ein.
Zielkomponenten	Scrollbar

setborderlayout

Synopsis `procedure j_setborderlayout (obj: integer) ;`

Argumente `obj integer`

Beschreibung Integriert einen BorderLayout Manager in die Komponente **obj**.

Zielkomponenten `Panel, Borderpanel, Window, Dialog, Frame`

setborderpos

Synopsis	procedure j_setborderpos (obj: integer; pos: integer) ;
Argumente	obj integer pos integer
Beschreibung	Positioniert die Komponente obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager ein- gestellt sein.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

setcolorbg

Synopsis procedure **j_setcolorbg** (obj: integer; r: integer; g,: integer; b:
integer) ;

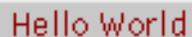
Argumente obj integer
 r integer
 g, integer
 b integer

Beschreibung Diese Prozedur setzt in dem Komponenten **obj** die Hintergrundfarbe. Als Parameter sind der **rot**, **grün** und **blau**-Anteil der Farbe zu übergeben. Unabhängig von der tatsächlichen vorhandenen Farbtiefe des Systems wird immer ein 24 Bit Farbwert angenommen. Dieser teilt sich in je 8 Bit für den Rot, den Grün und den Blau-Kanal auf. Somit können diese Parameter Werte zwischen 0 (dunkel) und 255 (hell) annehmen. Das Setzen einer neuen Hintergrundfarbe in einem Canvas bewirkt, das der Inhalt des Cavas überzeichnet wird.

Zielkomponenten Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

Beispiel

```
:
button := j_button(frame,'Hello World');
j_setcolorbg(button,150,0,0);
j_settext(button,'Hello World');
:
```



setcolor

Synopsis procedure **j_setcolor** (obj: integer; r: integer; g,: integer; b:
integer) ;

Argumente obj integer
 r integer
 g, integer
 b integer

Beschreibung Diese Prozedur setzt in der, durch **obj** referenzierten, Kompo-
nente die Vordergrundfarbe. Als Parameter sind der **rot**, **grün**
und **blau**-Anteil der Farbe zu übergeben. Unabhängig von der
tatsächlichen vorhandenen Farbtiefe des Systems wird immer ein
24 Bit Farbwert angenommen. Dieser teilt sich in je 8 Bit für den
Rot, den Grün und den Blau-Kanal auf. Somit können diese Pa-
rameter Werte zwischen 0 (dunkel) und 255 (hell) annehmen. Die
Vordergrundfarbe der Komponenten bleibt solange aktiv, bis diese
durch einen neuen Aufruf einer Farbfunktion überschrieben wird.

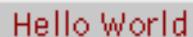
Zielkomponenten Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice,
Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window,
Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar,
Meter, Sevensegment

Beispiel

```

:
button := j_button(frame,'Hello World');
j_setcolor(button,150,0,0);
j_settext(button,'Hello World');
:

```



setcolumns

Synopsis `procedure j_setcolumns (obj: integer; columns: integer) ;`

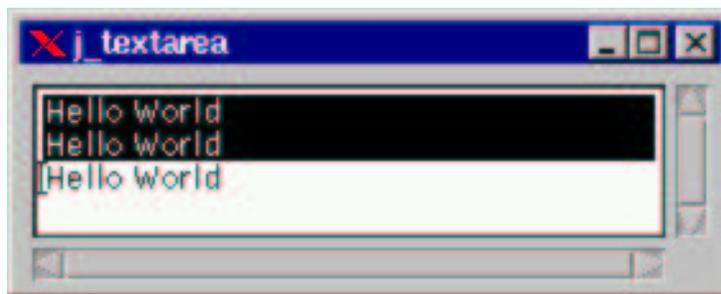
Argumente `obj` `integer`
 `columns` `integer`

Beschreibung Setzt die Anzahl Spalten von **obj** auf **columns**.

Zielkomponenten Textarea, Textfield, Gridlayout

Beispiel

```
:  
text := j_text(frame,10,4);  
j_setcolumns(text,30);  
:
```



setcurpos

Synopsis procedure **j_setcurpos** (obj: integer; pos: integer) ;

Argumente obj integer
 pos integer

Beschreibung Setzt den Text Cursor auf die Position **pos**.

Zielkomponenten Textarea, Textfield

setcursor

Synopsis	function j_setcursor (obj: integer; cursor: integer) : integer;
Argumente	obj integer cursor integer
Beschreibung	Setzt den Cursor der Komponenten obj auf cursor .
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment

setdanger

Synopsis `procedure j_setdanger (obj: integer; val: integer) ;`

Argumente `obj integer`
 `val integer`

Beschreibung Stellt den Warnlevel der Komponenten **obj** auf **val**.

Zielkomponenten Meter

setdebug

Synopsis	procedure j_setdebug (level: integer) ;
Argumente	level integer
Beschreibung	<p>Diese Prozedur setzt den Debuglevel auf level. Ab einem Level größer 0 erscheint ein Debugwindow, in dem die JAPI Calls protokolliert werden. Über den Parameter level kann die Protokolltiefe eingestellt werden. j_setdebug ist die einzige Funktion, die aufgerufen werden kann, bevor j_start() aufgerufen wurde. Die Levelbenen im einzelnen:</p> <ul style="list-style-type: none">0: keine Ausgabe (default Wert)1: Rückmeldung der konstruktiven Funktionen. Nur das Erzeugen der graphischen Objekte wird protokolliert.2: Wie 1, zusätzliche Ausgabe aller Aktionen, die vom Benutzer ausgeführt werden.3: Wie 2, zusätzlich werden alle weiteren Funktionen (außer den graphischen Befehlen) protokolliert.4: Wie 3, zusätzlich mit allen graphischen Befehlen.

setechochar

Synopsis	procedure j_setechochar (obj: integer; chr: char) ;
Argumente	obj integer chr char
Beschreibung	Bestimmt das Zeichen chr als Echo Zeichen für alle Benutzereingaben in die Komponente .
Zielkomponenten	Textfield

seteditable

Synopsis	procedure j_seteditable (obj: integer; bool: boolean) ;
Argumente	obj integer bool boolean
Beschreibung	Ist bool true , so ist die Komponente editierbar, ansonsten ist die Komponente nur lesbar für den Benutzer.
Zielkomponenten	Textarea, Textfield

setfixlayout

Synopsis	procedure j_setfixlayout (obj: integer) ;
Argumente	obj integer
Beschreibung	Integriert einen Fixlayout Manager in die Komponente obj (voreingestellte Layoutmanager).
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

setflowfill

Synopsis	procedure j_setflowfill (obj: integer; bool: boolean) ;
Argumente	obj integer bool boolean
Beschreibung	Setzt die Höhe (Breite) aller enthaltenen Komponenten auf die Höhe (Breite) von der Komponenten obj . Ein Flowlayout Manager wird vorausgesetzt.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

setflowlayout

Synopsis	procedure j_setflowlayout (obj: integer; align: integer) ;
Argumente	obj integer align integer
Beschreibung	Integriert einen Flowlayout Manager in die Komponente obj mit der Ausrichtung align .
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

setfocus

Synopsis	function j_setfocus (obj: integer) : integer;
Argumente	obj integer
Beschreibung	Weist den Input Focus der Komponenten obj zu.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

setfontname

Synopsis	procedure j_setfontname (obj: integer; name: integer) ;
Argumente	obj integer name integer
Beschreibung	Diese Prozedur setzt den Fonttyp in der Komponenten obj , auf name . Der Parameter name kann folgende Werte annehmen: <ul style="list-style-type: none"> • J_COURIER : Courier (Monospaced) • J_HELVETIA : Helvetia (Sans Serif) • J_TIMES : Times (Serif) • J_DIALOGIN : Dialog In • J_DIALOGOUT : Dialog Out
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popumenu

Beispiel

```

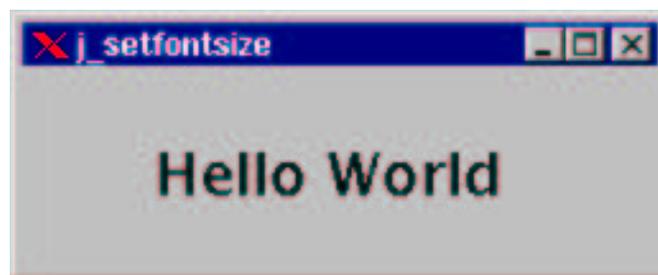
:
label := j_label(jframe,'Hello World');
j_setfontname(label,J_HELVETIA);
:

```



setfontsize

Synopsis	procedure j_setfontsize (obj: integer; size: integer) ;
Argumente	obj integer size integer
Beschreibung	Diese Prozedur setzt die Fontgröße in der Komponenten, obj , auf height . Die Größe size , kann prinzipiell beliebige positive Zahlen annehmen. Die Größe des Fonts hat die Einheit Pixel.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popumenu
Beispiel	<pre>: label := j_label(jframe,'Hello World'); j_setfontsize(label,24); :</pre>



setfontstyle

Synopsis	procedure j_setfontstyle (obj: integer; style: integer) ;
Argumente	obj integer style integer
Beschreibung	Diese Prozedur setzt den Fontstyle in der Komponenten, die durch obj referenziert wird, auf style . Für den Parameter style sind folgende Werte möglich: <ul style="list-style-type: none"> • J_PLAIN : normal • J_BOLD : fett • J_ITALIC : italic • J_BOLD + J_ITALIC : italic fett
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu
Beispiel	<pre> : label := j_label(jframe,'Hello World'); j_setfontstyle(label,J_BOLD+J_ITALIC); : </pre>



setfont

Synopsis procedure **j_setfont** (obj: integer; name: integer; style: integer;
size: integer) ;

Argumente obj integer
 name integer
 style integer
 size integer

Beschreibung Diese Prozedur setzt erzeugt einen neuen Font in der Komponente, die durch **obj** referenziert wird. Der Font wird durch die folgenden drei Parameter spezifiziert:

Der Parameter **name** kann folgende Werte annehmen:

- J_COURIER : Courier (Monospaced)
- J_HELVETIA : Helvetia (Sans Serif)
- J_TIMES : Times (Serif)
- J_DIALOGIN : Dialog In
- J_DIALOGOUT : Dialog Out

Für den Parameter **style** sind folgende Werte möglich:

- J_PLAIN : normal
- J_BOLD : fett
- J_ITALIC : italic
- J_BOLD + J_ITALIC : italic fett

Die Größe **size** wird als Integerzahl übergeben. Sie kann prinzipiell beliebige positive Zahlen annehmen.

Zielkomponenten Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu

Beispiel

```
:  
label := j_label(jframe,'Hello World');  
j_setfont(label,J_TIMES,J_PLAIN,18);  
:
```



setgridlayout

Synopsis	procedure j_setgridlayout (obj: integer; row: integer; col: integer) ;
Argumente	obj integer row integer col integer
Beschreibung	Integriert einen Gridlayout Manager in die Komponente obj mit row Reihen und col Spalten.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

sethgap

Synopsis	<code>procedure j_sethgap (obj: integer; hgap: integer) ;</code>				
Argumente	<table> <tr> <td style="padding-right: 20px;"><code>obj</code></td> <td><code>integer</code></td> </tr> <tr> <td><code>hgap</code></td> <td><code>integer</code></td> </tr> </table>	<code>obj</code>	<code>integer</code>	<code>hgap</code>	<code>integer</code>
<code>obj</code>	<code>integer</code>				
<code>hgap</code>	<code>integer</code>				
Beschreibung	<p>Setzt den horizontalen Abstand innerhalb der, durch obj referenzierten, Komponenten auf hgap Pixel. Die Komponente muss vom Typ Container sein (siehe Zielkomponenten). Weiterhin sollte in der Komponente ein Layoutmanager definiert sein, da diese Funktion nur in Verbindung mit einem Layoutmanager Sinn macht. Der Layoutmanager ordnet nun alle Objekte so an, daß deren horizontaler Abstand zueinander genau hgap Pixel beträgt.</p>				
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame				
Beispiel	<pre> : j_flowlayout(frame,J_HORIZONTAL); button1 := j_button(frame,'Button1'); button2 := j_button(frame,'Button2'); j_sethgap(frame,30); : </pre>				



seticon

Synopsis `procedure j_seticon (frame: integer; icon: integer) ;`

Argumente `frame integer`
 `icon integer`

Beschreibung Ordnet dem **frame** das Icon **icon** zu. Eine iconisierte Darstellung wird nicht von allen Plattformen unterstützt.

Zielkomponenten Frame

Beispiel

```
:  
frame := j_frame('Hello World');  
j_seticon(frame, j_loadimage('icon.gif'));  
:
```

setimage

Synopsis `procedure j_setimage (obj: integer; image: integer) ;`

Argumente `obj` `integer`
 `image` `integer`

Beschreibung Ordnet dem **obj** das Bild **image** zu.

Zielkomponenten `Graphicbutton, Graphiclabel`

Beispiel

```
:  
label := j_graphiclabel(frame,"mandel.gif");  
image := j_image("new.gif");  
j_setimage(label,image);  
:
```



setinsets

Synopsis procedure **j_setinsets** (obj: integer; top: integer; bottom: integer; left: integer; right: integer) ;

Argumente obj integer
 top integer
 bottom integer
 left integer
 right integer

Beschreibung Setzt den inneren Rahmen auf die angegebenen Größen.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
frame = j_frame('j_getinsets');
j_setinsets(frame,30,10,10,10);
:

```



setMax

Synopsis	function j_setmax (obj: integer; val: integer) : integer;
Argumente	obj integer val integer
Beschreibung	Stellt die obere Grenze der Komponenten auf den Wert von val ein.
Zielkomponenten	Scrollbar, Meter, Progress

setmin

Synopsis	function j_setmin (obj: integer; val: integer) : integer;
Argumente	obj integer val integer
Beschreibung	Stellt die untere Grenze der Komponenten auf den Wert von val ein.
Zielkomponenten	Scrollbar, Meter, Progress

setnamedcolorbg

Synopsis	procedure j_setnamedcolorbg (obj: integer; color: integer) ;
Argumente	obj integer color integer
Beschreibung	Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

setnamedcolor

Synopsis procedure **j_setnamedcolor** (obj: integer; color: integer) ;

Argumente obj integer
 color integer

Beschreibung Setzt die Vordergrundfarbe auf eine vordefinierte Farbe **color**.

Zielkomponenten Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice,
Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window,
Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar,
Meter, Sevenssegment

setnolayout

Synopsis	procedure j_setnolayout (obj: integer) ;
Argumente	obj integer
Beschreibung	Entfernt den aktuellen Layout Manager aus der Komponenten obj .
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

start

Synopsis `procedure j_start (port: integer) ;`

Argumente `port integer`

Beschreibung Diese Funktions ersetzt den default Port durch den übergebenen Parameter. Damit kann mehr als ein Kernel auf einem Rechner gestartet werden. Diese Funktion muss vor `j_start()` aufgerufen werden.

Beispiel

```
:  
j_setport(12345);  
if( not j_start ) then begin  
:
```

setpos

Synopsis	procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;
Argumente	obj integer xpos integer ypos integer
Beschreibung	Verschiebt die Komponente obj zur angegebenen Koordinate (xpos,ypos) an.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

setradiogroup

Synopsis	function j_setradiogroup (rbutton,: integer; rgroup: integer) : integer;
Argumente	rbutton, integer rgroup integer
Beschreibung	Weist dem Radiobutton rbutton die Radiogroup rgroup zu. Der Radiobutton wird aus der alten Radiogroup entfernt.
Zielkomponenten	Radiobutton

setresizable

Synopsis `procedure j_setresizable (obj: integer; resizable: boolean) ;`

Argumente `obj integer`
 `resizable boolean`

Beschreibung Ist **resizable** false , kann die Größe der Komponenten nicht vom Benutzer verändert werden.

Zielkomponenten Dialog, Frame

Beispiel

```
:  
frame := j_frame('fixsized Frame');  
j_setresizable(frame,false);  
:
```

setrows

Synopsis `procedure j_setrows (obj: integer; rows: integer) ;`

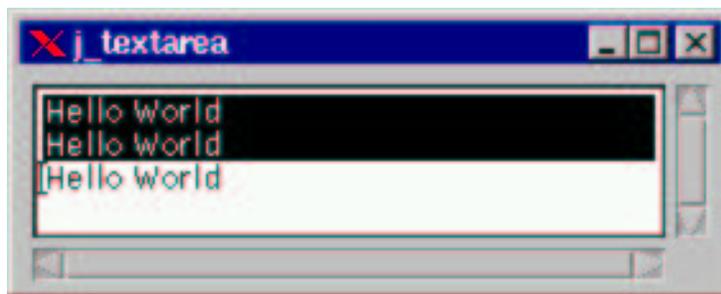
Argumente `obj integer`
 `rows integer`

Beschreibung Setzt die Anzahl Reihen von **obj** auf **rows**.

Zielkomponenten Textarea, GridLayout

Beispiel

```
:  
text := j_text(frame,30,10);  
j_setrows(text,4);  
:
```



setshortcut

Synopsis	procedure j_setshortcut (obj: integer; chr: char) ;
Argumente	obj integer chr char
Beschreibung	Bestimmt das Zeichen chr als Shortcut für die Komponente .
Zielkomponenten	MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu

setsize

Synopsis `procedure j_setsize (obj: integer; width: integer; height: integer) ;`

Argumente `obj integer`
 `width integer`
 `height integer`

Beschreibung Mit dieser Prozedur wird einer Komponente, die durch **obj** referenziert wird, eine neue Größe zugewiesen. Die Komponente erhält eine Breite von **width** Pixeln und eine Höhe von **height** Pixeln zugewiesen. Obwohl diese Prozedur prinzipiell auf alle Komponenten anwendbar ist, hat sie bei einigen Komponenten keine Auswirkung. So hat eine Choice beispielweise eine unveränderliche Höhe. Diese Höhe wird durch die Peer Komponente der jeweiligen Oberfläche vorgegeben. Eine weitere Ausnahme stellen die beiden Text Komponenten dar. Bei diesen Komponenten erfolgt die Größenangabe nicht in Bildschirmpunkten, sondern in der Anzahl der sichtbaren Textzeichen.

Zielkomponenten Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

Beispiel

```

:
button := j_button(frame,'Button');
j_setsize(button,100,100);
:

```



setslidesize

Synopsis `function j_setslidesize (obj: integer; val: integer) : integer;`

Argumente `obj integer`
 `val integer`

Beschreibung Setzt die Größe des Schiebers auf den Wert von **val**.

Zielkomponenten Scrollbar

setstate

Synopsis `procedure j_setstate (obj: integer; bool: boolean) ;`

Argumente `obj integer`
 `bool boolean`

Beschreibung Ist **bool** true so wird die Komponente selectiert.

Zielkomponenten Checkbox, Radiobutton, CheckmenuItem, Led

settext

Synopsis	procedure j_settext (obj: integer; str: PChar) ;
Argumente	obj integer str PChar
Beschreibung	Setzt den Textinhalt in der, durch obj referenzierten, Komponenten auf den Inhalt des Parameter str . Ist die Komponente vom Typ Frame oder Dialog, so wird der übergebene Text im Rahmen des Fensters eingetragen. Ansonsten wird der Textinhalt der Komponenten gesetzt. Der übergebene Text sollte keine Zeilenumbrüche enthalten, da fast alle Komponenten nur einzeilige Texte darstellen können. Einzige Ausnahme ist die Textarea Komponente, die mehrere Zeilen darstellen kann. Wird an diese Komponente ein Text übergeben, so darf dieser natürlich auch Zeilenumbrüche enthalten.
Zielkomponenten	Button, Label, Checkbox, Radiobutton, Dialog, Frame, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Textarea, Textfield
Beispiel	<pre> : button := j_button(frame, 'Hello World'); j_settext(button, 'Goodbye'); : </pre>



setunitinc

Synopsis function **j_setunitinc** (obj: integer; val: integer) : integer;

Argumente obj integer
 val integer

Beschreibung Stellt die minimale Wertänderung von die Komponente auf den Wert von **val** ein.

Zielkomponenten Scrollbar

setvalue

Synopsis	procedure j_setvalue (obj: integer; val: integer) ;
Argumente	obj integer val integer
Beschreibung	Stellt den aktuellen Wert der Komponenten obj auf val .
Zielkomponenten	Scrollbar, Progress, Meter, Sevensegment

setvgap

Synopsis `procedure j_setvgap (obj: integer; vgap: integer) ;`

Argumente `obj integer`
 `vgap integer`

Beschreibung Setzt den vertikalen Abstand innerhalb der, durch **obj** referenzier-
ten, Komponenten auf **vgap** Pixel. Die Komponente muss vom
Typ Container sein (siehe Zielkomponenten). Weiterhin sollte in
der Komponente ein Layoutmanager definiert sein, da diese Funk-
tion nur in Verbindung mit einem Layoutmanager Sinn macht. Der
Layoutmanager ordnet nun alle Objekte so an, daß deren vertika-
ler Abstand zueinander genau **vgap** Pixel beträgt.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:  
j_setflowlayout(frame,J_VERTICAL);  
button1 := j_button(frame,'Button1');  
button2 := j_button(frame,'Button2');  
j_setvgap(frame,30);  
:
```



setxor

Synopsis	procedure j_setxor (obj: integer; bool: boolean) ;
Argumente	obj integer bool boolean
Beschreibung	Schaltet den Painting Mode auf XOR, falls bool = true ist. Im XOR Modus heben sich zwei gleiche Graphikbefehle auf.
Zielkomponenten	Canvas, Image, Printer

sevenssegment

Synopsis function **j_sevenssegment** (obj: integer; color: integer) : integer;

Argumente obj integer
 color integer

Beschreibung Die Funktion erzeugt eine Siebensegment-Anzeige und liefert die Eventnummer der Komponente zurück. Die Farbe der Siebensegment-Anzeige kann eine der vordefinierten Farben sein (zB. J_RED, J_GREEN).

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:  
seven := j_sevenssegment(frame,J_GREEN);  
j_setvalue(seven,5);  
:
```



showpopup

Synopsis	procedure j_showpopup (obj: integer; xpos: integer; ypos: integer) ;
Argumente	obj integer xpos integer ypos integer
Beschreibung	Zeigt die Komponente an der Koordinate (xpos,ypos) an.
Zielkomponenten	Popupmenu

show

Synopsis	procedure j_show (obj: integer) ;
Argumente	obj integer
Beschreibung	Zeigt die Komponente obj auf dem Bildschirm an.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

sleep

Synopsis	function j_sleep (msec: integer) : integer;
Argumente	msec integer
Beschreibung	Unterbricht die Ausführung der Applikation für msec Millisekunden.

start

Synopsis `function j_start : boolean;`

Beschreibung Diese Funktion stellt eine Verbindung zu einem JAPI Kernel her, sofern auf der lokalen Maschine bereits ein Kernel läuft. Kann kein Kernel gefunden werden, so wird ein neuer Kernel gestartet und eine Verbindung aufgebaut. Schlägt dies auch fehl, so wird als Fehlercode `false` zurückgegeben

Beispiel

```
:
if( not j_start ) then begin
  writeln('cannot connect to JAPI Kernel');
  halt;
end;
:
```

sync

Synopsis `procedure j_sync ;`

Beschreibung Synchronisiert die Applikation mit dem JAPI Kernel.

textarea

Synopsis `function j_textarea (obj: integer; rows: integer; columns: integer) : integer;`

Argumente `obj integer`
 `rows integer`
 `columns integer`

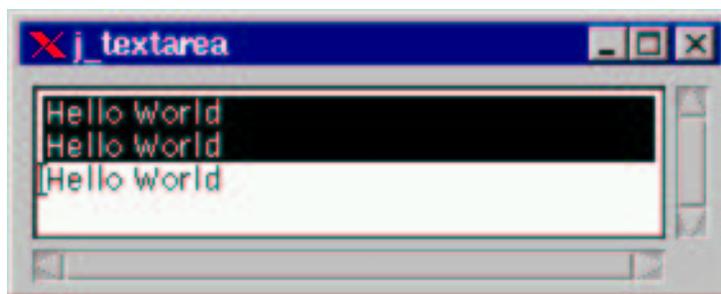
Beschreibung

Diese Funktion erzeugt eine neue Textarea und gibt dessen Eventnummer zurück. Im Fehlerfall wird `-1` zurückgegeben. Die Textarea hat eine sichtbare Größe von **rows** Reihen und **columns** Spalten, und ist somit abhängig vom voreingestellten Font. Soll eine Größe in Pixeleinheiten eingestellt werden, so kann nachträglich die Funktion `j_setsize()` verwendet werden. Die Textarea besitzt voreingestellt zwei Scrollbars, die sich automatisch anpassen. Eine Textarea liefert einen Event, sobald sich der Textinhalt ändert.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:
frame := j_frame('j_textarea');
text  := j_textarea(frame,30,4)
:
```



textfield

Synopsis `function j_textfield (obj: integer; columns: integer) : integer;`

Argumente `obj integer`
 `columns integer`

Beschreibung Diese Funktion erzeugt ein neues Textfeld und gibt dessen Eventnummer zurück. Im Fehlerfall wird `-1` zurückgegeben. Ein Textfeld dient für alle einzeiligen Eingaben. Das Textfeld hat eine sichtbare Größe von genau einer Reihe und **columns** Spalten, und ist somit abhängig vom voreingestellten Font. Soll eine Größe in Pixeleinheiten eingestellt werden, so kann nachträglich die Funktion `j_setsize()` verwendet werden. Ein Textfeld liefert ein Event, sobald die Eingabe mit **Return** abgeschlossen wird.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:
frame := j_frame('j_textfield');
text  := j_textfield(frame,30)
:
```



translate

Synopsis procedure **j_translate** (obj: integer; x: integer; y: integer) ;

Argumente obj integer
 x integer
 y integer

Beschreibung Setzt den Ursprung des Koordinatensystems auf (**x**, **y**).

Zielkomponenten Canvas, Image, Printer

vscrollbar

Synopsis `function j_vscrollbar (obj: integer) : integer;`

Argumente `obj` `integer`

Beschreibung Diese Funktion erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück. Im Fehlerfall wird `-1` zurückgegeben. Ist die Zielkomponente **obj** vom Typ `Scrollpane`, so wird die Eventnummer der vorhandenen vertikalen Scrollbar zurückgegeben. Eine Scrollbar erzeugt einen Event, sobald der Schieber bewegt wird.

Zielkomponenten `Panel`, `Borderpanel`, `Window`, `Dialog`, `Frame`, `Scrollpane`

Beispiel

```
:  
scroll:=j_vscrollbar(frame);  
j_setpos(scroll,120,40);  
j_setsize(scroll,20,100);  
:
```



windowlistener

Synopsis	function j_windowlistener (window: integer; kind: integer) : integer;
Argumente	window integer kind integer
Beschreibung	<p>Bindet an die Komponente window einen neuen Windowlistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird. Zulässige Werte für kind sind:</p> <ul style="list-style-type: none">• J_ACTIVATED : Der Componentlistener liefert einen Event, wenn die Komponente obj aktiviert wird.• J_DEACTIVATED : Der Componentlistener liefert einen Event, wenn die Komponente obj deaktiviert wird.• J_OPENED : Der Componentlistener liefert einen Event, wenn die Komponente obj geöffnet ist.• J_CLOSED : Der Componentlistener liefert einen Event, wenn die Komponente obj geschlossen ist.• J_ICONFIED : Der Componentlistener liefert einen Event, wenn die Komponente obj iconifiziert wird.• J_DEICONFIED : Der Componentlistener liefert einen Event, wenn die Komponente obj deiconifiziert wird.• J_CLOSING : Der Componentlistener liefert einen Event, wenn das close icon der Komponenten obj angeklickt wird.
Zielkomponenten	Window, Dialog, Frame

window

Synopsis	<code>function j_window (obj: integer) : integer;</code>
Argumente	<code>obj integer</code>
Beschreibung	Dies Funktion erzeugt ein neues Window und liefert dessen Eventnummer zurück. Im Fehlerfall wird <code>-1</code> zurückgeliefert. Ein Window ist ein einfaches eigenständiges Fenster, das keinen Fensterahmen besitzt. Wie ein Dialog, kann ein Window keine Menüleiste besitzen. Als Zielobjekt dieser Funktion ist nur ein Frame zulässig. Ein Window wird erst angezeigt, wenn ein entsprechendes <code>j_show()</code> Kommando ausgeführt wurde. Ein Window liefert keinen Event.

Zielkomponenten Frame

Beispiel

```

:
window := j_window(frame);
label  := j_label(window, 'Mouse pressed at ... ');
j_setnamedcolorbg(label, J_YELLOW);
:

```

Mouse pressed at 108:179