

GrCtrls

KOMPONENTY VIZUALIZAČNÍHO SYSTÉMU PRO GRAFICKÉ TERMINÁLY

Příručka uživatele a programátora



SofCon[®] spol. s r.o.
Střešovická 49
162 00 Praha 6
tel/fax: +420 220 180 454
E-mail: sofcon@sofcon.cz
www: <http://www.sofcon.cz>

Informace v tomto dokumentu byly pečlivě zkontrolovány a SofCon věří, že jsou spolehlivé, přesto SofCon nenese odpovědnost za případné nepřesnosti nebo nesprávnosti zde uvedených informací.

SofCon negarantuje bezchybnost tohoto dokumentu ani programového vybavení, které je v tomto dokumentu popsáno. Uživatel přebírá informace z tohoto dokumentu a odpovídající programové vybavení ve stavu, jak byly vytvořeny a sám je povinen provést validaci bezchybnosti produktu, který s použitím zde popsaného programového vybavení vytvořil.

SofCon si vyhrazuje právo změny obsahu tohoto dokumentu bez předchozího oznámení a nenese žádnou odpovědnost za důsledky, které z toho mohou vyplynout pro uživatele.

Datum vydání: 09.02.2004

Datum posledního uložení dokumentu: 09.02.2004

(Datum vydání a posledního uložení dokumentu musí být stejné)

Upozornění:

V dokumentu použité názvy výrobků, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Obsah :

1.O dokumentu	11
1.1. Revize dokumentu	11
1.2. Účel dokumentu	11
1.3. Rozsah platnosti	11
1.4. Související dokumenty	11
2.Termíny a definice	11
3.Úvod	12
3.1. Účel knihovny GrCtrls	12
4.Komponenty definované v jednotce GrCtrls	12
4.1. Komponenta TWindow	12
4.1.1. Položky a metody třídy TWindow	13
4.1.2. Implicitní nastavení komponenty TWindow	13
4.1.3. Události obsluhované komponentou TWindow	14
4.1.4. Oznámení generovaná komponentou TWindow	14
4.1.5. Validace hodnoty komponenty TWindow	15
4.1.6. Paleta komponenty TWindow	15
4.1.7. Reference	15
4.1.7.1. Konstruktor TWindow.Init	15
4.1.7.2. Metoda TWindow.InitFrame	16
4.2. Komponenta TDialog	16
4.2.1. Vyvolání dialogového okna v modálním stavu	16
4.2.2. Reference	17
4.2.2.1. Konstruktor TDialog.Init	17
4.2.2.2. Metoda TDialog.InitFrame	17
4.3. Komponenta TFrame	18
4.3.1. Implicitní nastavení komponenty TFrame	18
4.3.2. Události obsluhované komponentou TFrame	18
4.3.3. Oznámení generovaná komponentou TFrame	19
4.3.4. Validace hodnoty komponenty TFrame	19
4.3.5. Paleta komponenty TFrame	19
4.3.6. Reference	19
4.3.6.1. Konstruktor TFrame.Init	19
4.4. Komponenta TStaticText	20
4.4.1. Položky a metody třídy TStaticText	21
4.4.2. Implicitní nastavení komponenty TStaticText	22
4.4.3. Události obsluhované komponentou TStaticText	22
4.4.4. Oznámení generovaná komponentou TStaticText	22
4.4.5. Validace hodnoty komponenty TStaticText	23
4.4.6. Paleta komponenty TStaticText	23
4.4.7. Reference	23
4.4.7.1. Konstanty stf_	24
4.4.7.2. Položka TStaticText.Text	24
4.4.7.3. Položka TStaticText.BufferSize	24
4.4.7.4. Položka TStaticText.Offset	24
4.4.7.5. Položka TStaticText.Leading	25
4.4.7.6. Položka TStaticText.Flags	25
4.4.7.7. Položka TStaticText.TextFlags	25
4.4.7.8. Konstruktor TStaticText.Init	25

4.4.7.9.	Destruktor TStaticText.Done	26
4.4.7.10.	Metoda TStaticText.SetText	26
4.4.7.11.	Metoda TStaticText.GetText	27
4.4.7.12.	Metoda TStaticText.SetFlags	27
4.4.7.13.	Metoda TStaticText.SetTextParams	27
4.5.	Komponenta TProgressBar	28
4.5.1.	Položky a metody třídy TProgressBar	28
4.5.2.	Implicitní nastavení komponenty TProgressBar	29
4.5.3.	Události obsluhované komponentou TProgressBar	29
4.5.4.	Oznámení generovaná komponentou TProgressBar	30
4.5.5.	Validace hodnoty komponenty TProgressBar	30
4.5.6.	Paleta komponenty TProgressBar	30
4.5.7.	Reference	30
4.5.7.1.	Položka TProgressBar.Min	31
4.5.7.2.	Položka TProgressBar.Max	31
4.5.7.3.	Položka TProgressBar.Position	31
4.5.7.4.	Konstruktor TProgressBar.Init	31
4.5.7.5.	Metoda TProgressBar.SetPosition	31
4.5.7.6.	Metoda TProgressBar.SetRange	32
4.6.	Komponenta TImage	32
4.6.1.	Položky a metody třídy TImage	33
4.6.2.	Implicitní nastavení komponenty TImage	34
4.6.3.	Události obsluhované komponentou TImage	34
4.6.4.	Oznámení generovaná komponentou TImage	34
4.6.5.	Validace hodnoty komponenty TImage	35
4.6.6.	Paleta komponenty TImage	35
4.6.7.	Reference	35
4.6.7.1.	Konstanty imf_	35
4.6.7.2.	Položka TImage.Bitmap	36
4.6.7.3.	Položka TImage.Flags	36
4.6.7.4.	Konstruktor TImage.Init	36
4.6.7.5.	Metoda TImage.SetBitmap	36
4.6.7.6.	Metoda TImage.SetFlags	37
4.7.	Komponenta TEdit	37
4.7.1.	Položky a metody třídy TEdit	38
4.7.2.	Komponenta TEdit v modálním stavu	39
4.7.3.	Implicitní nastavení komponenty TEdit	40
4.7.4.	Události obsluhované komponentou TEdit	40
4.7.5.	Oznámení generovaná komponentou TEdit	41
4.7.6.	Validace hodnoty komponenty TEdit	41
4.7.7.	Paleta komponenty TEdit	42
4.7.8.	Reference	42
4.7.8.1.	Konstanty edf_	42
4.7.8.2.	Položka TEdit.Data	43
4.7.8.3.	Položka TEdit.MaxLen	43
4.7.8.4.	Položka TEdit.CurPos	43
4.7.8.5.	Položka TEdit.FirstPos	43
4.7.8.6.	Položka TEdit.InsMode	43
4.7.8.7.	Položka TEdit.Flags	43

4.7.8.8.	Položka TEdit.PasswordChar	44
4.7.8.9.	Konstruktor TEdit.Init	44
4.7.8.10.	Destruktor TEdit.Done	44
4.7.8.11.	Metoda TEdit.SetText	45
4.7.8.12.	Metoda TEdit.GetText	45
4.7.8.13.	Metoda TEdit.SetCurPos	45
4.7.8.14.	Metoda TEdit.SetFlags	46
4.7.8.15.	Metoda TEdit.SetInsMode	46
4.7.8.16.	Metoda TEdit.SetPasswordChar	46
4.7.8.17.	Metoda TEdit.Execute	47
4.7.8.18.	Funkce InlineEdit	47
4.7.8.19.	Funkce InlineEditEx	48
4.8.	Komponenta TButton	49
4.8.1.	Položky a metody třídy TButton	49
4.8.2.	Implicitní nastavení komponenty TButton	51
4.8.3.	Události obsluhované komponentou TButton	52
4.8.4.	Oznámení generovaná komponentou TButton	52
4.8.5.	Validace hodnoty komponenty TButton	52
4.8.6.	Paleta komponenty TButton	53
4.8.7.	Reference	53
4.8.7.1.	Konstanty btf_	53
4.8.7.2.	Položka TButton.Text	54
4.8.7.3.	Položka TButton.Flags	54
4.8.7.4.	Konstruktor TButton.Init	54
4.8.7.5.	Destruktor TButton.Done	55
4.8.7.6.	Metoda TButton.SetText	55
4.8.7.7.	Metoda TButton.SetFlags	56
4.9.	Komponenta TScrollBar	56
4.9.1.	Položky a metody třídy TScrollBar	57
4.9.1.1.	Význam příznaku sbfFixedPage	58
4.9.2.	Implicitní nastavení komponenty TScrollBar	58
4.9.3.	Události obsluhované komponentou TScrollBar	59
4.9.4.	Oznámení generovaná komponentou TScrollBar	60
4.9.5.	Validace hodnoty komponenty TScrollBar	60
4.9.6.	Paleta komponenty TScrollBar	60
4.9.7.	Reference	61
4.9.7.1.	Konstruktor TScrollBar.Init	61
4.10.	Komponenta TTrackBar	62
4.10.1.	Položky a metody třídy TTrackBar	62
4.10.2.	Implicitní nastavení komponenty TTrackBar	63
4.10.3.	Události obsluhované komponentou TTrackBar	64
4.10.4.	Oznámení generovaná komponentou TTrackBar	65
4.10.5.	Validace hodnoty komponenty TrackBar	65
4.10.6.	Paleta komponenty TTrackBar	65
4.10.7.	Reference	66
4.10.7.1.	Konstanty trf_	66
4.10.7.2.	Položka TTrackBar.Min	66
4.10.7.3.	Položka TTrackBar.Max	67
4.10.7.4.	Položka TTrackBar.Step	67

4.10.7.5.	Položka TTrackBar.PageStep	67
4.10.7.6.	Položka TTrackBar.Frequency	67
4.10.7.7.	Položka TTrackBar.Position	67
4.10.7.8.	Položka TTrackBar.Flags	67
4.10.7.9.	Konstruktor TTrackBar.Init	68
4.10.7.10.	Metoda TTrackBar.SetRange	68
4.10.7.11.	Metoda TTrackBar.SetPosition	68
4.10.7.12.	Metoda TTrackBar.SetParams	69
4.10.7.13.	Metoda TTrackBar.SetFlags	69
4.10.7.14.	Metoda TTrackBar.SetFrequency	70
4.10.7.15.	Metoda TTrackBar.DoChange	70
4.11.	Komponenta TUpDown	70
4.11.1.	Položky a metody třídy TUpDown	71
4.11.2.	Implicitní nastavení komponenty TUpDown	72
4.11.3.	Události obsluhované komponentou TUpDown	72
4.11.4.	Oznámení generovaná komponentou TUpDown	73
4.11.5.	Validace hodnoty komponenty TUpDown	73
4.11.6.	Paleta komponenty TUpDown	73
4.11.7.	Reference	74
4.11.7.1.	Konstanty udf_	74
4.11.7.2.	Položka TUpDown.Flags	75
4.11.7.3.	Položka TUpDown.Position	75
4.11.7.4.	Položka TUpDown.Min	75
4.11.7.5.	Položka TUpDown.Max	75
4.11.7.6.	Položka TUpDown.Step	75
4.11.7.7.	Položka TUpDown.Link	75
4.11.7.8.	Konstruktor TUpDown.Init	76
4.11.7.9.	Metoda TUpDown.DoChange	76
4.11.7.10.	Metoda TUpDown.SetRange	76
4.11.7.11.	Metoda TUpDown.SetPosition	77
4.11.7.12.	Metoda TUpDown.SetParams	77
4.11.7.13.	Metoda TUpDown.SetFlags	78
4.11.7.14.	Metoda TUpDown.SetLink	78
4.12.	Komponenta TListBox	78
4.12.1.	Položky a metody třídy TListBox	79
4.12.2.	Implicitní nastavení komponenty TListBox	81
4.12.3.	Události obsluhované komponentou TListBox	82
4.12.4.	Oznámení generovaná komponentou TListBox	82
4.12.5.	Data zobrazovaná komponentou TListBox	83
4.12.5.1.	Použití validátoru jako zdroje dat	83
4.12.5.2.	Použití oznámení nmGetData jako zdroje dat	83
4.12.6.	Validace hodnoty komponenty TListBox	84
4.12.7.	Paleta komponenty TListBox	85
4.12.8.	Reference	85
4.12.8.1.	Konstanty lbf_	86
4.12.8.2.	Položka TListBox.Flags	86
4.12.8.3.	Položka TListBox.TopOffset	86
4.12.8.4.	Položka TListBox.LeftOffset	87
4.12.8.5.	Položka TListBox.RightOffset	87

4.12.8.6.	Položka TListBox.BottomOffset	87
4.12.8.7.	Položka TListBox.ItemHeight	87
4.12.8.8.	Položka TListBox.ColCount	87
4.12.8.9.	Položka TListBox.ScrollBar	87
4.12.8.10.	Položka TListBox.ItemCount	87
4.12.8.11.	Položka TListBox.TopItem	88
4.12.8.12.	Položka TListBox.Selected	88
4.12.8.13.	Konstruktor TListBox.Init	88
4.12.8.14.	Metoda TListBox.GetData	88
4.12.8.15.	Metoda TListBox.GetItemBounds	89
4.12.8.16.	Metoda TListBox.SetItemCount	89
4.12.8.17.	Metoda TListBox.SelectItem	90
4.12.8.18.	Metoda TListBox.SetFlags	90
4.12.8.19.	Metoda TListBox.SetColCount	91
4.12.8.20.	Metoda TListBox.SetOffsets	91
4.12.8.21.	Metoda TListBox.SetItemHeight	91
4.12.8.22.	Metoda TListBox.PaintItems	92
4.12.8.23.	Metoda TListBox.PaintAllItems	92
4.12.8.24.	Metoda TListBox.ItemAtPos	93
4.13.	Komponenta TListView	93
4.13.1.	Položky a metody třídy TListView	94
4.13.2.	Definování sloupců komponenty TListView	96
4.13.3.	Implicitní nastavení komponenty TListView	96
4.13.4.	Události obsluhované komponentou TListView	97
4.13.5.	Oznámení generovaná komponentou TListView	98
4.13.6.	Data zobrazovaná komponentou TListView	98
4.13.7.	Validace hodnoty komponenty TListView	99
4.13.8.	Paleta komponenty TListView	99
4.13.9.	Reference	100
4.13.9.1.	Konstanty lvf_	101
4.13.9.2.	Konstanty lcf_	101
4.13.9.3.	Struktura TListColumn	101
4.13.9.4.	Funkce NewListColumn	102
4.13.9.5.	Funkce FreeColumn	103
4.13.9.6.	Položka TListView.Flags	103
4.13.9.7.	Položka TListView.Columns	103
4.13.9.8.	Položka TListView.ItemHeight	103
4.13.9.9.	Položka TListView.TopOffset	103
4.13.9.10.	Položka TListView.BottomOffset	104
4.13.9.11.	Položka TListView.HeaderHeight	104
4.13.9.12.	Položka TListView.TopItem	104
4.13.9.13.	Položka TListView.Selected	104
4.13.9.14.	Položka TListView.SelColumn	104
4.13.9.15.	Položka TListView.ItemCount	104
4.13.9.16.	Položka TListView.ScrollBar	104
4.13.9.17.	Konstruktor TListView.Init	105
4.13.9.18.	Metoda TListView.SetItemCount	105
4.13.9.19.	Metoda TListView.SelectItem	105
4.13.9.20.	Metoda TListView.SetFlags	106

4.13.9.21.	Metoda TListView.SetOffsets	106
4.13.9.22.	Metoda TListView.SetItemHeight	107
4.13.9.23.	Metoda TListView.SetHeaderHeight	107
4.13.9.24.	Metoda TListView.GetItemBounds	107
4.13.9.25.	Metoda TListView.GetRowHeight	108
4.13.9.26.	Metoda TListView.GetColWidth	108
4.13.9.27.	Metoda TListView.GetHeaderHeight	109
4.13.9.28.	Metoda TListView.GetRowCount	109
4.13.9.29.	Metoda TListView.GetData	109
4.13.9.30.	Metoda TListView.DoSelect	110
4.13.9.31.	Metoda TListView.PaintItems	110
4.13.9.32.	Metoda TListView.PaintAllItems	111
4.13.9.33.	Metoda TListView.ItemAtPos	111
4.13.9.34.	Metoda TListView.UpdateScrollBar	112
4.14.	Komponenta TListEdit	112
4.14.1.	Definice položek (řádků)	113
4.14.2.	Implicitní nastavení komponenty TListEdit	114
4.14.2.1.	Reference	115
4.14.2.2.	Konstanty lif_	115
4.14.2.3.	Struktura TListEditItem	116
4.14.2.4.	Funkce NewListEditItem	116
4.14.2.5.	Funkce NewListEditItemEx	117
4.14.2.6.	Procedura FreeListEditItem	118
4.14.2.7.	Položka TListEdit.Items	118
4.14.2.8.	Konstruktor TListEdit.Init	118
4.14.2.9.	Metoda TListEdit.GetData	119
4.14.2.10.	Metoda TListEdit.DoSelect	119
4.14.2.11.	Metoda TListEdit.DoEdit	120
4.14.2.12.	Metoda TListEdit.GetItemByIndex	120
4.14.2.13.	Metoda TListEdit.GetItemCount	121
4.15.	Komponenta TNavigator	121
4.15.1.	Položky a metody třídy TNavigator	122
4.15.2.	Vytvoření seznamu panelů komponenty TNavigator	123
4.15.3.	Implicitní nastavení komponenty TNavigator	124
4.15.4.	Události obsluhované komponentou TNavigator	125
4.15.5.	Oznámení generovaná komponentou TNavigator	125
4.15.6.	Validace hodnoty komponenty TNavigator	125
4.15.7.	Paleta komponenty TNavigator	125
4.15.8.	Reference	126
4.15.8.1.	Konstanty naf_	126
4.15.8.2.	Konstanty npf_	126
4.15.8.3.	Struktura TNavPanel	127
4.15.8.4.	Funkce NewTextPanel	128
4.15.8.5.	Funkce NewBitmapPanel	128
4.15.8.6.	Procedure FreePanel	129
4.15.8.7.	Položka TNavigator.Panels	129
4.15.8.8.	Položka TNavigator.Pages	130
4.15.8.9.	Položka TNavigator.Flags	130
4.15.8.10.	Konstruktor TNavigator.Init	130

4.15.8.11.	Destruktor TNavigator.Done	130
4.15.8.12.	Metoda TNavigator.PanelAtPos	131
4.15.8.13.	Metoda TNavigator.ClickPanel	131
4.15.8.14.	Metoda TNavigator.SetFlags	132
4.15.8.15.	Metoda TNavigator.FindPanel	132
4.16.	Komponenta TKeypad	132
4.16.1.	Položky a metody třídy TKeypad	134
4.16.2.	Definice rozvržení kláves komponenty TKeypad	135
4.16.3.	Implicitní nastavení komponenty TKeypad	136
4.16.4.	Události obsluhované komponentou TKeypad	137
4.16.5.	Oznámení generovaná komponentou TKeypad	137
4.16.6.	Validace hodnoty komponenty TKeypad	137
4.16.7.	Paleta komponenty TKeypad	137
4.16.8.	Reference	138
4.16.8.1.	Konstanty kpf_	138
4.16.8.2.	Konstanty kpbf_	138
4.16.8.3.	Konstanty kpa_	139
4.16.8.4.	Funkce NewKeypadButton	139
4.16.8.5.	Funkce NewStdKeypadButton	140
4.16.8.6.	Procedura FreeKeypadButton	141
4.16.8.7.	Struktura TKeypadButtton	141
4.16.8.8.	Položka TKeypad.PriButtons	142
4.16.8.9.	Položka TKeypad.SecButtons	142
4.16.8.10.	Položka TKeypad.Flags	142
4.16.8.11.	Položka TKeypad.BitmapGetter	142
4.16.8.12.	Položka TKeypad.AltState	143
4.16.8.13.	Konstruktor TKeypad.Init	143
4.16.8.14.	Destruktor TKeypad.Done	143
4.16.8.15.	Metoda TKeypad.PaintButton	144
4.16.8.16.	Metoda TKeypad.ClickButton	144
4.16.8.17.	Metoda TKeypad.GetButtonBitmap	144
4.16.8.18.	Metoda TKeypad.SetButtons	145
4.16.8.19.	Metoda TKeypad.SetFlags	145
4.16.8.20.	Metoda TKeypad.SetAltState	146
4.16.8.21.	Metoda TKeypad.IsButtonDown	146
4.16.8.22.	Metoda TKeypad.GetButtonKeyCode	146
4.16.8.23.	Metoda TKeypad.GetButtons	147
4.17.	Komponenta TMenuBar	147
4.17.1.	Položky a metody třídy TMenuBar	148
4.17.2.	Vytvoření struktury menu	149
4.17.3.	Implicitní nastavení komponenty TMenuBar	149
4.17.4.	Události obsluhované komponentou TMenuBar	150
4.17.5.	Oznámení generovaná komponentou TMenuBar	150
4.17.6.	Validace hodnoty komponenty TMenuBar	151
4.17.7.	Paleta komponenty TMenuBar	151
4.17.8.	Reference	151
4.17.8.1.	Konstruktor TMenuBar.Init	151
4.17.8.2.	Destruktor TMenuBar.Done	152
4.18.	Komponenta TMenuBar	152

4.18.1.	Reference	153
4.19.	Komponenta TMenuPopup	153
4.19.1.	Položky a metody třídy TMenuPopup	154
4.19.2.	Vytvoření struktury menu	155
4.19.3.	Vytvoření a inicializace komponenty TPopupMenu	155
4.19.4.	Implicitní nastavení komponenty TMenuPopup	155
4.19.5.	Události obsluhované komponentou TMenuPopup	156
4.19.6.	Oznámení generovaná komponentou TMenuPopup	156
4.19.7.	Validace hodnoty komponenty TMenuPopup	157
4.19.8.	Paleta komponenty TMenuPopup	157
4.19.9.	Reference	157
4.19.9.1.	Konstruktor TMenuPopup.Init	158
4.19.9.2.	Destruktor TMenuPopup.Done	158
4.19.9.3.	Procedure TMenuPopup.Popup	158
4.20.	Komponenta TMenuOverlay	159
4.20.1.	Položky a metody třídy TMenuOverlay	160
4.20.2.	Vytvoření struktury menu	161
4.20.3.	Implicitní nastavení komponenty TMenuOverlay	161
4.20.4.	Události obsluhované komponentou TMenuOverlay	162
4.20.5.	Oznámení generovaná komponentou TMenuOverlay	162
4.20.6.	Validace hodnoty komponenty TMenuOverlay	162
4.20.7.	Paleta komponenty TMenuOverlay	163
4.20.8.	Reference	163
4.20.8.1.	Konstanty mof_	164
4.20.8.2.	Položka TMenuOverlay.Flags	164
4.20.8.3.	Položka TMenuOverlay.ItemHeight	164
4.20.8.4.	Konstruktor TMenuOverlay.Init	164
4.20.8.5.	Metoda TMenuOverlay.SetFlags	165
4.20.8.6.	Metoda TMenuOverlay.SetItemHeight	165

1. O dokumentu

1.1. Revize dokumentu

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00	1.00	Cr	09.02.2004	První vydání

1.2. Účel dokumentu

Tento dokument slouží jako popis knihovny GrCtrls, která je součástí balíku vizualizačních knihoven pro jednotku KIT.

1.3. Rozsah platnosti

Určen pro programátory a uživatele programového vybavení SofCon.

1.4. Související dokumenty

Pro čtení tohoto dokumentu je potřeba seznámit se s manuálem Controls, Fonts, Bitmaps a příp. IoDrv.

Popis formátu verze knihovny a souvisejících funkcí je popsán v manuálu LibVer.

2. Termíny a definice

Používané termíny a definice jsou popsány v samostatném dokumentu Termíny a definice.

3. Úvod

3.1. Účel knihovny GrCtrls

Knihovna GrCtrls obsahuje sady komponent vizualizačního systému pro grafické terminály fy SofCon.

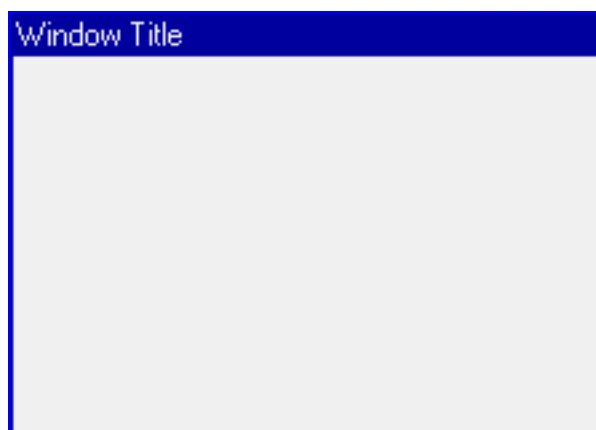
4. Komponenty definované v jednotce GrCtrls

V následujících kapitolách jsou popsány jednotlivé komponenty definované v knihovně GrCtrls. Pro snadnou orientaci je popis všech komponent strukturován naprosto stejně. Popis každé komponenty až na výjimky se skládá s následujících částí:

- Základní popis a použití komponenty
- Popis specifických položek a metod třídy komponenty
- Implicitní nastavení komponenty jejím vlastním konstruktorem
- Popis vstupních události obsluhovaných komponentou
- Popis oznámení generovaných komponentou
- Popis možnosti použití validátorů
- Struktura palety barev komponenty
- Referenční část s popisem jednotlivých konstant, položek a metod

4.1. Komponenta TWindow

Komponenta **TWindow** je vytváří ohraničenou obdélníkovou oblast určenou jako kontejner pro další komponenty. Na následujícím obrázku je ukázka této komponenty.



Nedílnou součástí komponenty **TWindow** je komponenta **TFrame** (viz. kapitola 4.3) vložena automaticky do okna při volání konstrukturu, zcela překrývající oblast okna (jedná se o komponentu na pozadí okna), která vykresluje pozadí, rámeček a titulek okna. Komponenta **TWindow** sama sebe nevykresluje (neimplementuje metodu **Paint**).

Komponentu **TWindow** lze použít se všemi typy grafických terminálů jako nemodální okno s libovolným obsahem.

4.1.1. Položky a metody třídy TWindow

Třída **TWindow** má následující konstruktor:

```
constructor Init( const ABounds: TRect; const ATitle: string );
```

Popis parametrů konstruktoru je uveden v následující tabulce:

ABounds	Umístění a rozměry komponenty v rámci vlastníka
ATitle	Text titulku okna

Komponenta **TWindow** rozšiřuje báзовou třídu **TGroup** o položky uvedené v následující tabulce:

Frame	Odkaz na komponentu TFrame , tj. rámeček okna. Položka je inicializována metodou InitFrame a je určena pouze pro čtení.
Title	Odkaz na text titulku okna alokovaný na hromadě. Položka je inicializována parametrem konstruktoru a je určena pouze pro čtení. Pro její nastavení použijte metodu SetTitle .

Komponenta **TWindow** rozšiřuje báзовou třídu **TGroup** o metodu uvedené v následující tabulce:

SetTitle	Metoda pro nastavení textu titulku okna
-----------------	---

4.1.2. Implicitní nastavení komponenty TWindow

V následující tabulce jsou uvedena implicitní nastavení všech položek komponenty. Všechny ostatní položky, které nejsou v tabulce uvedené, jsou inicializovány na hodnotu 0, **nil**, apod.

Položka	Hodnota
Bounds	parametr ABounds konstruktoru
EventMask	\$FFFF
CaretSize	(1, 8)
State	sfVisible
Options	ofSelectable or ofSharedPalette or ofTopSelect
Font	fidDefault
Palette	#\$0F#\$0F#\$0F#\$00#\$00#\$0F#\$00#\$00#\$00#\$00#\$0F (sdílená paleta)
Title	kopie parametru ATitle konstruktoru alokovana na hromadě

Pro komponentu **TWindow** jsou definovány konstanty nastavení třídy uvedené v následující tabulce. Tyto konstanty lze předat jako parametr metodě **Customize** (viz.

manuál ke knihovně Controls).

Konstanta	Význam
ccWindow	Implicitní nastavení definované ovladačem použitého terminálu.

4.1.3. Události obsluhované komponentou TWindow

Komponenta **TWindow** obsluhuje vstupní události podle následující tabulky

Událost	Popis činnosti
evMouseDown	Kliknutí do oblasti naktivního okna provede jeho aktivaci.
evKeyDown	vkNext Výběr následující komponenty vložené do okna, tj. přesun následující komponenty do ohniska.
	vkPrev Výběr předchozí komponenty vložené do okna, tj. přesun předchozí komponenty do ohniska.

4.1.4. Oznámení generovaná komponentou TWindow

Komponenta generuje následující oznámení:

Oznámení	Okamžik vyvolání	Parametry oznámení
nmEnter	Komponenta se dostane do ohniska	-
nmExit	Komponenta se dostane mimo ohnisko	-
nmHide	Volání metody Hide	-
nmShow	Volání metody Show	-
nmEnable	Volání metody Enable	-
nmDisable	Volání metody Disable	-
nmEndModal	Před ukončením modálního stavu.	Položka ModalResult struktury oznámení obsahuje důvod ukončení modálního stavu (tj. konstantu mrXXX). Položka Accept obsahuje hodnotu True. Ukončení modálního stavu lze zamezit nastavením této položky na hodnotu False.
nmCanExit	Před opuštěním ohniska, pokud je nastaven příznak ofValidate v položce Options	Položka Accept struktury oznámení obsahuje hodnotu False. Opuštění ohniska lze zakázat nastavením této položky na hodnotu False.

4.1.5. Validace hodnoty komponenty TWindow

S komponentu **TWindow** nelze použít žádný typ validátoru.

4.1.6. Paleta komponenty TWindow

Konstruktor komponenty TWindow nastavuje implicitně paletu barev komponenty podle tabulky uvedené níže. Komponenta TWindow nemá implementovanou metodu **Paint** a proto vyžaduje, aby komponenty do ní vložené zcela pokrývaly oblast okna. Paletu komponenty **TWindow** využívá speciální komponenta **TFrame** (viz. kapitola 4.3). Komponenta **TFrame** je umístěna jako poslední (na pozadí) v seznamu komponenty **TWindow** a zcela ji překrývá.

#	Barva	Popis
1	\$0F	Pozadí
2	\$0F	Linka vnějšího rámečku vlevo a nahoře
3	\$0F	Linka vnějšího rámečku vpravo a dole
4	■ \$00	Linka vnitřního rámečku vlevo a nahoře (neaktivní okno)
5	■ \$00	Linka vnitřního rámečku vpravo a dole (neaktivní okno)
6	\$0F	Pozadí textu titulku (neaktivní okno)
7	■ \$00	Text titulku (neaktivní okno)
8	■ \$00	Linka vnitřního rámečku vlevo a nahoře (aktivní okno)
9	■ \$00	Linka vnitřního rámečku vpravo a dole (aktivní okno)
10	■ \$00	Pozadí textu titulku (aktivní okno)
11	\$0F	Text titulku (aktivní okno)

4.1.7. Reference

Třída **TWindow** je potomkem třídy **TCustomWindow** popsané v dokumentaci ke knihovně Controls.

```
PWindow = ^TWindow;
TWindow = object( TCustomWindow )
public
  constructor Init( const ABounds: TRect; const ATitle: string );
  procedure InitFrame; virtual;
end;
```

4.1.7.1. Konstruktor TWindow.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```
constructor Init( const ABounds: TRect; const ATitle: string );
```

Parametry:

ABounds Umístění a rozměry komponenty v rámci jejího vlastníka.
 ATitle Text zobrazovaný v titulku okna

Návratové hodnoty:

Konstruktor nevrací žádnou hodnotu.

Poznámky:

Konstruktor **Init** nastaví všechny položky instance na implicitní hodnoty, viz. tabulka v kapitole 4.1.2 a zavolá metodu `InitFrame`, která provede inicializaci rámečku okna.

4.1.7.2. Metoda `TWindow.InitFrame`

Metoda **InitFrame** provádí inicializaci komponenty rámečku okna.

```
procedure InitFrame; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Třída **TWindow** předefinováá abstraktní metodu **InitFrame** svého předka, tak že vytvoří instanci komponenty **TFrame** a ukazatel na ní uloží do položky `Frame`. Nakonec u takto vytvořené instanci zavolá metodu `Customize` s parametrem `ccFrame`.

4.2. Komponenta `TDialog`

Komponenta **TDialog** je varianta komponenty **TWindow** určená pro modální dialogová okna. Komponenta **TDialog** přidává navíc obsluhu kláves `vkEsc` a `vkEnter`, které ukončují modální stav okna s výsledkem `mrCancel` resp. `mrOk`. Jinak pro tuto komponentu platí vše co pro komponentu `TWindow` (viz. kapitola 4.1).

4.2.1. Vyvolání dialogového okna v modálním stavu

Komponenta v modálním stavu je komponenta na níž jsou přesměrovány všechny události ze vstupních zařízení. Zpracování vstupních události v modálním stavu zajišťuje metoda **TControl.Execute**. Potomci třídy `TControl` tuto metodu musí předefinovat, tak aby prováděla požadované zpracování vstupních události. Metoda **Execute** se nikdy nevolá přímo, ale vždy skrze metodu **TGroup.ExecControl** (viz. dokumentace ke knihovně `Controls`).

Vyvolání dialogového okna v modálním stavu se tedy provede pomocí následujícího kódu.

```
AnyGroup^.ExecControl( MyDialog );
```

Proměnná `MyDialog` musí obsahovat ukazatel na inicializovanou instanci třídy

TDialog. Proměnná AnyGroup musí obsahovat ukazatel na libovolnou komponentu, ve stromu komponent, která je potomkem třídy **TGroup**, tj. např. **TApplication**, **TPage** apod.

Metoda **ExecControl** se ukončí až v okamžiku, kdy komponenta **TDialog** nebo libovolná do ní vložená komponenta zavolá metodu **EndModal** ukončující modální stav.

Metoda **ExecControl** vrací hodnotu typu Word zadanou jako parametr při volání metody **EndModal**. Tato hodnota specifikuje důvod ukočení modálního stavu a obsahuje jednu z konstant s prefixem mr_ (např. mrOk, mrCancel), viz. dokumentace ke knihovně Controls.

4.2.2. Reference

Třída **TDialog** je potomkem třídy **TCustomDialog** popsané v dokumentaci ke knihovně Controls.

```

PDialog = ^TDialog;
TDialog = object( TCustomDialog )
public
  constructor Init( const ABounds: TRect; const ATitle: string );
  procedure InitFrame; virtual;
end;

```

4.2.2.1. Konstruktor TDialog.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```

constructor Init( const ABounds: TRect; const ATitle: string );

```

Parametry:

ABounds	Umístění a rozměry komponenty v rámci jejího vlastníka.
ATitle	Text zobrazovaný v titulku okna

Návratové hodnoty:

Konstruktor nevrací žádnou hodnotu.

Poznámky:

Konstruktor **Init** nastaví všechny položky instance na implicitní hodnoty, viz. tabulka v kapitole 4.1.2 a zavolá metodu **InitFrame**, která provede inicializaci rámečku okna.

4.2.2.2. Metoda TDialog.InitFrame

Metoda **InitFrame** provádí inicializaci komponenty rámečku okna.

```

procedure InitFrame; virtual;

```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Třída **TDialog** předefinovává abstraktní metodu **InitFrame** svého předka, tak že vytvoří instanci komponenty **TFrame** a ukazatel na ní uloží do položky **Frame**. Nakonec u takto vytvořené instanci zavolá metodu **Customize** s parametrem **ccFrame**.

4.3. Komponenta TFrame

Komponenta **TFrame** je interní komponenta určená pro zobrazení rámečku, titulku, příp. speciálních tlačítek okna příp. dialogu, tj. komponent **TWindow** a **TDialog**. Aplikace nikdy nevytváří přímo instance této komponenty. Komponenta **TFrame** je vytvořena a inicializována speciální metodou **InitFrame** třídy **TWindow**.

4.3.1. Implicitní nastavení komponenty TFrame

V následující tabulce jsou uvedena implicitní nastavení všech položek komponenty. Všechny ostatní položky, které nejsou v tabulce uvedené, jsou inicializovány na hodnotu 0, **nil**, apod.

Položka	Hodnota
Bounds	parametr ABounds konstrukturu
EventMask	\$FFFF
CaretSize	(1, 8)
State	sfVisible
Options	ofSharedPalette
Font	fidDefault
Palette	#\$00 (sdílená paleta)

Pro komponentu **TFrame** jsou definovány konstanty nastavení třídy uvedené v následující tabulce. Tyto konstanty lze předat jako parametr metodě **Customize** (viz. manuál ke knihovně **Controls**).

Konstanta	Význam
ccFrame	Implicitní nastavení definované ovladačem použitého terminálu.

4.3.2. Události obsluhované komponentou TFrame

Komponenta **TFrame** neobsahuje žádné vstupní události.

4.3.3. Oznámení generovaná komponentou TFrame

Komponenta generuje následující oznámení:

Oznámení	Okamžik vyvolání	Parametry oznámení
nmHide	Volání metody Hide	-
nmShow	Volání metody Show	-
nmEnable	Volání metody Enable	-
nmDisable	Volání metody Disable	-

4.3.4. Validace hodnoty komponenty TFrame

S komponentu **TMenuOverlay** nelze použít žádný typ validátoru.

4.3.5. Paleta komponenty TFrame

Komponenta **TFrame** využívá paletu svého vlastníka (tj. okna do kterého je vložena). Předpokládá rozložení barev v paletě dané následující tabulkou:

#	Popis
1	Pozadí
2	Linka vnějšího rámečku vlevo a nahoře
3	Linka vnějšího rámečku vpravo a dole
4	Linka vnitřního rámečku vlevo a nahoře (neaktivní okno)
5	Linka vnitřního rámečku vpravo a dole (neaktivní okno)
6	Pozadí textu titulku (neaktivní okno)
7	Text titulku (neaktivní okno)
8	Linka vnitřního rámečku vlevo a nahoře (aktivní okno)
9	Linka vnitřního rámečku vpravo a dole (aktivní okno)
10	Pozadí textu titulku (aktivní okno)
11	Text titulku (aktivní okno)

4.3.6. Reference

Třída **TFrame** je potomkem třídy **TCustomFrame** popsané v dokumentaci ke knihovně Controls.

```
PFrame = ^TFrame;
TFrame = object( TCustomFrame )
public
    constructor Init( const ABounds: TRect );
    procedure Paint( ACanvas: PCanvas ); virtual;
end;
```

4.3.6.1. Konstruktor TFrame.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```
constructor Init( const ABounds: TRect );
```

Parametry:

ABounds Umístění a rozměry komponenty v rámci jejího vlastníka.

Návratové hodnoty:

Konstruktor nevrací žádnou hodnotu.

Poznámky:

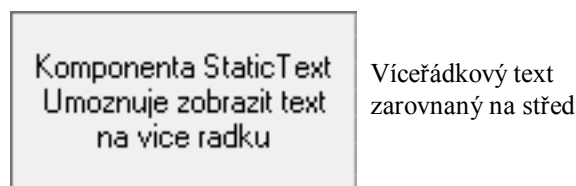
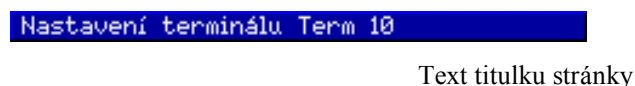
Konstruktor **Init** nastaví všechny položky instance na implicitní hodnoty, viz. tabulka v kapitole 4.3.1.

Pravděpodobně nikdy nebude potřeba volat konstruktor třídy **TFrame** přímo z aplikace. Konstruktor třídy **TFrame** je volán automaticky při vytváření okna nebo dialogu při volání metody **InitFrame** (viz. kapitola 4.1.7.2 a 4.2.2.2).

Obdélník daný parametrem ABounds, tj. umístění a rozměry komponenty TFrame, musí mít levý horní roh roven bodu (0, 0) a zároveň pravý horní roh musí odpovídat rozměrům okna, do něhož je rámeček vložen.

4.4. Komponenta TStaticText

Komponenta **TStaticText** je určena pro zobrazení libovolného textu. Na následujících obrázcích jsou zobrazeny ukázky této komponenty.



U komponenty **TStaticText** lze nastavovat několik parametrů, které ovlivňují její chování, resp. vykreslování. Lze nastavit:

- Zarovnání textu doleva, doprava a na střed
- Zarovnání textu nahoru, dolů a na střed
- Zobrazení víceřádkového textu
- Odsazení textu o zadaný počet pixelů
- Zobrazení rámečku kolem textu

4.4.1. Položky a metody třídy TStaticText

Třída **TStaticText** má následující konstruktor:

```
constructor Init( const ABounds: TRect; const AText: string );
```

Popis parametrů konstruktoru je uveden v následující tabulce:

ABounds	Umístění a rozměry komponenty v rámci vlastníka
AText	Zobrazovaný text

Komponenta **TStaticText** rozšiřuje báзовou třídu **TControl** o položky uvedené v následující tabulce:

Flags	<p>Příznaky ovlivňující chování komponenty. Může obsahovat kombinaci následujících příznaků:</p> <table border="1"> <tr> <td>stfBorder</td> <td>Zobrazení rámečku kolem komponenty</td> </tr> </table> <p>Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetFlags.</p>	stfBorder	Zobrazení rámečku kolem komponenty												
stfBorder	Zobrazení rámečku kolem komponenty														
TextFlags	<p>Příznaky pro upřesnění formátování vykreslovaného textu. Položka může obsahovat následující kombinaci příznaků:</p> <table border="1"> <tr> <td>tfLeft</td> <td>Zarovnání doleva</td> </tr> <tr> <td>tfRight</td> <td>Zarovnání doprava</td> </tr> <tr> <td>tfCenterX</td> <td>Zarovnání na střed osy X</td> </tr> <tr> <td>tfTop</td> <td>Zarovnání nahoru</td> </tr> <tr> <td>tfBottom</td> <td>Zarovnání dolů</td> </tr> <tr> <td>tfCenterY</td> <td>Zarovnání na střed osy Y</td> </tr> <tr> <td>tfMultiLine</td> <td>Zobrazení textu na více řádek</td> </tr> </table> <p>Položka je určena pouze pro čtení. Pro nastavení použijte metodu SetTextParams.</p>	tfLeft	Zarovnání doleva	tfRight	Zarovnání doprava	tfCenterX	Zarovnání na střed osy X	tfTop	Zarovnání nahoru	tfBottom	Zarovnání dolů	tfCenterY	Zarovnání na střed osy Y	tfMultiLine	Zobrazení textu na více řádek
tfLeft	Zarovnání doleva														
tfRight	Zarovnání doprava														
tfCenterX	Zarovnání na střed osy X														
tfTop	Zarovnání nahoru														
tfBottom	Zarovnání dolů														
tfCenterY	Zarovnání na střed osy Y														
tfMultiLine	Zobrazení textu na více řádek														
Offset	<p>Posunutí horního levého okraje textu od levého horního okraje komponenty v pixelech. Položka je určena pouze pro čtení. Pro nastavení použijte metodu SetTextParams.</p>														
Leading	<p>Mezera mezi jednotlivými řádkami při zobrazení textu na více řádek. Položka je určena pouze pro čtení. Pro nastavení použijte metodu SetTextParams.</p>														

Položky **Flags**, **TextFlags**, **Offset** a **Leading** lze nastavit při vytváření komponenty **TStaticText** před jejím vložením do skupiny vlastníka.

Komponenta **TStaticText** rozšiřuje báзовou třídu **TControl** o metodu uvedené v následující tabulce:

SetText	Metoda pro nastavení zobrazovaného textu
GetText	Metoda pro čtení zobrazovaného textu
SetFlags	Metoda pro nastavení položky Flags
SetTextParams	Metoda pro nastavení položek TextFlags , Offset a Leading

4.4.2. Implicitní nastavení komponenty TStaticText

V následující tabulce jsou uvedena implicitní nastavení všech položek komponenty. Všechny ostatní položky, které nejsou v tabulce uvedené, jsou inicializovány na hodnotu 0, nil, apod.

Položka	Hodnota
Bounds	parametr ABounds konstrukturu
EventMask	evMouseDown or evMouseDownbl or evMessage
CaretSize	(1, 8)
State	sfVisible
Options	ofSharedPalette
Font	fidDefault
Palette	#\$00 (sdílená paleta)
Text	kopie parametru AText konstrukturu
Flags	0
TextFlags	tfCenterY
Offset	0
Leading	0

Pro komponentu **TStaticText** jsou definovány konstanty nastavení třídy uvedené v následující tabulce. Tyto konstanty lze předat jako parametr metodě **Customize** (viz. manuál ke knihovně Controls).

Konstanta	Význam
ccStaticText	Implicitní nastavení definované ovladačem použitého terminálu.
ccStaticTextTitle	

4.4.3. Události obsluhované komponentou TStaticText

Komponenta **TStaticText** obsluhuje vstupní události podle následující tabulky

Událost	Popis činnosti
evMouseDown evMouseDownbl	Generování oznámení nmClick
evMessage	Nastavení textu komponenty. Položka Param struktury události musí obsahovat ukazatel na znakový řetězec.

4.4.4. Oznámení generovaná komponentou TStaticText

Komponenta generuje následující oznámení:

Oznámení	Okamžik vyvolání	Parametry oznámení
nmHide	Volání metody Hide	-

nmShow	Volání metody Show	-
nmEnable	Volání metody Enable	-
nmDisable	Volání metody Disable	-
nmClick	Kliknutí do oblasti komponenty	-

4.4.5. Validace hodnoty komponenty TStaticText

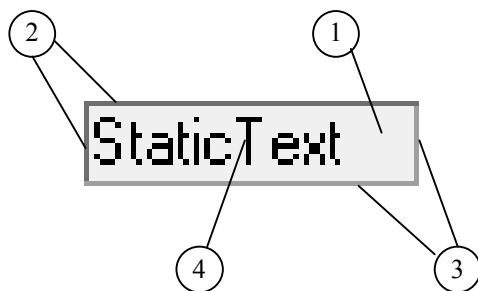
Přestože komponenta **TStaticText** neumožňuje editaci vlastní hodnoty, lze jí přiřadit validátor. Lze použít všechny typy validátorů, které implementují metodu `TransferText` s parametrem `vmLoad`, tj.: všechny potomky třídy **TTextValidator** a **TOrdinalValidator**.

Validátor je v případě této komponenty použit pouze jednosměrně a to k převodu hodnoty validované proměnné na zobrazovaný text .

4.4.6. Paleta komponenty TStaticText

Konstruktor komponenty **TStaticText** nastavuje implicitně paletu barev komponenty podle následující tabulky:

#	Barva	Popis
1	\$0F	Pozadí
2	■ \$00	Linka rámečku (vlevo a nahoře)
3	■ \$00	Linka rámečku (vpravo a dole)
4	■ \$00	Text



4.4.7. Reference

Třída **TStaticText** je potomkem třídy **TControl** popsané v dokumentaci ke knihovně Controls.

```

PStaticText = ^TStaticText;
TStaticText = object( TControl )
public
  Text      : PString;
  BuffSize  : Word;
  Offset    : TPoint;
  Leading   : Byte;
  Flags     : Byte;
  TextFlags : Byte;

```

```

constructor Init( const ABounds: TRect; const AText: string );
destructor Done; virtual;
procedure Paint( ACanvas: PCanvas ); virtual;
procedure SetText( const AText: string );
procedure GetText( var AText: string );
procedure SetFlags( ASet, AReset: Word ); virtual;
procedure SetTextParams( ATextFlags: Byte; AOffsetX,
    AOffsetY: Integer; ALeading: Byte );
procedure HandleEvent( var AEvent: TEvent ); virtual;
function Transfer( AMode: Integer ): Integer; virtual;
end;

```

4.4.7.1. Konstanty stf_

Konstanty s prefixem stf_ upřesňují chování komponenty **TStaticText**. Kombinace konstant je uložena v položce **Flags** (viz. kapitola 4.4.7.6).

V současné době je nedefinována jediná konstanta:

Identifikátor	Kód	Popis
stfBorder	\$0001	Pokud je nastaven tento příznak, pak metoda Paint vykresluje rámeček komponenty.

4.4.7.2. Položka TStaticText.Text

Položka **Text** obsahuje ukazatel na zobrazovaný znakový řetězec. Tento řetězec je alokovan na hromadě a je uvolněn destruktorem **Done**. Položka je interní. Pro její čtení slouží metoda **GetText** (viz. kapitola 4.4.7.11) a pro nastavení metoda **SetText** (viz. kapitola 4.4.7.10)

```
Text      : PString;
```

4.4.7.3. Položka TStaticText.BufferSize

Položka **BufferSize** udává prostor alokovaný na heapu pro zobrazovaný znakový řetězec. Jedná se o interní položku určenou pouze pro čtení.

```
BufferSize : Word;
```

4.4.7.4. Položka TStaticText.Offset

Položka **Offset** udává posunutí levého horního okraje zobrazovaného textu od levého horního okraje komponenty. Jedná se o strukturu **TPoint** se dvěma položkami **X** a **Y**. Posunutí v každé ose může být i záporné.

Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetTextParams** (viz. kapitola 4.4.7.13)

```
Offset    : TPoint;
```


4.4.7.5. Položka TStaticText.Leading

Položka **Leading** udává mezeru mezi řádky v případě zobrazení víceřádkového textu. V případě, že je tato položka nulová, pak řádky textu navazují bezprostředně a výška jednoho řádku je dána výškou znaků použitého fontu. Pokud je hodnota položky **Leading** nenulová, výška řádku je dána součtem výšky znaků použitého fontu a hodnoty této položky.

Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetTextParams** (viz. kapitola 4.4.7.13)

Leading : Byte;

4.4.7.6. Položka TStaticText.Flags

Položka **Flags** upřesňuje chování komponenty. Obsahuje kombinaci příznaků s prefixem **stf_** popsanych v kapitole 4.4.7.1. Položka je určena pouze pro čtení. Pro její nastavení slouží metodu **SetFlags** (viz. kapitola 4.4.7.12).

4.4.7.7. Položka TStaticText.TextFlags

Položka **TextFlags** upřesňuje chování rutiny pro vykreslení samotného textu komponenty. Položka obsahuje kombinaci příznaků s prefixem **tf_** definovaných v knihovně **Controls**.

Identifikátor	Kód	Popis
tfLeft	\$00	Zarovnání textu doleva
tfRight	\$01	Zarovnání textu doprava
tfCenterX	\$02	Zarovnání textu na střed komponenty v ose X
tfTop	\$00	Zarovnání textu nahoru
tfBottom	\$04	Zarovnání textu dolů
tfCenterY	\$08	Zarovnání textu na střed komponenty v ose Y
tfMultiline	\$10	Výpis textu na více řádků

Konstanty lze kombinovat pomocí logického součinu. Nelze však navzájem kombinovat konstanty **tfLeft**, **tfRight** a **tfCenterX** (resp. **tfTop**, **tfBottom** a **tfCenterY**).

4.4.7.8. Konstruktor TStaticText.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```
constructor Init( const ABounds: TRect; const AText: string );
```

Parametry:

ABounds Umístění a rozměry komponenty v rámci jejího vlastníka.
AText Zobrazovaný znakový řetězec.

Návratové hodnoty:

Konstruktor nevrací žádnou hodnotu.

Poznámky:

Konstruktor **Init** nastaví všechny položky instance na implicitní hodnoty, viz. tabulka v kapitole 4.4.2.

Na hromadě je alokován prostor pro zobrazovaný text do něhož je zkopírován text daný parametrem **AText**. Tento prostor na hromadě je uvolněn až voláním destrukturu **Done**.

4.4.7.9. Destruktor TStaticText.Done

Destruktor **Done** provádí uvolnění prostředku alokovaných konstruktorem instance..

```
destructor Done; virtual;
```

Parametry:

Destruktor nemá žádné parametry.

Návratové hodnoty:

Destruktor nevrací žádnou hodnotu.

Poznámky:

Destruktor **Done** uvolní z hromady buffer pro zobrazovaný text alokovaný konstruktorem.

4.4.7.10. Metoda TStaticText.SetText

Metoda **SetText** slouží ke změně zobrazovaného textu, tj. k nastavení položky **Text**.

```
procedure SetText( const AText: string );
```

Parametry:

AText Zobrazovaný znakový řetězec.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Metoda modifikuje buffer pro zobrazovaný text alokovaný na hromadě. Pokud je zadaný text delší, nebo výrazně kratší, než je alokovaná velikost bufferu, pak je provedena nová alokace.

4.4.7.11. Metoda TStaticText.GetText

Metoda **GetText** slouží k uložení zobraovaného textu do zadané proměnné.

```
procedure GetText( var AText: string );
```

Parametry:

AText Odkaz na proměnnou, do které bude uložen zobrazovaný text.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

4.4.7.12. Metoda TStaticText.SetFlags

Metoda **SetFlags** slouží k nastavení položky Flags (viz. kapitola 4.4.7.6)

```
procedure SetFlags( ASet, AReset: Word ); virtual;
```

Parametry:

AFlags Kombinace příznaků s prefixem stf_ (viz. kapitola 4.4.7.1), které se mají v položce Flags nastavit.

AReset Kombinace příznaků s prefixem stf_, které se mají v položce Flags vynulovat.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

Metoda nejprve vynuluje příznaky dané parametrem AReset a poté nastaví příznaky dané parametrem ASet. Metodu **SetFlags** lze tedy použít i pro přímé nastavení položky Flags. V takovém případě musí být parametr AReset nastaven na hodnotu \$FFFF a parametr ASet musí obsahovat požadovanou hodnotu položky Flags.

4.4.7.13. Metoda TStaticText.SetTextParams

Metoda **SetTextParams** slouží k nastavení položek TextFlags, Offset a Leading.

```
procedure SetTextParams( ATextFlags: Byte; AOffsetX,  
                          AOffsetY: Integer; ALeading: Byte );
```

Parametry:

ATextFlags Kombinace příznaků s prefixem stf_ (viz. kapitola 4.4.7.1)

	upřesňující rutiny pro vykreslování textu (viz. položka TextFlags v kapitole 4.4.7.7)
AOffsetX	Posunutí textu od levého okraje komponenty (viz. položka Offset v kapitole 4.4.7.4)
AOffsetY	Posunutí textu od horního okraje komponenty (viz. položka Offset v kapitole 4.4.7.4)
ALeading	Mezera mezi řádky textu (viz. položka Leading v kapitole 4.4.7.5)

Návratové hodnoty:

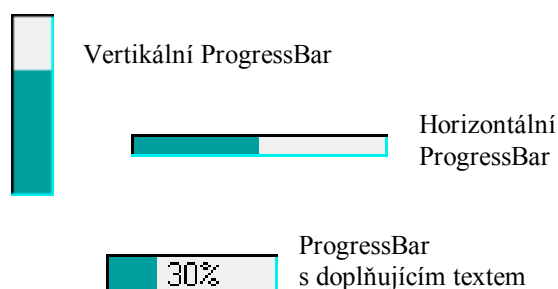
Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

4.5. Komponenta TProgressBar

Komponenta **TProgressBar** je určena pro grafické zobrazení hodnoty, příp. veličiny, jako je teplota, tlak, průběh déletrvající operace apod. Na následujících obrázcích jsou zobrazeny ukázky této komponenty.



4.5.1. Položky a metody třídy TProgressBar

Třída **TProgressBar** má následující konstruktor:

```
constructor Init( const ABounds: TRect );
```

Popis parametrů konstruktoru je uveden v následující tabulce:

ABounds	Umístění a rozměry komponenty v rámci vlastníka
----------------	---

Komponenta **TProgressBar** rozšiřuje báзовou třídu **TControl** o položky uvedené v následující tabulce:

Min	Dolní mez rozsahu. Položka je určena pouze pro čtení, pro nastavení použijte metodu SetRange .
------------	---

Max	Horní mez rozsahu. Položka je určena pouze pro čtení, pro nastavení použijte metodu SetRange .
Position	Aktuální zobrazovaná hodnota. Položka je určena pouze pro čtení, pro nastavení aktuální hodnoty použijte metodu SetPosition .

Komponenta **TProgressBar** rozšiřuje bázovou třídu **TControl** o metodu uvedené v následující tabulce:

SetPosition	Nastavení aktuální hodnoty (viz. kapitola 4.5.7.5)
SetRange	Nastavení mezí rozsahu (viz. kapitola 4.5.7.6)

4.5.2. Implicitní nastavení komponenty TProgressBar

V následující tabulce jsou uvedena implicitní nastavení všech položek komponenty. Všechny ostatní položky, které nejsou v tabulce uvedené, jsou inicializovány na hodnotu 0, **nil**, apod.

Položka	Hodnota
Bounds	parametr ABounds konstruktoru
EventMask	\$0000
CaretSize	(1, 8)
State	sfVisible
Options	ofSharedPalette
Font	fidDefault
Palette	#\$00 (sdílená paleta)
Min	0
Max	100
Position	50

Pro komponentu **TProgressBar** jsou definovány konstanty nastavení třídy uvedené v následující tabulce. Tyto konstanty lze předat jako parametr metodě **Customize** (viz. manuál ke knihovně Controls).

Konstanta	Význam
ccProgressBar	Implicitní nastavení definované ovladačem použitého terminálu.

4.5.3. Události obsluhované komponentou TProgressBar

Komponenta **TProgressBar** obsluhuje vstupní události podle následující tabulky

Událost	Popis činnosti
evMouseDown	Generování oznámení nmClick
evMouseDown	

4.5.4. Oznámení generovaná komponentou TProgressBar

Komponenta generuje následující oznámení:

Oznámení	Okamžik vyvolání	Parametry oznámení
nmHide	Volání metody Hide	-
nmShow	Volání metody Show	-
nmEnable	Volání metody Enable	-
nmDisable	Volání metody Disable	-
nmClick	Kliknutí do oblasti komponenty	-

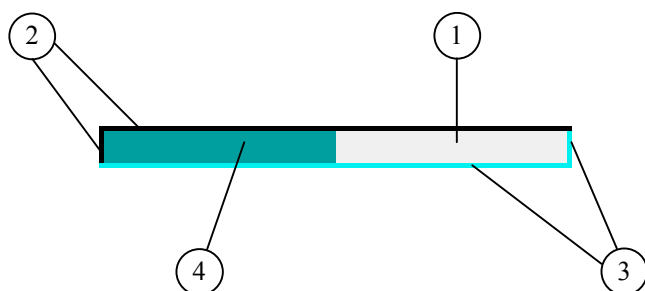
4.5.5. Validace hodnoty komponenty TProgressBar

Přestože komponenta **TProgressBar** neumožňuje editaci vlastní hodnoty, lze jí přiřadit validátor. Lze použít všechny typy validátorů, které implementují metodu **TransferOrdinal** s parametrem `vmLoad`, tj.: všechny potomky třídy **TOrdinalValidator**.

4.5.6. Paleta komponenty TProgressBar

Konstruktor komponenty **TProgressBar** nastavuje implicitně paletu barev komponenty podle následující tabulky:

#	Barva	Popis
1	\$0F	Pozadí
2	■ \$00	Linka rámečku (vlevo a nahoře)
3	■ \$00	Linka rámečku (vpravo a dole)
4	■ \$00	Výplň



4.5.7. Reference

Třída **TProgressBar** je potomkem třídy **TControl** popsané v dokumentaci ke knihovně Controls.

```

PProgressBar = ^TProgressBar;
TProgressBar = object( TControl )
public

```

```
Min      : Integer;  
Max      : Integer;  
Position : Integer;  
  
constructor Init( const ABounds: TRect );  
procedure Paint( ACanvas: PCanvas ); virtual;  
procedure SetPosition( APosition: Integer );  
procedure SetRange( AMin, AMax: Integer );  
procedure HandleEvent( var AEvent: TEvent ); virtual;  
function Transfer( AMode: Integer ): Integer; virtual;  
end;
```

4.5.7.1. Položka TProgressBar.Min

Položka **Min** obsahuje dolní mez rozsahu hodnoty komponenty. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetRange** (viz. kapitola 4.5.7.6)

```
Min      : Integer;
```

4.5.7.2. Položka TProgressBar.Max

Položka **Max** obsahuje horní mez rozsahu hodnoty komponenty. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetRange** (viz. kapitola 4.5.7.6)

```
Max      : Integer;
```

4.5.7.3. Položka TProgressBar.Position

Položka **Position** obsahuje aktuální hodnotu komponenty. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetPosition** (viz. kapitola 4.5.7.5)

```
Position : Integer;
```

4.5.7.4. Konstruktor TProgressBar.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```
constructor Init( const ABounds: TRect );
```

Parametry:

ABounds Umístění a rozměry komponenty v rámci jejího vlastníka.

Návratové hodnoty:

Konstruktor nevrací žádnou hodnotu.

Poznámky:

Konstruktor **Init** nastaví všechny položky instance na implicitní hodnoty, viz. tabulka v kapitole 4.5.2.

4.5.7.5. Metoda TProgressBar.SetPosition

Metoda **SetPosition** provádí nastavení hodnoty komponenty.

```
procedure SetPosition( APosition: Integer );
```

Parametry:

APosition Požadovaná nová hodnota komponenty.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu

Poznámky:

Metoda nastaví hodnotu komponenty, tj. položku Position na zadanou hodnotu. Pokud je nová hodnota mimo meze rozsahu, pak je nastavena jedna z krajního hodnot rozsahu.

Pokud se při volání metody **SetPosition** změní aktuální hodnota komponenty, pak je automaticky komponenta překreslena.

4.5.7.6. Metoda TProgressBar.SetRange

Metoda **SetRange** provádí nastavení mezí rozsahu komponenty, tj. položek Min a Max.

```
procedure SetRange( AMin, AMax: Integer );
```

Parametry:

AMin Dolní mez rozsahu komponenty
AMax Horní mez rozsahu komponenty

Návratové hodnoty:

Metoda nevrací žádnou hodnotu

Poznámky:

Metoda nastaví rozsah komponenty, tj. položky Min a Max na zadané hodnotu. Pokud je aktuální hodnota komponenty mimo tento rozsah, je upravena, tak aby ležela v tomto rozsahu. Po provedení metody je komponenta automaticky překreslena.

4.6. Komponenta TImage

Komponenta **TImage** je určena pro zobrazení obrázku (bitmapy) na zadaném místě na displeji terminálu. Na následujícím obrázku je ukázka této komponenty.



U komponenty **TImage** lze nastavovat několik parametrů, které ovlivňují její chování, příp. vykreslování. Lze nastavit:

- Zobrazení rámečku kolem komponenty
- Zarovnání bitmapy vlevo, vpravo nebo na střed
- Zarovnání bitmapy nahoru, dolů nebo na střed

4.6.1. Položky a metody třídy TImage

Třída **TImage** má následující konstruktor:

```
constructor Init( const ABounds: TRect; ABitmap: PBitmap );
```

Popis parametrů konstruktoru je uveden v následující tabulce:

ABounds	Umístění a rozměry komponenty v rámci vlastníka
ABitmap	Ukazatel na zobrazovanou bitmapu

Komponenta **TImage** rozšiřuje báзовou třídu **TControl** o položky uvedené v následující tabulce:

Bitmap	Zobrazovaná bitmapa. Tato položka je určena pouze pro čtení, pro změnu bitmapy použijte komponentu SetBitmap .	
Flags	Příznaky upřesňující chování komponenty. Položka může obsahovat následující kombinaci příznaků:	
	imfBorder	Zobrazení rámečku kolem bitmapy
	imfAlignLeft	Zarovnání bitmapy k levé straně komponenty
	imgAlignCenterX	Zarovnání bitmapy na střed osy X
	imfAlignRight	Zarovnání bitmapy k pravé straně komponenty
	imfAlignTop	Zarovnání bitmapy k horní straně komponenty
	imfAlignCenterY	Zarovnání bitmapy na střed osy Y
	imfAlignBottom	Zarovnání bitmapy k dolní straně komponenty
	Příznaky imfAlignLeft , imfAlignRight a imfAlignCenterX resp. imfAlignTop a imfAlignBottom a imfAlignCenterY nelze kombinovat. Položka Flags je určena pouze pro čtení. Pro její nastavení použijte metodu SetFlags .	

Komponenta **TImage** rozšiřuje báзовou třídu **TControl** o metodu uvedené

v následující tabulce:

SetBitmap	Metoda pro změnu zobrazované bitmapy
SetFlags	Metoda pro nastavení položky Flags.

4.6.2. Implicitní nastavení komponenty TImage

V následující tabulce jsou uvedena implicitní nastavení všech položek komponenty. Všechny ostatní položky, které nejsou v tabulce uvedené, jsou inicializovány na hodnotu 0, **nil**, apod.

Položka	Hodnota
Bounds	parametr ABounds konstrukturu
EventMask	evMouseDown or evMouseDownl
CaretSize	(1, 8)
State	sfVisible
Options	ofSharedPalette
Font	fidDefault
Palette	#\$0F#\$00#\$00 (sdílená paleta)
Bitmap	parametr ABitmap konstrukturu
Flags	imfAlignLeft or imgAlignTop

Pro komponentu **TImage** jsou definovány konstanty nastavení třídy uvedené v následující tabulce. Tyto konstanty lze předat jako parametr metodě **Customize** (viz. manuál ke knihovně Controls).

Konstanta	Význam
ccImage	Implicitní nastavení definované ovladačem použitého terminálu.

4.6.3. Události obsluhované komponentou TImage

Komponenta **TImage** obsluhuje vstupní události podle následující tabulky

Událost	Popis činnosti
evMouseDown evMouseDownl	Generování oznámení nmClick

4.6.4. Oznámení generovaná komponentou TImage

Komponenta generuje následující oznámení:

Oznámení	Okamžik vyvolání	Parametry oznámení
nmHide	Volání metody Hide	-
nmShow	Volání metody Show	-
nmEnable	Volání metody Enable	-

nmDisable	Volání metody Disable	-
nmClick	Kliknutí do oblasti komponenty	-

4.6.5. Validace hodnoty komponenty TImage

S komponentou **TImage** nelze použít žádný typ validátoru.

4.6.6. Paleta komponenty TImage

Konstruktor komponenty **TImage** nastavuje implicitně paletu barev komponenty podle následující tabulky:

#	Barva	Popis
1	\$0F	Pozadí
2	■ \$00	Linka rámečku (vlevo a nahoře)
3	■ \$00	Linka rámečku (vpravo a dole)

4.6.7. Reference

Třída **TImage** je potomkem třídy **TControl** popsané v dokumentaci ke knihovně Controls.

```

PImage = ^TImage;
TImage = object( TControl )
public
  Bitmap : PBitmap;
  Flags : Word;

  constructor Init( const ABounds: TRect; ABitmap: PBitmap );
  procedure Paint( ACanvas: PCanvas ); virtual;
  procedure SetBitmap( ABitmap: PBitmap );
  procedure SetFlags( ASet, AReset: Word ); virtual;
  procedure HandleEvent( var AEvent: TEvent ); virtual;
end;

```

4.6.7.1. Konstanty imf_

Konstanty s prefixem imf_ upřesňují chování komponenty **TImage**. Kombinace konstant je uložena v položce Flags této komponenty (viz. kapitola 4.6.7.3).

Identifikátor	Kód	Popis
imfBorder	\$0001	Zobrazení rámečku kolem bitmapy
imfAlignLeft	\$0000	Zarovnání bitmapy k levé straně komponenty
imfAlignCenterX	\$0002	Zarovnání bitmapy na střed osy X
imfAlignRight	\$0004	Zarovnání bitmapy k pravé straně komponenty
imfAlignTop	\$0000	Zarovnání bitmapy k horní straně komponenty
imfAlignCenterY	\$0008	Zarovnání bitmapy na střed osy Y
imfAlignBottom	\$0010	Zarovnání bitmapy k dolní straně komponenty

Příznaky `imfAlignLeft`, `imfAlignRight`, `imgAlignCenterX` resp. `imfAlignTop` a `imfAlignBottom` a `imfAlignCenterY` nelze kombinovat.

4.6.7.2. Položka `TImage.Bitmap`

Položka **Bitmap** obsahuje odkaz na strukturu vykreslované bitmapy, tj. strukturu **TBitmap** (viz. dokumentace ke knihovně `Bitmaps`). Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetBitmap** (viz. kapitola 4.6.7.5)

4.6.7.3. Položka `TImage.Flags`

Položka **Flags** upřesňuje chování komponenty. Obsahuje kombinaci příznaků s prefixem `imf_` popsáných v kapitole 4.6.7.1. Položka je určena pouze pro čtení. Pro její nastavení slouží metodu **SetFlags** (viz. kapitola 4.6.7.6).

4.6.7.4. Konstruktor `TImage.Init`

Konstruktor **Init** provádí inicializaci instance třídy.

```
constructor Init( const ABounds: TRect; ABitmap: PBitmap );
```

Parametry:

<code>ABounds</code>	Umístění a rozměry komponenty v rámci jejího vlastníka.
<code>ABitmap</code>	Ukazatel na strukturu bitmapy <code>TBitmap</code> (viz. dokumentace ke knihovně <code>Bitmaps</code>)

Návratové hodnoty:

Konstruktor nevrací žádnou hodnotu.

Poznámky:

Konstruktor **Init** nastaví všechny položky instance na implicitní hodnoty, viz. tabulka v kapitole 4.6.2.

4.6.7.5. Metoda `TImage.SetBitmap`

Metoda **SetBitmap** slouží k nastavení zobrazované bitmapy, tj. položky `Bitmap` (viz. kapitola 4.6.7.2)

```
procedure SetBitmap( ABitmap: PBitmap );
```

Parametry:

<code>ABitmap</code>	Ukazatel na strukturu bitmapy <code>TBitmap</code> (viz. dokumentace ke knihovně <code>Bitmaps</code>)
----------------------	---

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

4.6.7.6. Metoda TImage.SetFlags

Metoda **SetFlags** slouží k nastavení položky **Flags** (viz. kapitola 4.6.7.3)

```
procedure SetFlags( ASet, AReset: Word ); virtual;
```

Parametry:

AFlags Kombinace příznaků s prefixem **imf_** (viz. kapitola 4.6.7.1).

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:


Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

4.7. Komponenta TEdit

Komponenta **TEdit** je určena k editaci a zadávání textu, čísel apod. Komponenta zobrazuje zadávaný text a reaguje na stisk kláves vkládáním nebo přepisováním znaků. Na následujících obrázcích jsou zobrazeny ukázky této komponenty.

 Edit bez rámečku

 Edit se značkama

 Edit s textem zarovnaným doprava

U komponenty **TEdit** lze nastavovat několik parametrů, které ovlivňují její chování, příp. vykreslování. Lze nastavit:

- Zarovnání textu doleva nebo doprava
- Zobrazení značek oznamujících pokračování textu vlevo nebo vpravo
- Nahrazení zobrazovaných znaků specifikovaným znakem (např. hvězdičkou pro zadávání hesla)
- Zobrazení rámečku kolem textu

4.7.1. Položky a metody třídy TEdit

Třída **TEdit** má následující konstruktor:

```
constructor Init( const ABounds: TRect; AMaxLen: Integer );
```

Popis parametrů konstruktoru je uveden v následující tabulce:

ABounds	Umístění a rozměry komponenty v rámci vlastníka
AMaxLen	Maximální délka editovaného textu

Komponenta **TEdit** rozšiřuje báзовou třídu **TControl** o položky uvedené v následující tabulce:

Flags	Příznaky určující chování komponenty. Může obsahovat kombinaci následujících příznaků:	
	edfBorder	Zobrazení rámečku kolem komponenty
	edfMarks	Zobrazení značek oznamujících pokračování textu vlevo nebo vpravo.
	edfAlignRight	Zarovnání editovaného textu k pravému okraji komponenty.
	Položka Flags je určena pouze pro čtení. Pro její nastavení použijte metodu SetFlags .	
PasswordChar	Substituční znak pro zadávání hesla. Pokud je tato položka nastavena na hodnotu #0, pak jsou editovaný text zobrazen normálně. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetPasswordChar .	
MaxLen	Maximální délka editovaného textu. Položka je inicializována parametrem konstruktoru a je určena pouze pro čtení.	
CurPos	Aktuální pozice kurzoru. První znak je na pozici 1. Položka je určena pouze ke čtení. Pro nastavení aktuální pozice slouží metoda SetCurPos .	
InsMode	Nastavení této položky určuje zda je zvolen režim vkládání nebo přepisování znaků. Položka je určena pouze pro čtení. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetInsMode .	

Položky **Flags**, **PasswordChar**, **InsMode** lze nastavit při vytváření komponenty **TEdit** před jejím vložením do skupiny vlastníka.

Komponenta **TEdit** rozšiřuje báзовou třídu **TControl** o metodu uvedené v následující tabulce:

SetText	Metoda pro nastavení editovaného textu
GetText	Metoda pro čtení editovaného textu
SetCurPos	Metoda pro nastavení pozice kurzoru
SetFlags	Metoda pro nastavení položky Flags
SetInsMode	Metoda pro nastavení položky InsMode
SetPasswordChar	Metoda pro nastavení položky PasswordChar

4.7.2. Komponenta TEdit v modálním stavu

Komponenta v modálním stavu je komponenta na níž jsou přesměrovány všechny události ze vstupních zařízení. Zpracování vstupních události v modálním stavu zajišťuje metoda **TControl.Execute**. Potomci třídy TControl tuto metodu musí předefinovat, tak aby prováděla požadované zpracování vstupních události. Metoda **Execute** se nikdy nevolá přímo, ale vždy skrze metodu **TGroup.ExecControl** (viz. dokumentace ke knihovně Controls).

Vyvolání komponenty **TEdit** v modálním stavu lze provést např. následujícím kódem:

```
var
  R      : TRect;
  Edit   : TEdit;
  S      : string[40];

begin
  S := ...;
  R.Assign( ... );
  Edit := New( PEdit, Init( R, SizeOf(S) - 1 ) );
  Edit^.Customize( ccEdit );
  Edit^.SetText( S );

  if Group^.ExecControl( Edit ) = mrOk then
  begin
    Edit^.GetText( S );
    { Zpracování textu komponenty }
  end;

  Edit^.Free;
end;
```

Metoda **ExecControl** se ukončí v okamžiku, kdy uživatel stiskne klávesu vkEnter příp. vkEsc a vrátí hodnotu mrOk příp. mrCancel.

Pro zjednodušené vyvolání komponenty TEdit v modálním stavu existuje funkce **InlineEdit** (viz. kapitola 4.7.8.18).

```
function InlineEdit( AOwner: PGroup; ABounds: TRect; AClass: Word;
  AMaxLen: Integer; AValidator: PValidator ): Word;
```

Příklad:

```
var
  MyInt : Integer;

begin
  { Editace proměnné MyInt v decimálním tvaru v rozsahu 0 až 100 }

  MyInt := 50;

  if InlineEdit( Group, R, ccEdit, 10, New( PIntegerValidator,
```

```

    Init( @MyInt, nvfRadixDec, 0, 100 ) = mrOk then
  begin
    { Byla stisknuta klávesa vkEnter }
  end;
end;

```

4.7.3. Implicitní nastavení komponenty TEdit

V následující tabulce jsou uvedena implicitní nastavení všech položek komponenty. Všechny ostatní položky, které nejsou v tabulce uvedené, jsou inicializovány na hodnotu 0, **nil**, apod.

Položka	Hodnota
Bounds	parametr ABounds konstrukturu
EventMask	evKeyDown or evMouseDown or evMessage
CaretSize	nastavováno automaticky podle velikosti fontu a režimu
State	sfVisible or sfCursorVis
Options	ofSelectable or ofSharedPalette
Font	fidDefault
Palette	#\$0F#\$00#\$00#\$00 (sdílená paleta)
MaxLen	parametr AMaxLen konstrukturu
Flags	edfBorder
Data	ukazatel na buffer o délce MaxLen alokovaný na hromadě
FirstPos	1
CurPos	1
InsMode	True
PasswordChar	#\$00

Pro komponentu **TEdit** jsou definovány konstanty nastavení třídy uvedené v následující tabulce. Tyto konstanty lze předat jako parametr metodě **Customize** (viz. manuál ke knihovně Controls).

Konstanta	Význam
ccEdit	Implicitní nastavení definované ovladačem použitého terminálu.

4.7.4. Události obsluhované komponentou TEdit

Komponenta **TEdit** obsluhuje vstupní události podle následující tabulky:

Událost	Popis činnosti
evMouseDown	Výběr komponenty.
evMessage	cmSetText Nastavení editovaného textu. Položka Param struktury události musí obsahovat ukazatel na znakový řetězec.
evKeyDown	vkLeft Přesun kurzoru na znak o jednu pozici vlevo
	vkRight Přesun kurzoru na znak o jednu pozici vpravo

vkHome	Přesun kurzoru na pozici prvního znaku
vkEnd	Přesun kurzoru za pozici posledního znaku
vkDelete	Smazání znaku na pozici kurzoru
vkBackspace	Smazání znaku nalevo od kurzoru
vkInsert	Přepínání mezi režimem vkládání a přepisování
vkClear	Smazání celé řádky
vkEnter	Ukončení modálního stavu s výsledkem mrOk
vkEsc	Ukončení modálního stavu s výsledkem mrCancel
#32..#255	Vložení/přepis znaku na pozici kurzoru.

Klávesy vkEnter a vkEsc jsou obsluhovány pouze v případě, že je komponenta TEdit v modálním stavu.

4.7.5. Oznámení generovaná komponentou TEdit

Komponenta generuje následující oznámení:

Oznámení	Okamžik vyvolání	Parametry oznámení
nmEnter	Komponenta se dostane do ohniska	-
nmExit	Komponenta se dostane mimo ohnisko	-
nmHide	Volání metody Hide	-
nmShow	Volání metody Show	-
nmEnable	Volání metody Enable	-
nmDisable	Volání metody Disable	-
nmEndModal	Před ukončením modálního stavu.	Položka ModalResult struktury oznámení obsahuje důvod ukončení modálního stavu (tj. konstantu mrXXX). Položka Accept obsahuje hodnotu True. Ukončení modálního stavu lze zamezit nastavením této položky na hodnotu False.
nmCanExit	Před opuštěním ohniska, pokud je nastaven příznak ofValidate v položce Options	Položka Accept struktury oznámení obsahuje hodnotu False. Opuštění ohniska lze zakázat nastavením této položky na hodnotu False.

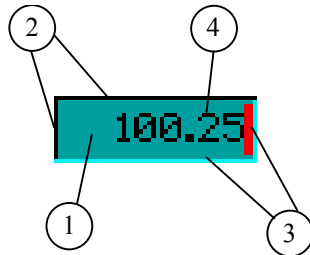
4.7.6. Validace hodnoty komponenty TEdit

S komponentu **TEdit** lze použít následující všechny typy validátorů implementujících metodu **TransferText**, tj. validátory odvozené ze tříd **TTextValidator**, **TNumericValidator**.

4.7.7. Paleta komponenty TEdit

Konstruktor komponenty **TEdit** nastavuje implicitně paletu barev komponenty podle následující tabulky:

#	Barva	Popis
1	\$0F	Pozadí
2	■ \$00	Linka rámečku (vlevo a nahoře)
3	■ \$00	Linka rámečku (vpravo a dole)
4	■ \$00	Text



4.7.8. Reference

Třída **TEdit** je potomkem třídy **TControl** popsané v dokumentaci ke knihovně Controls.

```

PEdit = ^TEdit;
TEdit = object( TControl )
public
    Data          : PString;
    MaxLen        : Integer;
    CurPos        : Integer;
    FirstPos      : Integer;
    InsMode       : Boolean;
    Flags         : Word;
    PasswordChar  : Char;

    constructor Init( const ABounds: TRect; AMaxLen: Integer );
    destructor Done; virtual;
    procedure Paint( ACanvas: PCanvas ); virtual;
    procedure HandleEvent( var AEvent: TEvent ); virtual;
    procedure SetState( AState: Word; AEnable: Boolean ); virtual;
    procedure SetText( const AText: string );
    procedure GetText( var AText: string );
    procedure SetCurPos( APos: Integer );
    procedure SetFlags( ASet, AReset: Word ); virtual;
    procedure SetInsMode( AInsMode: Boolean );
    procedure SetPasswordChar( AChar: Char );
    function Execute: Word; virtual;
    function Transfer( AMode: Integer ): Integer; virtual;
end;

```

4.7.8.1. Konstanty edf_

Konstanty s prefixem **edf_** upřesňují chování komponenty **TEdit**. Kombinace konstant je uložena v položce **Flags** této komponenty (viz. kapitola 4.7.8.7).

Identifikátor	Kód	Popis
edfBorder	\$0001	Zobrazení rámečku kolem bitmapy
edfMarks	\$0002	Zobrazení značek oznamujících pokračování textu vlevo nebo vpravo
edfAlignRight	\$0004	Zarovnání editovaného textu k pravému okraji komponenty

4.7.8.2. Položka TEdit.Data

Položka **Data** obsahuje ukazatel na buffer pro editovaný text o velikosti **MaxLen** (viz. kapitola 4.7.8.3) alokovaný konstruktorem třídy. Položka je určena pouze pro čtení.

```
Data          : PString;
```

4.7.8.3. Položka TEdit.MaxLen

Položka **MaxLen** obsahuje maximální počet znaků v buffer editovaného textu. Položka je inicializována parametrem konstrukturu a je určena pouze pro čtení.

```
MaxLen        : Integer;
```

4.7.8.4. Položka TEdit.CurPos

Položka **CurPos** obsahuje aktuální pozici kurzoru v editovaném textu. Rozsah položky je od 1 do počtu znaků textu v bufferu. Položka je určena pouze pro čtení. Lze ji nastavit pomocí metody **SetCurPos** (viz. kapitola 4.7.8.13).

```
CurPos       : Integer;
```

4.7.8.5. Položka TEdit.FirstPos

Položka **FirstPos** obsahuje pozici prvního vykreslovaného znaku textu v případě, že je text odrolován tak, že začátek textu je mimo oblast komponenty. Položka je nastavována automaticky a je určena pouze pro čtení.

```
FirstPos      : Integer;
```

4.7.8.6. Položka TEdit.InsMode

Položka **InsMode** je nastavena na hodnotu True, pokud je komponenta TEdit v režimu vkládání znaků. Pokud je tato položka nastavena na hodnotu False, pak je komponenta v režimu přepisování znaků. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetInsMode** (viz. kapitola 4.7.8.15)

```
InsMode       : Boolean;
```

4.7.8.7. Položka TEdit.Flags

Položka **Flags** upřesňuje chování komponenty. Obsahuje kombinaci příznaků s prefixem **edf_** popsanych v kapitole 4.7.8.1. Položka je určena pouze pro čtení. Pro její nastavení slouží metodu **SetFlags** (viz. kapitola 4.7.8.14).

```
Flags         : Word;
```

4.7.8.8. Položka TEdit.PasswordChar

Položka **PasswordChar** slouží k nastavení zástupného znaku pro speciální režim komponenty **TEdit** pro zadávání hesla. Obsahuje znak, jímž bude při vykreslování komponenty nahrazen každý jiný znak editovaného textu. Pokud je položka PasswordChar nastavena na hodnotu #0, potom komponenta vykresluje editovaný text normálně.

```
PasswordChar : Char;
```

4.7.8.9. Konstruktor TEdit.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```
constructor Init( const ABounds: TRect; AMaxLen: Integer );
```

Parametry:

ABounds	Umístění a rozměry komponenty v rámci jejího vlastníka.
AMaxLen	Maximální počet znaků editovaného textu..

Návratové hodnoty:

Konstruktor nevrací žádnou hodnotu.

Poznámky:

Konstruktor **Init** nastaví všechny položky instance na implicitní hodnoty, viz. tabulka v kapitole 4.7.3.

Konstruktor alokuje na hromadě buffer pro editovaný text o velikosti AMaxLen + 1 bajtů. Tento buffer je uvolněn až volání destrukturu **Done**.

4.7.8.10. Destruktor TEdit.Done

Destruktor **Done** provádí uvolnění prostředků alokovaných konstruktorem **Init**.

```
destructor Done; virtual;
```

Parametry:

Destruktor nemá žádné parametry.

Návratové hodnoty:

Konstruktor nevrací žádnou hodnotu.

Poznámky:

Destruktor uvolní z paměti bufferu pro editovaný text alokovaný konstruktorem.

4.7.8.11. Metoda TEdit.SetText

Metoda **SetText** slouží k nastavení editovaného textu.

```
procedure SetText( const AText: string );
```

Parametry:

AText Znakový řetězec, který bude překopírován do bufferu pro editovaný text. Pokud je tento řetězec delší, než je maximální délka editovaného textu daná položkou MaxLen, pak bude oříznut zprava.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

V případě potřeby metoda provede překreslení komponenty pomocí metody **Repaint**.

4.7.8.12. Metoda TEdit.GetText

Metoda **GetText** slouží k překopírování editovaného textu do připravené proměnné.

```
procedure GetText( var AText: string );
```

Parametry:

AText Odkaz na proměnná, do které bude překopírován aktuální obsah bufferu pro editovaný text.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

4.7.8.13. Metoda TEdit.SetCurPos

Metoda **SetCurPos** slouží k nastavení aktuální pozice kurzoru v editovaném textu, tj. položky CurPos (viz. kapitola 4.7.8.4)

```
procedure SetCurPos( APos: Integer );
```

Parametry:

APos Požadovaná pozice kurozru v textu. Hodnota se může

pohybovat v rozsahu 1 až do aktuálního počtu znaků editovaného textu.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

4.7.8.14. Metoda TEdit.SetFlags

Metoda **SetFlags** slouží k nastavení položky Flags (viz. kapitola 4.7.8.14)

```
procedure SetFlags( ASet, AReset: Word ); virtual;
```

Parametry:

AFlags Kombinace příznaků s prefixem edf_ (viz. kapitola 4.7.8.1).

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

4.7.8.15. Metoda TEdit.SetInsMode

Metoda **SetInsMode** slouží k režimu přepisování nebo vkládání znaků, tj. položky InsMode (viz. kapitola 4.7.8.6)

```
procedure SetInsMode( AInsMode: Boolean );
```

Parametry:

AInsMode Parametr určuje režim komponenty. Pokud je nastaven na hodnotu True, pak se jedná o režim vkládání znaků a v opačném případě o režim přepisování znaků.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

4.7.8.16. Metoda TEdit.SetPasswordChar

Metoda **SetPasswordChar** slouží k nastavení zástupného znaku pro speciální režim komponenty **TEdit** pro zadávání hesla, tj. nastavení položky PasswordChar (viz. kapitola 4.7.8.8).

```
procedure SetPasswordChar( AChar: Char );
```

Parametry:

AChar Zástupný znak. Obvykle se jedná o znak '*'. Pokud je tento parametr nastaven na hodnotu #0, pak jsou znaky editovaného textu zobrazeny normálně.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

V případě potřeby metoda provede překreslení komponenty pomocí metody **Repaint**.

4.7.8.17. Metoda TEdit.Execute

Metoda **Execute** uvede komponentu do modálního stavu.

```
function Execute: Word; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda vrací hodnotu reprezentující důvod ukončení modálního stavu, tj. buď konstanty mrOk, při stisku klávesy vkEnter, nebo mrCancel při stisku klávesy vkEsc.

Poznámky:

Třída **TEdit** předefinovává komponenty **Execute** třídy **TControl**, tak že spustí obsluhu událostí komponenty **TEdit** v modálním stavu. Tento modální stav je ukončen buď stiskem klávesy vkEnter nebo vkEsc.

Metodu **Execute** nikdy nevolejte přímo, ale vždy použijte metodu **ExecControl** třídy **TGroup** nebo jejího potomka.

4.7.8.18. Funkce InlineEdit

Funkce **InlineEdit** slouží k vytvoření komponenty **TEdit** a její spuštění v modálním stavu ve spojení s validátorem.

```
function InlineEdit( AOwner: PGroup; ABounds: TRect; AClass: Word;  
  AMaxLen: Integer; AValidator: PValidator ): Word;
```

Parametry:

AOwner	Vlastník komponenty TEdit , tj. komponenta do níž bude vytvořená komponenta TEdit vložena.
ABounds	Umístění a rozměry komponenty TEdit .
AClass	Třída nastavení komponenty, tj konstanty s prefixem cc_.
AMaxLen	Maximální délka počet znaků editační řádky. V případě zadání hodnoty nula, je maximální počet znaků nastaven dle použitého validátoru.
AValidator	Odkaz na vytvořenou instanci validátoru.

Návratové hodnoty:

Funkce vrací hodnotu reprezentující důvod ukončení modálního stavu komponenty, tj. buď konstanty `mrOk`, při stisku klávesy `vkEnter`, nebo `mrCancel` při stisku klávesy `vkEsc`.

Poznámky:

Funkce vytvoří komponentu **TEdit** s parametrem **ABounds** a **AMaxLen**. Tuto komponentu zkofigutruje pomocí metody **Customize** s parametrem **AClass**. Přiřadí jí identifikátor **cidInlineEdit** a spustí ji v modálním stavu pomocí metody **ExecControl** v komponentě **AGroup**.

V případě chyby validátoru komponenta vyvolá oznámení `nmError` (viz. dokumentace ke knihovně `Controls`). Toto oznámení může být obslouženo na úrovni komponenty daném parametrem **AGroup**.

4.7.8.19. Funkce InlineEditEx

Funkce **InlineEditEx** slouží k vytvoření komponenty **TEdit** a její spuštění v modálním stavu ve spojení s validátorem. Tato funkce rozšiřuje možnosti funkce **InlineEdit** o možnost zadání hodnoty položky **HelpCtx** vytvořené komponenty **TEdit**.

```
function InlineEditEx( AOwner: PGroup; ABounds: TRect; AClass: Word;
  AMaxLen: Integer; AHelpCtx: Word; AValidator: PValidator ): Word;
```

Parametry:

AOwner	Vlastník komponenty TEdit , tj. komponenta do níž bude vytvořená komponenta TEdit vložena.
ABounds	Umístění a rozměry komponenty TEdit .
AClass	Třída nastavení komponenty, tj konstanty s prefixem cc_.
AMaxLen	Maximální délka počet znaků editační řádky. V případě zadání hodnoty nula, je maximální počet znaků nastaven dle použitého validátoru.
AHelpCtx	Hodnota položky HelpCtx vytvořené komponenty TEdit .
AValidator	Odkaz na vytvořenou instanci validátoru.

Návratové hodnoty:

Funkce vrací hodnotu reprezentující důvod ukončení modálního stavu komponenty, tj. buď konstanty `mrOk`, při stisku klávesy `vkEnter`, nebo `mrCancel` při stisku klávesy `vkEsc`.

Poznámky:

Funkce vytvoří komponentu `TEdit` s parametrem `ABounds` a `AMaxLen`. Tuto komponentu zkofigutruje pomocí metody `Customize` s parametrem `AClass`. Přiřadí jí identifikátor `cidInlineEdit` a spustí ji v modálním stavu pomocí metody `ExecControl` v komponentě `AGroup`.

V případě chyby validátoru komponenta vyvolá oznámení `nmError` (viz. dokumentace ke knihovně `Controls`). Toto oznámení může být obslouženo na úrovni komponenty daném parametrem `AGroup`.

4.8. Komponenta `TButton`

Komponenta `TButton` implementuje tlačítko s množstvím nastavitelných vlastností. Na následujících obrázcích jsou zobrazeny ukázky této komponenty.



Komponenta `TButton` je primárně určena pro terminály s dotykovým panelem, lze ji však použít i u terminálu vybavených pouze klávesnicí.

U komponenty `TButton` lze nastavovat několik parametrů, které ovlivňují její chování, příp. vykreslování. Lze nastavit:

- Ostrými nebo zaoblené hrany tlačítka
- Funkci dvoustavového přepínače
- Reakci tlačítka při stisku nebo uvolnění
- Typ akce prováděné při stisku tlačítka

Při stisku tlačítka lze provést následující akce:

- Generování oznámení `nmClick`
- Emulaci stisku klávesy
- Ukončení modálního stavu
- Skok na definovanou stránku komponenty `TPageControl`
- Volání definované stránky komponenty `TPageControl`
- Návrat do předchozí stránky komponenty `TPageControl`

4.8.1. Položky a metody třídy `TButton`

Třída `TButton` má následující konstruktor:

```

constructor Init( const ABounds: TRect;
  const AText: string; AMode: Word; AParam: Word );

```

Popis parametrů konstruktoru je uveden v následující tabulce:

ABounds	Umístění a rozměry komponenty v rámci vlastníka	
AText	Text zobrazený uvnitř tlačítka	
AMode	Příznaky upřesňující chování tlačítka. Parametr je kombinací následujících příznaků:	
	btmNotify ^{*)}	Vyvolání oznámení nmClick při stisku tlačítka.
	btmKeyDown ^{*)}	Emulace stisku klávesy, jejíž kód je uveden v parametru AParam
	btmEndModal ^{*)}	Ukončení modálního stavu s výsledkem uvedeným v parametru AParam.
	btmGoto ^{*)}	Volání metody GotoPage komponenty TPageControl uvedené v položce Pages s parametrem AParam.
	btmCall ^{*)}	Volání metody CallPage komponenty TPageControl uvedené v položce Pages s parametrem AParam.
	btmReturn ^{*)}	Volání metody ReturnPage komponenty TPageControl uvedené v položce Pages.
	btmNext ^{*)}	Volání metody GotoNextPage komponenty TPageControl uvedené v položce Pages.
	btmPrev ^{*)}	Volání metody GotoPrevPage komponenty TPageControl uvedené v položce Pages.
	btmPush	Pokud je nastaven příznak btmPush, tlačítko reaguje na stisk. V opačném případě reaguje až na uvolnění.
btmSwitch	V případě, že je nastaven příznak btmSwitch, tlačítko se chová jako dvoustavový prepínač. Stav tlačítka je uchováván v položce Pressed.	
Příznaky označené *) nelze mezi sebou kombinovat.		
AParam	Doplňující parametr, obsahující buď identifikátor stránky, kód klávesy. Viz tabulka u parametru AMode.	

Třída **TButton** vychází ze třídy **TCustomButton** určené pro implementaci různých typů tlačítek. V následujícím textu jsou všechny položky a metody obou tříd společně.

Komponenta **TButton** rozšiřuje básovou třídu **TControl** o položky uvedené v následující tabulce:

Mode	Položka Mode upřesňuje chování tlačítka. Obsahuje kombinaci příznaků btm_ (viz. komponenta TCustomButton v knihovně Controls). Položka je inicializována parametrem konstruktoru a je určena pouze ke čtení.
Flags	Položka Flags upřesňuje vykreslování tlačítka. Obsahuje kombinaci následujících příznaků:

	btfRound	Vykreslení tlačítka se zaoblenými rohy. V případě, že tento příznak není nastaven, je tlačítko vykresleno s ostrými rohy.
		Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetFlags .
Text		Položka Text obsahuje ukazatel na text vypisovaný uvnitř tlačítka. Položka je určena pouze ke čtení. Pro nastavení textu použijte metodu SetText .
Pressed		Položka Pressed obsahuje stav tlačítka, pokud je v režimu dvoustavového přepínače (tj. v položce Mode je nastaven příznak btmSwitch). Pokud je tlačítko ve stavu stisknuto obsahuje tato položka hodnotu True. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetPressed .
Param		Položka Param obsahuje doplňující parametr k akci prováděné při stisku tlačítka. Položka je inicializovaná parametrem konstrukturu a je určena pouze pro čtení (viz. komponenta TCustomButton v knihovně Controls)
Pages		V případě, že akce při stisku tlačítka mění stav komponenty TPageControl (tj. provádí změnu stránky), pak tato položka musí obsahovat odkaz na správnou komponentu TPageControl.

Komponenta **TButton** rozšiřuje báзовou třídu **TControl** o metodu uvedené v následující tabulce:

SetText	Metoda slouží k nastavení textu zobrazovaného uvnitř tlačítka.
SetFlags	Metoda pro nastavení položky Flags
SetPressed	Metoda pro nastavení položky Pressed

4.8.2. Implicitní nastavení komponenty TButton

V následující tabulce jsou uvedena implicitní nastavení všech položek komponenty. Všechny ostatní položky, které nejsou v tabulce uvedené, jsou inicializovány na hodnotu 0, nil, apod.

Položka	Hodnota
Bounds	parametr ABounds konstrukturu
EventMask	\$FFFF
CaretSize	(1, 8)
State	sfVisible
Options	ofSelectable or ofSharedPalette or ofFirstClick
Font	fidDefault
Palette	#\$0F#\$00#\$00#\$0F#\$00#\$00#\$00#\$00#\$00#\$0F#\$00#\$00 (sdílená paleta)
Mode	podle parametru AMode konstrukturu
Param	podle parametru AParam konstrukturu
Pressed	False
Down	False

Flags	btfRound
Text	ukazatel na kopie parametru AText

Pro komponentu **TButton** jsou definovány konstanty nastavení třídy uvedené v následující tabulce. Tyto konstanty lze předat jako parametr metodě **Customize** (viz. manuál ke knihovně Controls).

Konstanta	Význam
ccButton	Implicitní nastavení definované ovladačem použitého terminálu.

4.8.3. Události obsluhované komponentou TButton

Komponenta **TButton** obsluhuje vstupní události podle následující tabulky

Událost	Popis činnosti
evMouseDown evMouseRep evMouseDown	Akce prováděná při stisku tlačítka.
evKeyDown vkEnter	Akce prováděná při stisku tlačítka.

4.8.4. Oznámení generovaná komponentou TButton

Komponenta generuje následující oznámení:

Oznámení	Okamžik vyvolání	Parametry oznámení
nmEnter	Komponenta se dostane do ohniska	-
nmExit	Komponenta se dostane mimo ohnisko	-
nmHide	Volání metody Hide	-
nmShow	Volání metody Show	-
nmEnable	Volání metody Enable	-
nmDisable	Volání metody Disable	-
nmClick	Kliknutí do oblasti komponenty	-

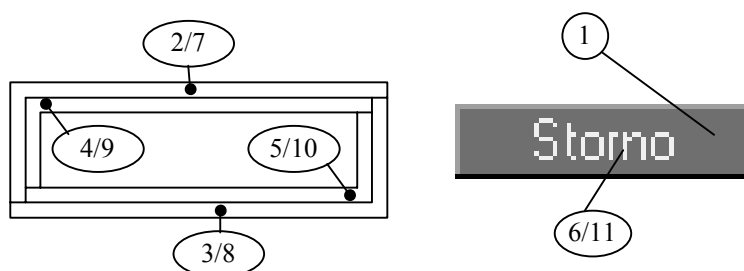
4.8.5. Validace hodnoty komponenty TButton

S komponentu **TButton** lze použít všechny typy validátorů, které implementují metodu **TransferOrdinal**. Validátor má význam pouze v případě, že je tlačítko nastaveno jako dvoustavový přepínač (tj. má nastaven příznak `btmSwitch` v položce `Mode`).

4.8.6. Paleta komponenty TButton

Konstruktor komponenty **TButton** nastavuje implicitně paletu barev komponenty podle následující tabulky:

#	Barva	Popis	Stav tlačítka
1	\$0F	Pozadí	
2	■ \$00	Linka vnějšího rámečku vlevo a nahoře	uvolněné
3	■ \$00	Linka vnějšího rámečku vpravo a dole	uvolněné
4	\$0F	Linka vnitřního rámečku vlevo a nahoře	uvolněné
5	■ \$00	Linka vnitřního rámečku vpravo a dole	uvolněné
6	■ \$00	Text	uvolněné
7	■ \$00	Linka vnějšího rámečku vlevo a nahoře	stisknuté
8	■ \$00	Linka vnějšího rámečku vpravo a dole	stisknuté
9	■ \$00	Linka vnitřního rámečku vlevo a nahoře	stisknuté
10	\$0F	Linka vnitřního rámečku vpravo a dole	stisknuté
11	■ \$00	Text	stisknuté
12	■ \$00	Obdélník pro zvýraznění ohniska	



4.8.7. Reference

Třída **TButton** je potomkem třídy **TCustomButton** popsané v dokumentaci ke knihovně Controls.

```

PButton = ^TButton;
TButton = object( TCustomButton )
public
  Text      : PString;
  Flags     : Word;

  constructor Init( const ABounds: TRect;
    const AText: string; AMode: Word; AParam: Word );
  destructor Done; virtual;
  procedure Paint( ACanvas: PCanvas ); virtual;
  procedure SetState( AState: Word; AEnable: Boolean ); virtual;
  procedure SetText( const AText: string );
  procedure SetFlags( ASet, AReset: Word ); virtual;
end;
```

4.8.7.1. Konstanty btf_

Konstanty s prefixem **btf_** upřesňují chování komponenty **TButton**. Kombinace konstant je uložena v položce **Flags** této komponenty (viz. kapitola 4.8.7.3).

Identifikátor	Kód	Popis
btfRound	\$0001	Pokud je nastaven tento příznak, pak je tlačítko je vykresleno se zaoblenými rohy. V opačném případě má tlačítko tvar obdelníku s ostrými rohy.

4.8.7.2. Položka TButton.Text

Položka **Text** obsahuje znakový řetězec zobrazený uvnitř tlačítka. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetText** (viz. kapitola 4.8.7.6)

```
Text : PString;
```

4.8.7.3. Položka TButton.Flags

Položka **Flags** upřesňuje chování komponenty. Obsahuje kombinaci příznaků s prefixem btf_ popsaných v kapitole 4.8.7.1. Položka je určena pouze pro čtení. Pro její nastavení slouží metodu **SetFlags** (viz. kapitola 4.8.7.7).

```
Flags : Word;
```

4.8.7.4. Konstruktor TButton.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```
constructor Init( const ABounds: TRect; const AText: string;
  Mode: Word; AParam: Word );
```

Parametry:

ABounds Umístění a rozměry komponenty v rámci jejího vlastníka.
AText Maximální počet znaků editovaného textu..
AMode Příznaky upřesňující chování tlačítka. Parametr je kombinací následujících příznaků:

btfNotify ^{*)}	Vyvolání oznámení OnClick při stisku tlačítka.
btfKeyDown ^{*)}	Emulace stisku klávesy, jejíž kód je uveden v parametru AParam
btfEndModal ^{*)}	Ukončení modálního stavu s výsledkem uvedeným v parametru AParam .
btfGoto ^{*)}	Volání metody GotoPage komponenty TPageControl uvedené v položce Pages s parametrem AParam .
btfCall ^{*)}	Volání metody CallPage komponenty TPageControl uvedené v položce Pages s parametrem AParam .
btfReturn ^{*)}	Volání metody ReturnPage komponenty TPageControl uvedené v položce Pages .
btfNext ^{*)}	Volání metody GotoNextPage komponenty TPageControl uvedené v položce Pages .
btfPrev ^{*)}	Volání metody GotoPrevPage komponenty TPageControl uvedené v položce Pages .
btfPush	Pokud je nastaven příznak btfPush , tlačítko

	reaguje na stisk. V opačném případě reaguje až na uvolnění.
btmSwitch	V případě, že je nastaven příznak btmSwitch, tlačítko se chová jako dvoustavový přepínač. Stav tlačítka je uchováván v položce Pressed.

Příznaky označené *) nelze mezi sebou kombinovat.

AParam

Doplňující parametr, obsahující buď identifikátor stránky, kód klávesy. Viz tabulka u parametru AMode.

Návratové hodnoty:

Konstruktor nevrací žádnou hodnotu.

Poznámky:

Konstruktor **Init** nastaví všechny položky instance na implicitní hodnoty, viz. tabulka v kapitole 4.8.2.

Konstruktor alokuje na hromadě kopii textu tlačítka. Tato kopie je uvolněna až při volání destrukturu **Done**.

4.8.7.5. Destruktor TButton.Done

Destruktor **Done** uvolňuje prostředky alokované konstruktorem instance třídy.

```
destructor Done; virtual;
```

Parametry:

Destruktor nemá žádné parametry.

Návratové hodnoty:

Destruktor nevrací žádnou hodnotu.

Poznámky:

Destruktor uvolní z paměti kopii textu tlačítka alokovanou na hromadě při volání konstrukturu instance.

4.8.7.6. Metoda TButton.SetText

Metoda **SetText** slouží k nastavení textu zobrazovaného uvnitř tlačítka.

```
procedure SetText( const AText: string );
```

Parametry:

AText Požadovaný text tlačítka.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Metoda uvolní z paměti stavající text tlačítka. Alokuje na hromadě prostor, do kterého zkopíruje znakový řetězec daný parametrem `AText`. V případě potřeby metoda provede překreslení komponenty metodou **Repaint**.

4.8.7.7. Metoda `TButton.SetFlags`

Metoda **SetFlags** slouží k nastavení položky `Flags` (viz. kapitola 4.8.7.3)

```
procedure SetFlags( ASet, AReset: Word ); virtual;
```

Parametry:

`AFlags` Kombinace příznaků s prefixem `btf_` (viz. kapitola 4.8.7.1).

Návratové hodnoty:

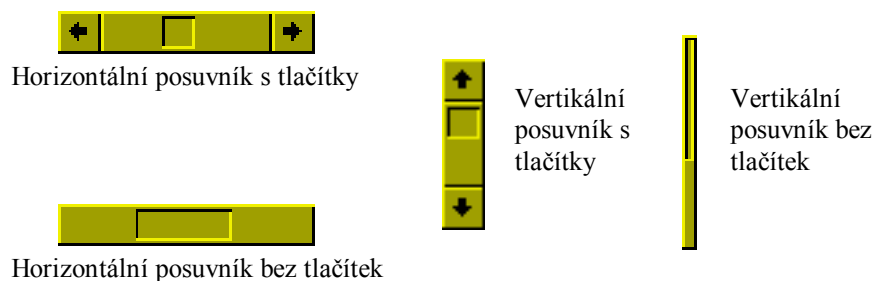
Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

4.9. Komponenta `TScrollBar`

Komponentu **TScrollBar** (posuvník) lze použít k různým účelům. Pomocí posuvníku lze např. zadávat celočíselné hodnoty v omezeném rozsahu, spíše se však uplatní jako doplňková komponenta ke komponentám **TListView**, **TListBox**, **TMenuOverlay** apod. pro zobrazení aktuálně viditelného výřezu seznamu. Na následujících obrázcích jsou zobrazeny ukázky této komponenty.



Komponentu **TScrollBar** je primárně určena pro terminály s dotykovým panelem, lze ji však použít i u terminálu vybavených pouze klávesnicí (především variantu bez tlačítek po stranách).

U komponenty **TScrollBar** lze nastavovat několik parametrů, které ovlivňují její chování, příp. vykreslování. Lze nastavit:

- Meze rozsahu
- Velikost jednoduchého kroku (tj. krok při stisku kláves vkLeft, vkRight resp. vkUp a vkDown).
- Velikost kroku při stránkování (tj. krok při stisku kláves vkPageUp, vkPageDown)
- Posuvník s fixní nebo variabilní velikostí indikátoru pozice
- Zobrazení tlačítek na stranách posuvníku
- Generování oběžníku (broadcast) při změně pozice posuvníku
- Generování oznámení při změně pozice posuvníku

4.9.1. Položky a metody třídy TScrollBar

Třída **TScrollBar** má následující konstruktor:

```
constructor Init( const ABounds: TRect );
```

Popis parametrů konstruktoru je uveden v následující tabulce:

ABounds	Umístění a rozměry komponenty v rámci vlastníka
---------	---

Komponenta **TScrollBar** rozšiřuje bázovou třídu **TControl** o položky uvedené v následující tabulce:

Flags	Příznaky upřesňující chování komponenty. Položka může obsahovat kombinaci následujících příznaků:	
	sbfFixedPage	Příznak určuje, zda má indikátor pozice posuvníku pevnou šířku (příznak je nastaven) nebo je šířka odvozena od položky Min, Max a PageStep (příznak není nastaven). Viz. kapitola 4.9.1.1.
	sbfBroadcast	Komponenta při změně hodnoty generuje oběžník, tj. událost evBroadcast.
	sbfNotify	Komponenta při změně hodnoty generuje oznámení nmChange.
	sbfButtons	Na stranách komponenty jsou vykreslena tlačítka.
Min	Dolní mez rozsahu hodnoty posuvníku. Položka je určena pouze pro čtení. Pro nastavení rozsahu použijte metodu SetRange nebo SetParams .	
Max	Horní mez rozsahu hodnoty posuvníku. Položka je určena pouze pro čtení. Pro nastavení rozsahu použijte metodu SetRange nebo SetParams .	
Position	Aktuální hodnota posuvníku. Položka je určena pouze pro čtení. Pro nastavení aktuální hodnoty použijte metodu SetPosition .	
Step	Krok při stisku klávesy vkLeft, vkRight resp. vkUp, vkDown nebo	

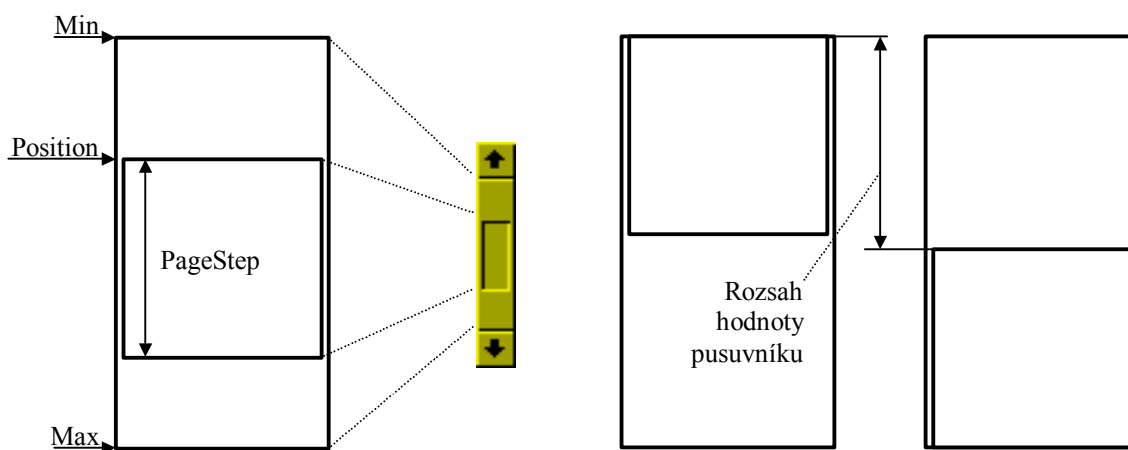
	při kliknutí na tlačítko na jedné ze stran posuvníku. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetParams .
PageStep	Krok při stisku klávesy vkPageUp, vkPageDown nebo při kliknutí do oblasti mezi indikátorem pozice a tlačítkem na jedné ze stran posuvníku. Pokud není nastaven příznak sbfFixedPage, pak tato položka udává zároveň velikost stránky (viz. kapitola 4.9.1.1). Pro její nastavení použijte metodu SetParams .

Komponenta **TScrollBar** rozšiřuje básovou třídu **TControl** o metodu uvedené v následující tabulce:

SetRange	Metoda pro nastavení rozsahu posuvníku
SetPosition	Metoda pro nastavení aktuální pozice posuvníku
SetParams	Metoda pro nastavení rozsahu, aktuální pozice, kroku a velikosti stránky

4.9.1.1. Význam příznaku sbfFixedPage

Komponenta **TScrollBar** může pracovat ve dvou režimech na základě příznaku sbfFixedPage v položce Flags. Pokud je tento příznak nastaven, pak má ukazatel pozice posuvníku (jezdec) fixní velikost (má vždy tvar čtverce) a krajní hodnoty odpovídají nastaveným hodnotám Min a Max. Pokud příznak sbfFixedPage nastaven není pak je šířka indikátoru pozice odvozena od rozsahu posuvníku, tj. položek Min a Max a velikosti nastavené stránky, tj. položky PageStep. Viz. následující obrázek:



Použití posuvníku bez nastaveného příznaku sbfFixedPage se předpokládá ve spojení s další komponentou zobrazující výřez z určitého celku. Šířka posuvníku odpovídá proporcionálně poměru zobrazeného výřezu ku zmíněnému celku. Horní (resp. levé) krajní poloze ukazatele pozice odpovídá hodnota Min. Dolní (resp. pravé) poloze ukazatele odpovídá hodnota Max snižená o velikost stránky + 1. Tj. pozice se pohybuje v rozsahu $\langle \text{Min}, \text{Max} - \text{PageStep} + 1 \rangle$.

4.9.2. Implicitní nastavení komponenty TScrollBar

V následující tabulce jsou uvedena implicitní nastavení všech položek komponenty.

Všechny ostatní položky, které nejsou v tabulce uvedené, jsou inicializovány na hodnotu 0, **nil**, apod.

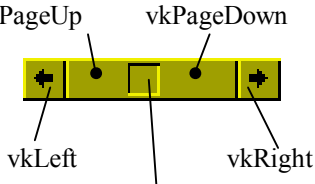
Položka	Hodnota
Bounds	parametr ABounds konstrukturu
EventMask	\$FFFF
CaretSize	(1, 8)
State	sfVisible
Options	ofSelectable or ofSharedPalette
Font	fidDefault
Palette	#\$0F#\$00#\$00#\$00#\$00#\$00 (sdílená paleta)
Flags	sbfBroadcast or sbfButtons
Min	0
Max	100
Step	1
PageStep	10
Position	50

Pro komponentu **TScrollBar** jsou definovány konstanty nastavení třídy uvedené v následující tabulce. Tyto konstanty lze předat jako parametr metodě **Customize** (viz. manuál ke knihovně Controls).

Konstanta	Význam
ccScrollBar	Implicitní nastavení definované ovladačem použitého terminálu.

4.9.3. Události obsluhované komponentou TScrollBar

Komponenta **TScrollBar** obsluhuje vstupní události podle následující tabulky

Událost	Popis činnosti								
evMouseDown evMouseRep evMouseDown	<p>Oblast komponenty je rozdělena na pět částí.</p>  <p>Při kliknutí mimo indikátor posuvníku dojde ke změně hodnoty posuvníku odpovídající uvedenému kódu klávesy na obrázku. Při kliknutí do oblasti indikátoru je možné při držení tlačítka myši plynule měnit jeho pozici.</p>								
evKeyDown	<table border="1"> <tbody> <tr> <td>vkLeft</td> <td>setPosition(Position - Step)</td> </tr> <tr> <td>vkRight</td> <td>setPosition(Position + Step)</td> </tr> <tr> <td>vkUp</td> <td>setPosition(Position - Step)</td> </tr> <tr> <td>vkDown</td> <td>setPosition(Position + Step)</td> </tr> </tbody> </table>	vkLeft	setPosition(Position - Step)	vkRight	setPosition(Position + Step)	vkUp	setPosition(Position - Step)	vkDown	setPosition(Position + Step)
vkLeft	setPosition(Position - Step)								
vkRight	setPosition(Position + Step)								
vkUp	setPosition(Position - Step)								
vkDown	setPosition(Position + Step)								

vkPageUp	SetPosition(Position – PageStep)
vkPageDown	SetPosition(Position + PageStep)
vkHome	SetPosition(Min)
vkEnd	SetPosition(Max)

Pozn.: Pokud je posuvník horizontální, pak reaguje na klávesy vkLeft a vkRight a nereaguje na klávesy vkUp a vkDown. Podobně naopak, pokud je posuvník vertikální reaguje na klávesy vkUp a vkDown a nereaguje na klávesy vkLeft a vkRight.

4.9.4. Oznámení generovaná komponentou TScrollBar

Komponenta generuje následující oznámení:

Oznámení	Okamžik vyvolání	Parametry oznámení
nmEnter	Komponenta se dostane do ohniska	-
nmExit	Komponenta se dostane mimo ohnisko	-
nmHide	Volání metody Hide	-
nmShow	Volání metody Show	-
nmEnable	Volání metody Enable	-
nmDisable	Volání metody Disable	-
nmCanExit	Před opuštěním ohniska, pokud je nastaven příznak ofValidate v položce Options	Položka Accept struktury oznámení obsahuje hodnotu False. Opuštění ohniska lze zakázat nastavením této položky na hodnotu False.
nmChange	Změna hodnoty komponenty a zároveň je nastaven příznak sbfNotify v položce Flags	-

Komponenta dále může generovat událost evBroadcast s příkazem **cmScrollBarChanged** při změně aktuální hodnoty. Tato událost se generuje v případě, že je nastaven příznak sbfBroadcast v položce Flags.

4.9.5. Validace hodnoty komponenty TScrollBar

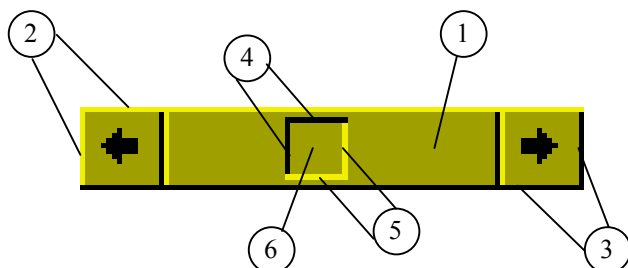
S komponentu **TScrollBar** lze použít všechny typy validátorů implementujících metodu **TransferOrdinal**, tj. všechny potomky třídy **TOrdinalValidator**.

4.9.6. Paleta komponenty TScrollBar

Konstruktor komponenty **TScrollBar** nastavuje implicitně paletu barev komponenty podle následující tabulky:

#	Barva	Popis
---	-------	-------

1	\$0F	Pozadí
2	■ \$00	Linka rámečku komponenty (vlevo a nahoře)
3	■ \$00	Linka rámečku komponenty (vpravo a dole)
4	■ \$00	Linka rámečku indikátoru (vlevo a nahoře)
5	■ \$00	Linka rámečku indikátoru (vpravo a dole)
6	■ \$00	Pozadí indikátoru



4.9.7. Reference

Třída **TScrollBar** je potomkem třídy **TCustomScrollBar** popsané v dokumentaci ke knihovně Controls.

```

PScrollBar = ^TScrollBar;
TScrollBar = object( TCustomScrollBar )
public
  constructor Init( const ABounds: TRect );
  procedure Paint( ACanvas: PCanvas ); virtual;
  procedure PaintIndicator; virtual;
  procedure GetIndicatorCoords( var ARect: TRect;
                               var AIndPos, AIndSize: Integer ); virtual;
end;

```

4.9.7.1. Konstruktor TScrollBar.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```

constructor Init( const ABounds: TRect );

```

Parametry:

ABounds Umístění a rozměry komponenty v rámci jejího vlastníka.

Návratové hodnoty:

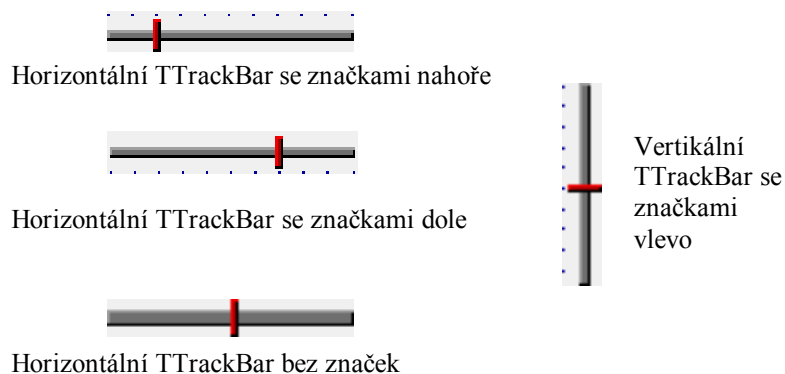
Konstruktor nevrací žádnou hodnotu.

Poznámky:

Konstruktor **Init** nastaví všechny položky instance na implicitní hodnoty, viz. tabulka v kapitole 4.9.2.

4.10. Komponenta TTrackBar

Komponenta **TTrackBar** je určena k zadávání celočíselných hodnot v omezeném rozsahu. Na následujících obrázcích jsou zobrazeny ukázky této komponenty.



Komponenta **TTrackBar** je primárně určena pro terminály s dotykovým panelem, lze ji však použít i u terminálu vybavených pouze klávesnicí.

U komponenty **TTrackBar** lze nastavovat několik parametrů, které ovlivňují její chování, příp. vykreslování. Lze nastavit:

- Meze rozsahu
- Velikost jednoduchého kroku (tj. krok při stisku kláves vkLeft, vkRight resp. vkUp a vkDown).
- Velikost kroku při stránkování (tj. krok při stisku kláves vkPageUp, vkPageDown)
- Zobrazení značek s nastavitelnou frekvencí
- Pozici značek (vlevo/vpravo, nahoře/dole)
- Generování oběžníku (broadcast) při změně pozice posuvníku
- Generování oznámení při změně pozice posuvníku

4.10.1. Položky a metody třídy TTrackBar

Třída **TTrackBar** má následující konstruktor:

```
constructor Init( const ABounds: TRect );
```

Popis parametrů konstruktoru je uveden v následující tabulce:

ABounds	Umístění a rozměry komponenty v rámci vlastníka
----------------	---

Komponenta **TTrackBar** rozšiřuje básovou třídu **TControl** o položky uvedené v následující tabulce:

Flags	Příznaky upřesňující chování komponenty. Položka obsahuje kombinaci následujících příznaků:
--------------	---

	trfTickMarks	Po stranách komponenty jsou zobrazeny značky v odstupech daných položkou Frequency
	trfTopLeft	Značky umístěné vlevo (příp. nahoře). Pokud příznak není nastaven pak vpravo (příp. dole)
	trfDragging	zatím neimplementováno
	Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetFlags .	
Min	Dolní mez rozsahu hodnoty. Položka je určena pouze pro čtení. Pro nastavení rozsahu použijte metodu SetRange nebo SetParams .	
Max	Horní mez rozsahu hodnoty. Položka je určena pouze pro čtení. Pro nastavení rozsahu použijte metodu SetRange nebo SetParams .	
Step	Krok při stisku klávesy vkLeft, vkRight resp. vkUp, vkDown nebo při kliknutí na tlačítko na jedné ze stran posuvníku. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetParams .	
PageStep	Krok při stisku klávesy vkPageUp, vkPageDown nebo při kliknutí do oblasti mezi indikátorem pozice a tlačítkem na jedné ze stran posuvníku. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetParams	
Frequency	Frekvence zobrazených značek. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetFrequency .	
Position	Aktuální hodnota (pozice ukazatele). Položka je určena pouze pro čtení. Pro nastavení aktuální hodnoty použijte metodu SetPosition .	

Komponenta **TTrackBar** rozšiřuje básovou třídu **TControl** o metodu uvedené v následující tabulce:

SetRange	Metoda pro nastavení rozsahu posuvníku
SetPosition	Metoda pro nastavení aktuální pozice posuvníku
SetParams	Metoda pro nastavení rozsahu, aktuální pozice a kroku
SetFrequency	Metoda pro nastavení položky Frequency.
SetFlags	Metoda pro nastavení položky Flags.

4.10.2. Implicitní nastavení komponenty TTrackBar

V následující tabulce jsou uvedena implicitní nastavení všech položek komponenty. Všechny ostatní položky, které nejsou v tabulce uvedené, jsou inicializovány na hodnotu 0, nil, apod.

Položka	Hodnota
Bounds	parametr ABounds konstrukturu
EventMask	\$FFFF
CaretSize	(1, 8)
State	sfVisible
Options	ofSelectable or ofSharedPalette or ofFirstClick
Font	fidDefault
Palette	#\$OF#\$00#\$00#\$0F#\$0F#\$0F#\$00#\$00#\$00#\$00#\$00#\$00 (sdílená paleta)

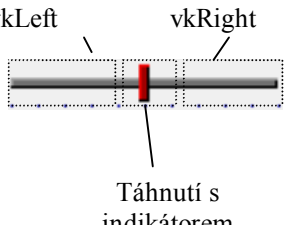
Flags	trfTickMarks
Min	0
Max	10
Step	1
PageStep	1
Position	0
Frequency	1

Pro komponentu **TTrackBar** jsou definovány konstanty nastavení třídy uvedené v následující tabulce. Tyto konstanty lze předat jako parametr metodě **Customize** (viz. manuál ke knihovně Controls).

Konstanta	Význam
ccTrackBar	Implicitní nastavení definované ovladačem použitého terminálu.

4.10.3. Události obsluhované komponentou TTrackBar

Komponenta **TTrackBar** obsluhuje vstupní události podle následující tabulky

Událost	Popis činnosti																
evMouseDown evMouseRep evMouseDown	<p>Oblast komponenty je rozdělena na pět částí.</p>  <p>Při kliknutí mimo indikátor posuvníku dojde ke změně hodnoty posuvníku odpovídající uvedenému kódu klávesy na obrázku. Při kliknutí do oblasti indikátoru je možné při držení tlačítka myši plynule měnit jeho pozici.</p>																
evKeyDown	<table border="1"> <tr> <td>vkLeft</td> <td>SetPosition(Position – Step)</td> </tr> <tr> <td>vkRight</td> <td>SetPosition(Position + Step)</td> </tr> <tr> <td>vkUp</td> <td>SetPosition(Position – Step)</td> </tr> <tr> <td>vkDown</td> <td>SetPosition(Position + Step)</td> </tr> <tr> <td>vkPageUp</td> <td>SetPosition(Position – PageStep)</td> </tr> <tr> <td>vkPageDown</td> <td>SetPosition(Position + PageStep)</td> </tr> <tr> <td>vkHome</td> <td>SetPosition(Min)</td> </tr> <tr> <td>vkEnd</td> <td>SetPosition(Max)</td> </tr> </table>	vkLeft	SetPosition(Position – Step)	vkRight	SetPosition(Position + Step)	vkUp	SetPosition(Position – Step)	vkDown	SetPosition(Position + Step)	vkPageUp	SetPosition(Position – PageStep)	vkPageDown	SetPosition(Position + PageStep)	vkHome	SetPosition(Min)	vkEnd	SetPosition(Max)
vkLeft	SetPosition(Position – Step)																
vkRight	SetPosition(Position + Step)																
vkUp	SetPosition(Position – Step)																
vkDown	SetPosition(Position + Step)																
vkPageUp	SetPosition(Position – PageStep)																
vkPageDown	SetPosition(Position + PageStep)																
vkHome	SetPosition(Min)																
vkEnd	SetPosition(Max)																

Pozn.: Pokud je komponenta horizontální, pak reaguje na klávesy vkLeft a vkRight a nereaguje na klávesy vkUp a vkDown. Podobně naopak, pokud je komponenta vertikální reaguje na klávesy vkUp a vkDown a nereaguje na klávesy vkLeft a vkRight.

4.10.4. Oznámení generovaná komponentou TTrackBar

Komponenta generuje následující oznámení:

Oznámení	Okamžik vyvolání	Parametry oznámení
nmEnter	Komponenta se dostane do ohniska	-
nmExit	Komponenta se dostane mimo ohnisko	-
nmHide	Volání metody Hide	-
nmShow	Volání metody Show	-
nmEnable	Volání metody Enable	-
nmDisable	Volání metody Disable	-
nmCanExit	Před opuštěním ohniska, pokud je nastaven příznak ofValidate v položce Options	Položka Accept struktury oznámení obsahuje hodnotu False. Opuštění ohniska lze zakázat nastavením této položky na hodnotu False.
nmChange	Změna hodnoty komponenty	

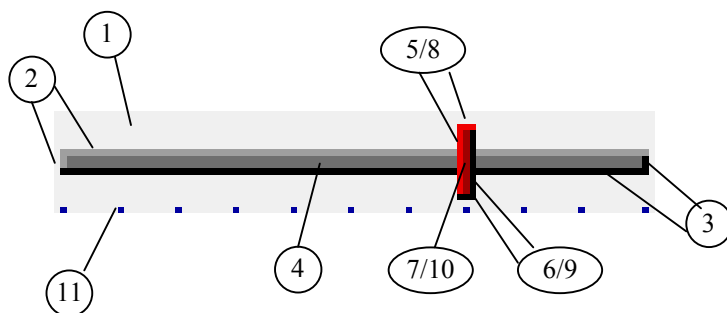
4.10.5. Validace hodnoty komponenty TrackBar

S komponentu **TTrackBar** lze použít všechny typy validátorů implementujících metodu **TransferOrdinal**, tj. všechny potomky třídy **TOrdinalValidator**.

4.10.6. Paleta komponenty TTrackBar

Konstruktor komponenty **TTrackBar** nastavuje implicitně paletu barev komponenty podle následující tabulky:

#	Barva	Popis
1	\$0F	Pozadí
2	■ \$00	Linka rámečku lišty indikátoru vlevo a nahoře
3	■ \$00	Linka rámečku lišty indikátoru vpravo a dole
4	\$0F	Pozadí lišty indikátoru
5	\$0F	Linka rámečku indikátoru vlevo a nahoře (komponenta mimo ohnisko)
6	\$0F	Linka rámečku indikátoru vpravo a dole (komponenta mimo ohnisko)
7	■ \$00	Pozadí indikátoru (komponenta mimo ohnisko)
8	■ \$00	Linka rámečku indikátoru vlevo a nahoře (komponenta v ohnisku)
9	■ \$00	Linka rámečku indikátoru vpravo a dole (komponenta v ohnisku)
10	■ \$00	Pozadí indikátoru (komponenta v ohnisku)
11	■ \$00	Značky
12	■ \$00	Obdélník pro zvýraznění komponenty v ohnisku



4.10.7. Reference

Třída **TTrackBar** je potomkem třídy **TControl** popsané v dokumentaci ke knihovně Controls.

```

PTrackBar = ^TTrackBar;
TTrackBar = object( TControl )
public
  Min      : Integer;
  Max      : Integer;
  Step     : Integer;
  PageStep : Integer;
  Frequency : Integer;
  Position : Integer;
  Flags    : Word;

  constructor Init( const ABounds: TRect );
  procedure Paint( ACanvas: PCanvas ); virtual;
  procedure HandleEvent( var AEvent: TEvent ); virtual;
  procedure SetState( AState: Word; AEnable: Boolean ); virtual;
  function Transfer( AMode: Integer ): Integer; virtual;
  procedure SetRange( AMin, AMax: Integer );
  procedure SetPosition( APosition: Integer );
  procedure SetParams( AMin, AMax, AStep, APageStep,
    APosition: Integer );
  procedure SetFlags( ASet, AReset: Word ); virtual;
  procedure SetFrequency( AFrequency: Integer );
  procedure DoChange; virtual;
end;

```

4.10.7.1. Konstanty trf_

Konstanty s prefixem `trf_` upřesňují chování komponenty **TTrackBar**. Kombinace konstant je uložena v položce `Flags` této komponenty (viz. kapitola 4.10.7.8).

Identifikátor	Kód	Popis
<code>trfTickMarks</code>	<code>\$0001</code>	Po stranách komponenty jsou zobrazeny značky v odstupech daných položkou <code>Frequency</code> (viz. kapitola 4.10.7.6)
<code>trfTopLeft</code>	<code>\$0002</code>	Značky jsou umístěny vlevo (příp. nahoře). Pokud příznak není nastaven pak vpravo (příp. dole)
<code>trfDragging</code>	<code>\$0004</code>	Zatím neimplementováno

4.10.7.2. Položka TTrackBar.Min

Položka **Min** obsahuje dolní mez rozsahu komponenty. Položka je určena pouze pro čtení. Pro její nastavení slouží metody **SetRange** (viz. kapitola 4.10.7.10) příp.

SetParams (viz. kapitola 4.10.7.12).

Min : Integer;

4.10.7.3. Položka TTrackBar.Max

Položka **Max** obsahuje horní mez rozsahu komponenty. Položka je určena pouze pro čtení. Pro její nastavení slouží metody **SetRange** (viz. kapitola 4.10.7.10) příp. **SetParams** (viz. kapitola 4.10.7.12).

Min : Integer;

4.10.7.4. Položka TTrackBar.Step

Položka **Step** obsahuje krok změny hodnoty komponenty při stisku kláves vkLeft, vkRight příp. vkUp a vkDown. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu **SetParams** (viz. kapitola 4.10.7.12).

Step : Integer;

4.10.7.5. Položka TTrackBar.PageStep

Položka **PageStep** obsahuje krok změny hodnoty komponenty při stisku kláves vkPageUp a vkPageDown. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu **SetParams** (viz. kapitola 4.10.7.12).

PageStep : Integer;

4.10.7.6. Položka TTrackBar.Frequency

Položka **Frequency** určuje odstup značek zobrazených po stranách komponenty v případě nastaveného příznaku trfTickMarks v položce **Flags**. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu **SetFrequency** (viz. kapitola 4.10.7.8)

Frequency : Integer;

4.10.7.7. Položka TTrackBar.Position

Položka **Position** obsahuje aktuální hodnotu komponenty. Hodnota položky se pohybuje v rozsahu daném položkami Min a Max. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetPosition** (viz. kapitola 4.10.7.11).

Position : Integer;

4.10.7.8. Položka TTrackBar.Flags

Položka **Flags** upřesňuje chování komponenty. Obsahuje kombinaci příznaků s prefixem trf_ popsaných v kapitole 4.10.7.1. Položka je určena pouze pro čtení. Pro její nastavení slouží metodu **SetFlags** (viz. kapitola 4.10.7.13).

Flags : Word;

4.10.7.9. Konstruktor TTrackBar.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```
constructor Init( const ABounds: TRect );
```

Parametry:

ABounds Umístění a rozměry komponenty v rámci jejího vlastníka.

Návratové hodnoty:

Konstruktor nevrací žádnou hodnotu.

Poznámky:

Konstruktor **Init** nastaví všechny položky instance na implicitní hodnoty, viz. tabulka v kapitole 4.10.2.

4.10.7.10. Metoda TTrackBar.SetRange

Metoda **SetRange** slouží k nastavení rozsahu komponenty, tj. nastavení položek Min a Max.

```
procedure SetRange( AMin, AMax: Integer );
```

Parametry:

AMin Dolní mez rozsahu komponenty
AMax Horní mez rozsahu komponenty

Návratové hodnoty:

Metoda nevrací žádnou hodnotu

Poznámky:

Hodnota dolní meze rozsahu komponenty musí menší než hodnota horní meze rozsahu. Pokud je položka Position mimo nově zadaný rozsah, pak je její hodnota upravena tak, aby odpovídala jedné z krajních hodnot rozsahu. V případě potřeby je komponenta překreslena metodou **Repaint**.

4.10.7.11. Metoda TTrackBar.SetPosition

Metoda **SetPosition** slouží k nastavení aktuální hodnoty komponenty, tj. položky Position.

```
procedure SetPosition( APosition: Integer );
```

Parametry:

APosition Požadovaná aktuální hodnota komponenty.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu

Poznámky:

Pokud se parametr APosition liší od aktuální hodnoty, pak je komponenta překreslena a je zavolána metoda **DoChange** (viz. kapitola 4.10.7.15)

4.10.7.12. Metoda TTrackBar.SetParams

Metoda **SetParams** slouží k nastavení základních parametrů komponenty, tj. položek Min, Max, Step, PageStep a Position..

```
procedure SetParams( AMin, AMax, AStep, APageStep,  
    APosition: Integer );
```

Parametry:

AMin	Dolní mez rozsahu komponenty
AMax	Horní mez rozsahu komponenty
AStep	Krok při stisku kláves vkLeft a vkRight příp. vkUp a vkDown
APageStep	Krok při stisku kláves vkPageUp a vkPageDown
APosition	Požadovaná aktuální hodnota komponenty

Návratové hodnoty:

Metoda nevrací žádnou hodnotu

Poznámky:

Pokud se parametr APosition liší od aktuální hodnoty, pak je komponenta překreslena a je zavolána metoda **DoChange** (viz. kapitola 4.10.7.15). V případě potřeby je komponenta překreslena.

4.10.7.13. Metoda TTrackBar.SetFlags

Metoda **SetFlags** slouží k nastavení položky Flags (viz. kapitola 4.10.7.8)

```
procedure SetFlags( ASet, AReset: Word ); virtual;
```

Parametry:

AFlags	Kombinace příznaků s prefixem trf_ (viz. kapitola 4.10.7.1).
--------	--

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

4.10.7.14. Metoda TTrackBar.SetFrequency

Metoda **SetFrequency** slouží k nastavení odstupe značek zobrazovaných po stranách komponenty, tj. položky Frequency (viz. kapitola 4.10.7.6)

```
procedure SetFrequency( AFrequency: Integer );
```

Parametry:

AFrequency Požadovaný odstup značek.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

4.10.7.15. Metoda TTrackBar.DoChange

Metoda **DoChange** je automaticky zavolána vždy, když dojde ke změně hodnoty komponenty, tj. ke změně položky Position.

```
procedure DoChange; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Metoda **DoChange** implicitně vyvolá oznámení nmChange. Potomci třídy **TTrackBar** mohou tuto metodu předefinovat.

4.11. Komponenta TUpDown

Komponenta **TUpDown** je jednoduchá komponenta určená ke změně celočíselné proměnné v nastavitelném rozsahu o definovaný krok nahoru nebo dolů. Na následujících obrázcích jsou zobrazeny ukázky této komponenty.

Vertikální
UpDownHorizontální
UpDown

Komponenta **TUpDown** je primárně určena pro terminály s dotykovým panelem, lze ji však použít i u terminálu vybavených pouze klávesnicí.

U komponenty **TUpDown** lze nastavovat několik parametrů, které ovlivňují její chování, příp. vykreslování. Lze nastavit:

- Meze rozsahu
- Velikost kroku při stisku tlačítka
- Zobrazení rámečku kolem komponenty

4.11.1. Položky a metody třídy TUpDown

Třída **TUpDown** má následující konstruktor:

```
constructor Init( const ABounds: TRect );
```

Popis parametrů konstruktoru je uveden v následující tabulce:

ABounds	Umístění a rozměry komponenty v rámci vlastníka
---------	---

Komponenta **TUpDown** rozšiřuje báзовou třídu **TControl** o položky uvedené v následující tabulce:

Flags	Příznaky upřesňující chování komponenty. Hodnota položky je kombinací následujících příznaků: <table border="1" data-bbox="454 1377 1348 1422"> <tr> <td>udfBorder</td> <td>Zobrazení rámečku kolem komponenty</td> </tr> </table> Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetFlags .	udfBorder	Zobrazení rámečku kolem komponenty
udfBorder	Zobrazení rámečku kolem komponenty		
Min	Dolní mez rozsahu hodnoty komponenty. Položka je určena pouze pro čtení. Pro nastavení mezí použijte metodu SetRange nebo SetParams .		
Max	Horní mez rozsahu hodnoty komponenty. Položka je určena pouze pro čtení. Pro nastavení mezí použijte metodu SetRange nebo SetParams .		
Step	Krok změny hodnoty komponenty při stisku klávesy vkLeft, vkRight příp. vkUp, vkDown nebo při kliknutí do oblasti komponenty. Položka je určena pouze ke čtení. Pro její nastavení použijte metodu SetParams .		
Position	Aktuální hodnota komponenty. Položka je určena pouze pro čtení. Pro nastavení aktuální hodnoty použijte metodu SetPosition nebo SetParams .		
Link	Odkaz na komponentu, které bude při každé změně hodnoty		

	komponenty odeslána zpráva <code>cmSetText</code> obsahující hodnotu komponenty převedenou na znakový řetězec. Položka může obsahovat odkaz na komponentu typu <code>TStaticText</code> nebo <code>TEdit</code> . Položka je určena pouze pro čtení. Pro její nastavení použijte metodu <code>SetLink</code> .
--	--

Komponenta **TUpDown** rozšiřuje básovou třídu **TControl** o metodu uvedené v následující tabulce:

SetPosition	Metoda pro nastavení aktuální hodnoty komponenty
SetRange	Metoda pro nastavení rozsahu hodnoty komponenty
SetParams	Metoda pro nastavení položek Min, Max, Step a Position.
SetFlags	Metoda pro nastavení položky Flags
SetLink	Metoda pro nastavení položky Link

4.11.2. Implicitní nastavení komponenty TUpDown

V následující tabulce jsou uvedena implicitní nastavení všech položek komponenty. Všechny ostatní položky, které nejsou v tabulce uvedené, jsou inicializovány na hodnotu 0, `nil`, apod.

Položka	Hodnota
Bounds	parametr <code>ABounds</code> konstrukturu
EventMask	<code>evKeyDown + evMouseDown + evMouseDown + evMouseDown + evMouseDown</code>
CaretSize	(1, 8)
State	<code>sfVisible</code>
Options	<code>ofSelectable or ofSharedPalette or ofFirstClick</code>
Font	<code>fidDefault</code>
Palette	<code>#\$00</code> (sdílená paleta)
Flags	<code>udfBorder</code>
Min	0
Max	100
Step	1
Position	0

Pro komponentu **TUpDown** jsou definovány konstanty nastavení třídy uvedené v následující tabulce. Tyto konstanty lze předat jako parametr metodě `Customize` (viz. manuál ke knihovně `Controls`).

Konstanta	Význam
<code>ccUpDown</code>	Implicitní nastavení definované ovladačem použitého terminálu.

4.11.3. Události obsluhované komponentou TUpDown

Komponenta **TUpDown** obsluhuje vstupní události podle následující tabulky

Událost		Popis činnosti
evMouseDown evMouseRep evMouseDown		Kliknutím do oblasti komponenty způsobí změnu hodnoty položky Position buď kladným nebo záporným směrem o hodnotu položky Step.
evKeyDown	vkLeft	SetPosition(Position – Step)
	vkRight	SetPosition(Position + Step)
	vkUp	SetPosition(Position – Step)
	vkDown	SetPosition(Position + Step)
	vkHome	SetPosition(Min)
	vkEnd	SetPosition(Max)

Pozn.: Pokud je komponenta horizontální, pak reaguje na klávesy vkLeft a vkRight a nereaguje na klávesy vkUp a vkDown. Podobně naopak, pokud je komponenta vertikální reaguje na klávesy vkUp a vkDown a nereaguje na klávesy vkLeft a vkRight.

4.11.4. Oznámení generovaná komponentou TUpDown

Komponenta generuje následující oznámení:

Oznámení	Okamžik vyvolání	Parametry oznámení
nmEnter	Komponenta se dostane do ohniska	-
nmExit	Komponenta se dostane mimo ohnisko	-
nmHide	Volání metody Hide	-
nmShow	Volání metody Show	-
nmEnable	Volání metody Enable	-
nmDisable	Volání metody Disable	-
nmCanExit	Před opuštěním ohniska, pokud je nastaven příznak ofValidate v položce Options	Položka Accept struktury oznámení obsahuje hodnotu False. Opuštění ohniska lze zakázat nastavením této položky na hodnotu False.
nmChange	Změna hodnoty komponenty	

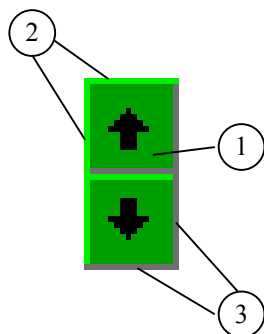
4.11.5. Validace hodnoty komponenty TUpDown

S komponentu **TUpDown** lze použít všechny typy validátorů implementujících metodu **TransferOrdinal**, tj. všechny potomky třídy **TOrdinalValidator**.

4.11.6. Paleta komponenty TUpDown

Konstruktor komponenty **TUpDown** nastavuje implicitně paletu barev komponenty podle následující tabulky:

#	Barva	Popis
1	\$0F	Pozadí
2	■ \$00	Linka rámečku (vlevo a nahoře)
3	■ \$00	Linka rámečku (vpravo a dole)
4	■ \$00	Obdélník pro zvýraznění komponenty v ohnisku



4.11.7. Reference

Třída **TUpDown** je potomkem třídy **TControl** popsané v dokumentaci ke knihovně Controls.

```

PUpDown = ^TUpDown;
TUpDown = object( TControl )
public
  Flags      : Word;
  Position   : Integer;
  Min        : Integer;
  Max        : Integer;
  Step       : Integer;
  Link       : PControl;

  constructor Init( const ABounds: TRect );
  procedure Paint( ACanvas: PCanvas ); virtual;
  procedure HandleEvent( var AEvent: TEvent ); virtual;
  procedure SetState( AState: Word; AEnable: Boolean ); virtual;
  function Transfer( AMode: Integer ): Integer; virtual;
  procedure DoChange; virtual;
  procedure SetRange( AMin, AMax: Integer );
  procedure SetPosition( APosition: Integer );
  procedure SetParams( AMin, AMax, AStep, APosition: Integer );
  procedure SetFlags( ASet, AReset: Word ); virtual;
  procedure SetLink( ALink: PControl );
end;

```

4.11.7.1. Konstanty udf_

Konstanty s prefixem `udf_` upřesňují chování komponenty **TUpDown**. Kombinace konstant je uložena v položce `Flags` této komponenty (viz. kapitola 4.11.7.2).

Identifikátor	Kód	Popis
<code>udfBorder</code>	<code>\$0001</code>	Zobrazení rámečku kolem komponenty

4.11.7.2. Položka TUpDown.Flags

Položka **Flags** upřesňuje chování komponenty. Obsahuje kombinaci příznaků s prefixem `udf_` popsáných v kapitole 4.11.7.1. Položka je určena pouze pro čtení. Pro její nastavení slouží metodu **SetFlags** (viz. kapitola 4.11.7.13).

```
Flags      : Word;
```

4.11.7.3. Položka TUpDown.Position

Položka **Position** obsahuje aktuální hodnotu komponenty. Hodnota položky se pohybuje v rozsahu daném položkami `Min` a `Max`. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetPosition** (viz. kapitola 4.11.7.11).

```
Position   : Integer;
```

4.11.7.4. Položka TUpDown.Min

Položka **Min** obsahuje dolní mez rozsahu komponenty. Položka je určena pouze pro čtení. Pro její nastavení slouží metody **SetRange** (viz. kapitola 4.11.7.10) příp. **SetParams** (viz. kapitola 4.11.7.12).

```
Min        : Integer;
```

4.11.7.5. Položka TUpDown.Max

Položka **Max** obsahuje horní mez rozsahu komponenty. Položka je určena pouze pro čtení. Pro její nastavení slouží metody **SetRange** (viz. kapitola 4.11.7.10) příp. **SetParams** (viz. kapitola 4.11.7.12).

```
Max        : Integer;
```

4.11.7.6. Položka TUpDown.Step

Položka **Step** obsahuje krok změny hodnoty komponenty při stisku kláves `vkLeft`, `vkRight` příp. `vkUp` a `vkDown` nebo při klinutí na jedno z tlačítek v oblasti komponenty. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu **SetParams** (viz. kapitola 4.11.7.12).

```
Step       : Integer;
```

4.11.7.7. Položka TUpDown.Link

Položka **Link** obsahuje odkaz na komponentu, které bude při změně hodnoty komponenty poslána pomocí funkce **Message** zpráva `cmSetText`. Cílem této zprávy je nastavení textu komponenty podle aktuální hodnoty komponenty **TUpDown**. Komponentou odkazovanou parametrem `Link` může být např. `TStaticText` nebo `TEdit`. Parametr zprávy `cmSetText` je ukazatel na znakový řetězec. Tento znakový řetězec je buď decimální reprezentace hodnoty komponenty a nebo text vrácený metodou **EnumLookup** validátoru přiřazeného komponentě **TUpDown**. Položka `Link` může mít hodnotu `nil`. Položka je určena pouze pro čtení, pro její nastavení slouží metoda **SetLink** (viz. kapitola 4.11.7.14)

Link : PControl;

4.11.7.8. Konstruktor TUpDown.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```
constructor Init( const ABounds: TRect );
```

Parametry:

ABounds Umístění a rozměry komponenty v rámci jejího vlastníka.

Návratové hodnoty:

Konstruktor nevrací žádnou hodnotu.

Poznámky:

Konstruktor **Init** nastaví všechny položky instance na implicitní hodnoty, viz. tabulka v kapitole 4.11.2.

4.11.7.9. Metoda TUpDown.DoChange

Metoda **DoChange** je automaticky zavolána vždy, když dojde ke změně hodnoty komponenty, tj. ke změně položky Position.

```
procedure DoChange; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Metoda **DoChange** implicitně vyvolá oznámení nmChange a odešle zprávu komponentě cmSetText, na kterou se odkazuje položka Link (viz. kapitola 4.11.7.7). Potomci třídy **TUpDown** mohou tuto metodu předefinovat.

4.11.7.10. Metoda TUpDown.SetRange

Metoda **SetRange** slouží k nastavení rozsahu komponenty, tj. nastavení položek Min a Max.

```
procedure SetRange( AMin, AMax: Integer );
```

Parametry:

AMin	Dolní mez rozsahu komponenty
AMax	Horní mez rozsahu komponenty

Návratové hodnoty:

Metoda nevrací žádnou hodnotu

Poznámky:

Hodnota dolní meze rozsahu komponenty musí menší než hodnota horní meze rozsahu. Pokud je položka Position mimo nově zadaný rozsah, pak je její hodnota upravena tak, aby odpovídala jedné z krajních hodnot rozsahu. V případě potřeby je komponenta překreslena metodou **Repaint**.

4.11.7.11. Metoda TUpDown.SetPosition

Metoda **SetPosition** slouží k nastavení aktuální hodnoty komponenty, tj. položky Position.

```
procedure SetPosition( APosition: Integer );
```

Parametry:

APosition Požadovaná aktuální hodnota komponenty.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu

Poznámky:

Pokud se parametr APosition liší od aktuální hodnoty, pak je komponenta překreslena a je zavolána metoda **DoChange** (viz. kapitola 4.11.7.9)

4.11.7.12. Metoda TUpDown.SetParams

Metoda **SetParams** slouží k nastavení základních parametrů komponenty, tj. položek Min, Max, Step a Position..

```
procedure SetParams( AMin, AMax, AStep, APosition: Integer );
```

Parametry:

AMin	Dolní mez rozsahu komponenty
AMax	Horní mez rozsahu komponenty
AStep	Krok při stisku kláves vkLeft a vkRight příp. vkUp a vkDown
APosition	Požadovaná aktuální hodnota komponenty

Návratové hodnoty:

Metoda nevrací žádnou hodnotu

Poznámky:

Pokud se parametr APosition liší od aktuální hodnoty, pak je komponenta překreslena a je zavolána metoda **DoChange** (viz. kapitola 4.11.7.9). V případě potřeby je komponenta překreslena.

4.11.7.13. Metoda TUpDown.SetFlags

Metoda **SetFlags** slouží k nastavení položky Flags (viz. kapitola 4.11.7.2)

```
procedure SetFlags( ASet, AReset: Word ); virtual;
```

Parametry:

AFlags Kombinace příznaků s prefixem trf_ (viz. kapitola 4.11.7.1).

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

4.11.7.14. Metoda TUpDown.SetLink

Metoda **SetLink** slouží k nastavení položky Link (viz. kapitola 4.11.7.7)

```
procedure SetLink( ALink: PControl );
```

Parametry:

ALink Odkaz na inicializovanou komponentu obsluhující zprávu evMessage s kódem cmSetText. Obvykle se jedná o komponentu TStaticText nebo TEdit.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:**4.12. Komponenta TListBox**

Komponenta **TListBox** je určena k zobrazení jednoduchého seznamu hodnot a výběru jedné ze zobrazených položek. Na následujících obrázcích jsou zobrazeny ukázky této komponenty.

Bush, Barbara
Carter, Rosalynn
Cleveland, Frances Folsom
Clinton, Hillary Rodham
Coolidge, Grace Goodhue

ListBox s jedním sloupcem

Alabama	California
Alaska	Colorado
Alberta	Connecticut
Arizona	Florida
Arkansas	Georgia

ListBox s více sloupci (2)

U komponenty **TListBox** lze nastavovat několik parametrů, které ovlivňují její chování, příp. vykreslování. Lze nastavit:

- Počet vykreslovaných sloupců
- Zobrazení rámečku kolem komponenty
- Zobrazení svislých a vodorovných oddělovacích čar
- Zarovnání textu vlevo, vpravo a na střed
- Zarovnání textu nahoru, dolů a na střed
- Výšku položky
- Odsazení textu položky zleva, zprava, shora a zdola

4.12.1. Položky a metody třídy TListBox

Třída **TListBox** má následující konstruktor:

```
constructor Init( const ABounds: TRect; AScrollBar: PScrollBar );
```

Popis parametrů konstruktoru je uveden v následující tabulce:

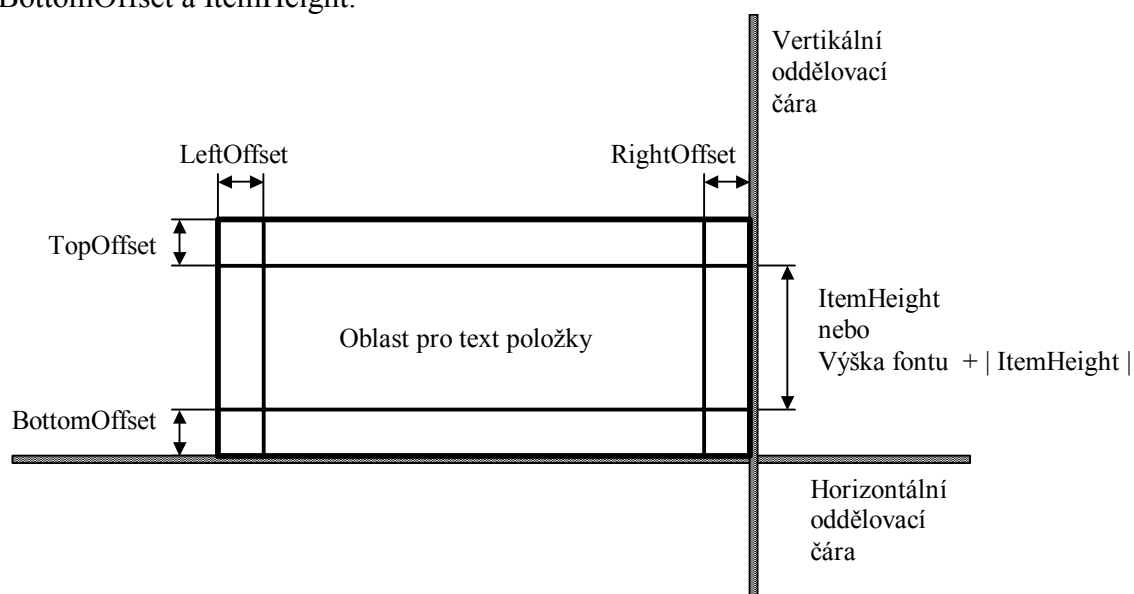
ABounds	Umístění a rozměry komponenty v rámci vlastníka
AScrollBar	Přidružená komponenta TScrollBar zobrazující aktuálně viditelný výřez ze seznamu položek.

Komponenta **TListBox** rozšiřuje bázovou třídu **TControl** o položky uvedené v následující tabulce:

Flags	Příznaky upřesňující chování komponenty. Položka obsahuje kombinaci následujících příznaků:	
	lbfBorder	Zobrazení rámečku kolem komponenty
	lbfLeft	Zarovnání textu položek k levému okraji
	lbfRight	Zarovnání textu položek k pravému okraji
	lbfCenter	Zarovnání textu položek na střed
	lbfVertDiv	Zobrazení vertikálních oddělovacích čar
	lbfHorzDiv	Zobrazení horizontálních oddělovacích čar
	lbfUseEnter	Obsluha klávesy vkEnter, která generuje oznámení nmSelect.
	lbfLookup	Komponenta zobrazuje text jednotlivých variant výčtového typu daného připojeným validátorem.

	Příznaky lbfLeft, lbfRight a lbfCenter nelze navzájem kombinovat. Položka Flags je určena pouze pro čtení. Pro její nastavení použijte metodu SetFlags .
TopOffset	Posunutí oblasti pro zobrazení textu položky od jejího horního okraje. Viz. obrázek níže. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetOffsets .
LeftOffset	Posunutí oblasti pro zobrazení textu položky od jejího levého okraje. Viz. obrázek níže. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetOffsets .
RightOffset	Posunutí oblasti pro zobrazení textu položky od jejího pravého okraje. Viz. obrázek níže. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetOffsets .
BottomOffset	Posunutí oblasti pro zobrazení textu položky od jejího spodního okraje. Viz. obrázek níže. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetOffsets .
ItemHeight	Výška oblasti pro zobrazení textu položky. Pokud je hodnota této položky menší nebo rovna nule, pak je výška této oblasti dána výškou znaků použitého fontu zvětšená o absolutní hodnotu této položky. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetItemHeight .
ScrollBar	Odkaz na přidruženou komponentu ScrollBar. Položka je inicializovaná parametrem konstruktora a je určena pouze pro čtení.
ColCount	Počet zobrazovaných sloupců. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetColCount .
ItemCount	Celkový počet položek. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetItemCount .
Selected	Aktuálně vybraná položka. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SelectItem .

Následující obrázek znázorňuje význam položek LeftOffset, TopOffset, RightOffset, BottomOffset a ItemHeight.



Komponenta **TListBox** rozšiřuje bázovou třídu **TControl** o metodu uvedené v následující tabulce:

SetItemCount	Metoda pro nastavení celkového počtu položek
SelectItem	Metoda pro nastavení aktuálně vybrané položky
SetFlags	Metoda pro nastavení položky Flags
SetColCount	Metoda pro nastavení počtu zobrazovaných sloupců
SetOffsets	Metoda pro nastavení položek LeftOffset, TopOffset, RightOffset a BottomOffset
SetItemHeight	Metoda pro nastavení položky ItemHeight

4.12.2. Implicitní nastavení komponenty TListBox

V následující tabulce jsou uvedena implicitní nastavení všech položek komponenty. Všechny ostatní položky, které nejsou v tabulce uvedené, jsou inicializovány na hodnotu 0, **nil**, apod.

Položka	Hodnota
Bounds	parametr ABounds konstrukturu
EventMask	\$FFFF
CaretSize	(1, 8)
State	sfVisible
Options	ofSelectable or ofSharedPalette or ofFirstClick
Font	fidDefault
Palette	#\$0F#\$00#\$00#\$00#\$00#\$00#\$0F#\$0F#\$00 (sdílená paleta)
Flags	0
ColCount	1
Selected	-1
LeftOffset	1
TopOffset	1
RightOffset	1
BottomOffset	1
ItemHeight	0
ScrollBar	podle parametru AScrollBar konstrukturu

Pro komponentu **TListBox** jsou definovány konstanty nastavení třídy uvedené v následující tabulce. Tyto konstanty lze předat jako parametr metodě **Customize** (viz. manuál ke knihovně Controls).

Konstanta	Význam
ccListBox	Implicitní nastavení definované ovladačem použitého terminálu.

4.12.3. Události obsluhované komponentou TListBox

Komponenta **TListBox** obsluhuje vstupní události podle následující tabulky

Událost	Popis činnosti	
evMouseDown evMouseRep evMouseDown	Kliknutí do oblasti položky způsobí její výběr.	
evMouseDbl	Dvojklik do oblasti položky vyvolá oznámení <code>nmSelect</code> .	
evBroadcast <code>cmScrollBarChanged</code>	Zpráva přicházející od přidruženého posuvníku (komponenty <code>TScrollBar</code> v položce <code>ScrollBar</code>) způsobí změnu položky <code>TopItem</code> , tj. posunutí viditelného výřezu z celkového počtu položek.	
evKeyDown	<code>vkLeft</code>	<code>SetSelected(Selected - RowCount)</code>
	<code>vkRight</code>	<code>SetSelected(Selected + RowCount)</code>
	<code>vkUp</code>	<code>SetSelected(Selected - 1)</code>
	<code>vkDown</code>	<code>SetSelected(Selected + 1)</code>
	<code>vkPageUp</code>	<code>SetSelected(Selected - RowCount * ColCount)</code>
	<code>vkPageDown</code>	<code>SetSelected(Selected + RowCount * ColCount)</code>
	<code>vkHome</code>	<code>SetSelected(0)</code>
	<code>vkEnd</code>	<code>SetSelected(ItemCount - 1)</code>
<code>vkEnter</code>	Pokud je nastaven příznak <code>lbfUseEnter</code> v položce <code>Flags</code> , pak stisk klávesy <code>vkEnter</code> způsobí vyvolání oznámení <code>nmSelect</code> .	

Pozn.: `RowCount` je počet zobrazovaných řádek. Klávesy `vkLeft` a `vkRight` jsou obsluhovány pouze tehdy, jestliže počet zobrazovaných sloupců (`ColCount`) je větší než jedna.

4.12.4. Oznámení generovaná komponentou TListBox

Komponenta generuje následující oznámení:

Oznámení	Okamžik vyvolání	Parametry oznámení
<code>nmEnter</code>	Komponenta se dostane do ohniska	-
<code>nmExit</code>	Komponenta se dostane mimo ohnisko	-
<code>nmHide</code>	Volání metody <code>Hide</code>	-
<code>nmShow</code>	Volání metody <code>Show</code>	-
<code>nmEnable</code>	Volání metody <code>Enable</code>	-
<code>nmDisable</code>	Volání metody <code>Disable</code>	-
<code>nmCanExit</code>	Před opuštěním ohniska, pokud je nastaven příznak <code>ofValidate</code> v položce <code>Options</code>	Položka <code>Accept</code> struktury oznámení obsahuje hodnotu <code>False</code> . Opuštění ohniska lze zakázat nastavením této položky na hodnotu <code>False</code> .
<code>nmChange</code>	Změna vybrané položky	-

nmSelect	Výběr položky	-
nmGetData	Požadavek na uživatelská data, pokud není nastaven příznak <code>lbfLookup</code> v položce <code>Flags</code>	V položce <code>Index</code> struktury oznámení je uložen index položky. V položce <code>Text</code> je hodnota <code>nil</code> . Obsluha oznámení musí naplnit položku <code>Text</code> ukazatelem na znakový řetězec, který má být zobrazen v políčku daném položkou <code>Index</code> . Viz. kapitola 4.12.5.2.

4.12.5. Data zobrazovaná komponentou `TListBox`

Komponenta `TListBox` sama neudrhuje ani neobsahuje data, která zobrazuje. Existují dvě varianty naplnění komponenty daty. Buď lze využít metody `EnumLookup` validátoru, která vrací v textové podobě název varianty výčtového typu, nebo lze využít oznámení `nmGetData`. Obě varianty jsou popsány v následujících dvou kapitolách.

4.12.5.1. Použití validátoru jako zdroje dat

Validátor ordinálního typu, tj. jakýkoli potomek třídy `TOrdinalValidator` obsahuje metodu `GetMinMax` a `EnumLookup`. Metoda `GetMinMax` zjišťuje minimální a maximální ordinální hodnotu validované proměnné a metoda `EnumLookup` zjišťuje názvy jednotlivých variant výčtového typu.

Pokud je nastaven příznak `lbfLookup` v položce `Flags` pak po přiřazení validátoru komponentě `TListBox` metodou `SetValidator` je automaticky nastaven počet položek listboxu podle maximální hodnoty navrácené metodou `GetMinMax` validátoru a při každém vykreslení položky metoda `GetData` volá metodu `EnumLookup` validátoru. Komponenta listbox tedy obsahuje všechny varianty výčtového typu.

4.12.5.2. Použití oznámení `nmGetData` jako zdroje dat

Při každém požadavku na zobrazení určité položky je vyvolání oznámení `nmGetData`, jehož úkolem je získat text, který má být zobrazen. Oznámení `nmGetData` je automaticky generováno při překreslování komponenty, při výběru položku apod.

Oznámení `nmGetData` má následující parametry:

Položka	Obsah
<code>Code</code>	Konstant <code>nmGetData</code>
<code>Control</code>	Odkaz na instanci komponenty <code>TListBox</code>
<code>Index</code>	Index položky, jejíž data jsou požadována
<code>Text</code>	Položka, která má být obsluhou oznámení vyplněna, tak aby obsahovala ukazatel na znakový řetězec.

Obsluha oznámení může vypadat např. následovně:

```
const
  MyListBoxText: array[0..5] of string[10] =
```

```

    ( 'Polozka_0', 'Polozka_1',
      'Polozka_2', 'Polozka_3',
      'Polozka_4', 'Polozka_5' );

const
  cidMyListBox = 1000;

procedure NotificationHandler( N : TNotification );
begin
  case N.Control^.Id of
    cidMyListBox:
      begin
        case N.Code of
          nmGetData:
            begin
              N.Text := @MyListBoxText[ N.Index ];
              ClearNotification( N );
            end;
          end;
        end;
      end;
  end;
end;

```

Je potřeba si uvědomit, že znakový řetězec musí být platný i při opuštění procedury NotificationHandler, tzn. že ukazatel uložený do položky Text struktury oznámení se nesmí odkazovat na řetězec alokovaný na zásobníku. Není tedy možné

```

procedure NotificationHandler( N : TNotification );
var
  S : string[10];
begin
  case N.Control^.Id of
    cidMyListBox:
      begin
        case N.Code of
          nmGetData:
            begin
              S := ...;
              N.Text := @S;  { !!! promenna S nemuze byt alokovana na
                             zasobniku, protoze po navratu z teto
                             procedury je obsah lokalni promenne
                             nedefinovan }
              ClearNotification( N );
            end;
          end;
        end;
      end;
  end;
end;

```

Pokud by však proměnná S byla alokována jako lokální typová konstanta, pak je vše v pořádku.

4.12.6. Validace hodnoty komponenty TListBox

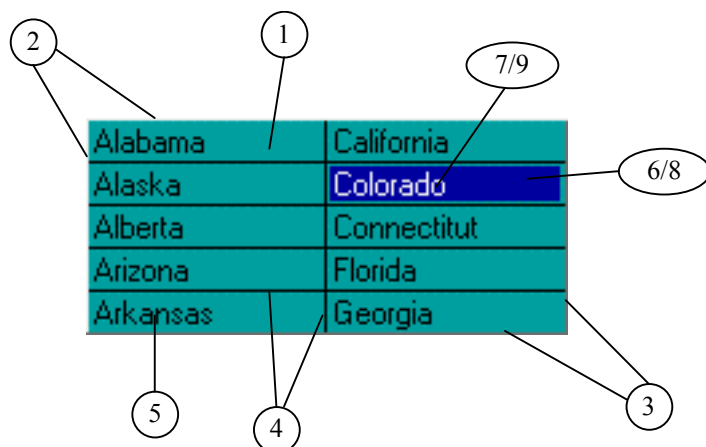
S komponentu **TListBox** lze použít všechny typy validátorů implementujících metodu **TransferOrdinal**, tj. všechny potomky třídy **TOrdinalValidator**. Hodnota validované proměnné koresponduje s vybranou položku.

Pokud je v položce Flags nastaven příznak `lbfLookup`, pak se předpokládá použití validátoru odvozeného ze třídy `TEnumValidator`.

4.12.7. Paleta komponenty TListBox

Konstruktor komponenty `TListBox` nastavuje implicitně paletu barev komponenty podle následující tabulky:

#	Barva	Popis
1	\$0F	Pozadí
2	■ \$00	Linka rámečku (vlevo a nahoře)
3	■ \$00	Linka rámečku (vpravo a dole)
4	■ \$00	Vertikální a horizontální oddělovací čáry
5	■ \$00	Text položky
6	■ \$00	Pozadí vybrané položky (komponenta mimo ohnisko)
7	\$0F	Text vybrané položky (komponenta mimo ohnisko)
8	\$0F	Pozadí vybrané položky (komponenta v ohnisku)
9	■ \$00	Text vybrané položky (komponenta v ohnisku)



4.12.8. Reference

Třída `TListBox` je potomkem třídy `TControl` popsané v dokumentaci ke knihovně Controls.

```

PListBox = ^TListBox;
TListBox = object( TControl )
public
  Flags          : Word;
  TopOffset      : Integer;
  LeftOffset     : Integer;
  RightOffset    : Integer;
  BottomOffset   : Integer;
  ItemHeight     : Integer;
  ColCount       : Integer;
  ScrollBar      : PScrollBar;

```

```

ItemCount      : Integer;
TopItem        : Integer;
Selected       : Integer;

constructor Init( const ABounds: TRect; AScrollBar: PScrollBar );
procedure Paint( ACanvas: PCanvas ); virtual;
procedure HandleEvent( var AEvent: TEvent ); virtual;
procedure SetState( AState: Word; AEnable: Boolean ); virtual;
procedure SetValidator( AValidator: PValidator ); virtual;
function GetData( AItemIndex: Integer ): PString; virtual;
procedure GetItemBounds( AItemIndex: Integer;
    var ABounds: TRect );
procedure SetItemCount( AItemCount: Integer );
procedure SelectItem( AItemIndex: Integer );
procedure SetFlags( ASet, AReset: Word ); virtual;
procedure SetColCount( AColCount: Integer );
procedure SetOffsets( ALeft, ATop, ARight, ABottom: Integer );
procedure SetItemHeight( AHeight: Integer );
procedure PaintItems( AItemIndex, ACount: Integer );
procedure PaintAllItems;
function ItemAtPos( APos: TPoint ): Integer;
end;

```

4.12.8.1. Konstanty lbf_

Konstanty s prefixem lbf_ upřesňují chování komponenty **TListBox**. Kombinace konstant je uložena v položce **Flags** této komponenty (viz. kapitola 4.10.7.8).

Identifikátor	Kód	Popis
lbfBorder	\$0001	Zobrazení rámečku kolem komponenty
lbfAlignMask	\$0006	Maska pro příznaky lbfLeft, lbfCenter a lbfRight
lbfLeft	\$0000	Zarovnání textu položek k levému okraji
lbfCenter	\$0002	Zarovnání textu položek k pravému okraji
lbfRight	\$0004	Zarovnání textu položek na střed
lbfVertDiv	\$0008	Zobrazení vertikálních oddělovacích čar
lbfHorzDiv	\$0010	Zobrazení horizontálních oddělovacích čar
lbfUseEnter	\$0020	Obsluha klávesy vkEnter, která generuje oznámení nmSelect.
lbfLookup	\$0040	

Příznaky lbfLeft, lbfRight a lbfCenter nelze navzájem kombinovat.

4.12.8.2. Položka TListBox.Flags

Položka **Flags** upřesňuje chování komponenty. Obsahuje kombinaci příznaků s prefixem lbf_ popsaných v kapitole 4.12.8.1. Položka je určena pouze pro čtení. Pro její nastavení slouží metodu **SetFlags** (viz. kapitola 4.12.8.18).

```
Flags          : Word;
```

4.12.8.3. Položka TListBox.TopOffset

Položka **TopOffset** obsahuje počet pixelů posunutí oblasti pro zobrazení textu položky od horního okraje buňky. Viz. obrázek v kapitole 4.12.1. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetOffsets** (viz. kapitola 4.12.8.20).

```
TopOffset      : Integer;
```

4.12.8.4. Položka TListBox.LeftOffset

Položka **LeftOffset** obsahuje počet pixelů posunutí oblasti pro zobrazení textu položky od levého okraje buňky. Viz. obrázek v kapitole 4.12.1. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetOffsets** (viz. kapitola 4.12.8.20).

```
LeftOffset     : Integer;
```

4.12.8.5. Položka TListBox.RightOffset

Položka **RightOffset** obsahuje počet pixelů posunutí oblasti pro zobrazení textu položky od pravého okraje buňky. Viz. obrázek v kapitole 4.12.1. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetOffsets** (viz. kapitola 4.12.8.20).

```
RightOffset    : Integer;
```

4.12.8.6. Položka TListBox.BottomOffset

Položka **BottomOffset** obsahuje počet pixelů posunutí oblasti pro zobrazení textu položky od pravého okraje buňky. Viz. obrázek v kapitole 4.12.1. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetOffsets** (viz. kapitola 4.12.8.20).

```
BottomOffset   : Integer;
```

4.12.8.7. Položka TListBox.ItemHeight

Položka **ItemHeight** obsahuje výšku oblasti (tj. počet pixelů) pro zobrazení textu položky. Pokud je hodnota této položky menší nebo rovna nule, pak je výška této oblasti dána výškou znaků použitého fontu zvětšená o absolutní hodnotu této položky. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetItemHeight** (viz. kapitola 4.12.8.21).

```
ItemHeight     : Integer;
```

4.12.8.8. Položka TListBox.ColCount

Položka **ColCount** obsahuje počet zobrazovaných sloupců. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetColCount** (viz. kapitola 4.12.8.19).

4.12.8.9. Položka TListBox.ScrollBar

Položka **ScrollBar** obsahuje odkaz na přidruženou komponentu **TScrollBar**. Položka je inicializovaná parametrem konstruktora a je určena pouze pro čtení.

```
ScrollBar      : PScrollBar;
```

4.12.8.10. Položka TListBox.ItemCount

Položka **ItemCount** obsahuje celkový počet zobrazovaných položek. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetItemCount** (viz. kapitola

4.12.8.16).

```
ItemCount      : Integer;
```

4.12.8.11. Položka TListBox.TopItem

Položka **TopItem** obsahuje index první viditelné položky zobrazené v první řádce prvního sloupce komponenty. Položka je aktualizována automaticky a je určena pouze pro čtení.

```
TopItem        : Integer;
```

4.12.8.12. Položka TListBox.Selected

Položka **Selected** obsahuje index vybrané položky. Položka je aktualizována automaticky při procházení položek komponenty. Je určena pouze pro čtení. Pro její explicitní nastavení slouží metoda **SelectItem** (viz. kapitola 4.12.8.17).

```
Selected       : Integer;
```

4.12.8.13. Konstruktor TListBox.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```
constructor Init( const ABounds: TRect; AScrollBar: PScrollBar );
```

Parametry:

ABounds	Umístění a rozměry komponenty v rámci jejího vlastníka.
AScrollBar	Odkaz na přidružený posuvník, tj. komponentu TScrollBar . Parametr může mít hodnotu nil .

Návratové hodnoty:

Konstruktor nevrací žádnou hodnotu.

Poznámky:

Konstruktor **Init** nastaví všechny položky instance na implicitní hodnoty, viz. tabulka v kapitole 4.12.2.

4.12.8.14. Metoda TListBox.GetData

Metoda **GetData** vrací text položky se zadaným indexem. Metoda je volána automaticky pokaždé, když komponenta potřebuje vykreslit konkrétní položku.

```
function GetData( AItemIndex: Integer ): PString; virtual;
```

Parametry:

AItemIndex	Index položky. Hodnota parametru musí být větší nebo rovna
------------	--

nule a menší, než je celkový počet položek.

Návratové hodnoty:

Metoda vrací ukazatel na znakový řetězec nebo hodnotu **nil**, pokud má být položka prázdná.

Poznámky:

Metoda **GetData** je definována jako virtuální. Potomci třídy **TListBox** ji mohou předefinovat.

Implicitní chování metody **GetData** definované ve třídě **TListBox** je následující:

Pokud je nastaven příznak **lbfLookup** v položce **Flags**, pak metoda vrací hodnotu získanou pomocí metody **EnumLookup** validátoru připojeného ke komponentě, tj. **Listbox** zobrazuje varianty výčtového typu.

Pokud není nastaven příznak **lbfLookup** v položce **Flags**, pak metoda vyvolá oznámení **nmGetData** s nastaveným parametrem **Index**. Obsluha oznámení musí vyplnit položku **Text** oznámení a metoda **GetData** vrátí obsah této položky.

4.12.8.15. Metoda **TListBox.GetItemBounds**

Metoda **GetItemBounds** slouží ke zjištění umístění a rozměrů obdélníku pro zobrazení textu položky.

```
procedure GetItemBounds( AItemIndex: Integer; var ABounds: TRect );
```

Parametry:

AItemIndex	Index položky. Hodnota parametru musí být větší nebo rovna nule a menší, než je celkový počet položek.
ABounds	Parametr, který bude naplněn umístěním a rozměry obdélníku položky.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud položka se zadaným indexem není aktuálně viditelná. Pak bude po provedení metody parametr **ABounds** prázdný, tj. všechny souřadnice obdélníku budou nastaveny na hodnotu nula.

4.12.8.16. Metoda **TListBox.SetItemCount**

Metoda **SetItemCount** slouží k nastavení celkového počtu zobrazovaných položek, tj. nastavení položky **ItemCount** (viz. kapitola 4.12.8.10)

```
procedure SetItemCount( AItemCount: Integer );
```

Parametry:

AItemCount Celkový počet položek. Hodnota parametru musí být větší nebo rovna nule.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **PaintAllItems**.

4.12.8.17. Metoda TListBox.SelectItem

Metoda **SelectItem** slouží k nastavení aktuálně vybrané položky, tj. nastavení položky **Selected** (viz. kapitola 4.12.8.12)

```
procedure SelectItem( AItemIndex: Integer );
```

Parametry:

AItemIndex Index požadované položky. Hodnota parametru musí být větší nebo rovna nule a menší než je celkový počet položek.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

4.12.8.18. Metoda TListBox.SetFlags

Metoda **SetFlags** slouží k nastavení položky **Flags** (viz. kapitola 4.12.8.2)

```
procedure SetFlags( ASet, AReset: Word ); virtual;
```

Parametry:

AFlags Kombinace příznaků s prefixem **lbf_** (viz. kapitola 4.12.8.1).

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

4.12.8.19. Metoda TListBox.SetColCount

Metoda **SetColCount** slouží k nastavení počtu zobrazovaných sloupců, tj. položky ColCount (viz. kapitola 4.12.8.8)

```
procedure SetColCount( AColCount: Integer );
```

Parametry:

AColCount Počet zobrazovaných sloupců. Hodnota parametru musí být větší než nula.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

4.12.8.20. Metoda TListBox.SetOffsets

Metoda **SetOffsets** slouží k nastavení posunutí oblasti pro zobrazení textu položky od okrajů buňky, tj. položek LeftOffset, TopOffset, RightOffset a BottomOffset.

```
procedure SetOffsets( ALeft, ATop, ARight, ABottom: Integer );
```

Parametry:

ALeft	Posunutí od levého okraje buňky
ATop	Posunutí od horního okraje buňky
ARight	Posunutí od pravého okraje buňky
ABottom	Posunutí od spodního okraje buňky

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

4.12.8.21. Metoda TListBox.SetItemHeight

Metoda **SetItemHeight** slouží k nastavení výšky oblasti pro zobrazení textu položky, tj. nastavení položky ItemHeight (viz. kapitola 4.12.8.7).

```
procedure SetItemHeight( AHeight: Integer );
```

Parametry:

AHeight Parametr udává výšku oblasti (tj. počet pixelů) pro zobrazení textu položky. Pokud je hodnota tohoto parametru menší nebo rovna nule, pak je výška oblasti dána výškou znaků použitého fontu zvětšená o absolutní hodnotu této položky.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

4.12.8.22. Metoda TListBox.PaintItems

Metoda **PaintItems** překreslí obsah všech viditelných položek v zadaném rozsahu

```
procedure PaintItems( AItemIndex, ACount: Integer );
```

Parametry:

AItemIndex První vykreslovaná položka.
ACount Počet vykreslovaných položek.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Metoda **PaintItems** vykresluje pouze text a pozadí položek. Ze zadaného rozsahu jsou vykresleny pouze viditelné položky.

4.12.8.23. Metoda TListBox.PaintAllItems

Metoda **PaintAllItems** překreslí obsah všech viditelných položek.

```
procedure PaintAllItems;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Metoda **PaintAllItems** překreslí obsah všech viditelných položek. Vykresluje pouze text položek a proto je obvykle podstatně rychlejší než metoda **Repaint** překreslující celou komponentu včetně rámečku, oddělovacích čar apod.

4.12.8.24. Metoda TListBox.ItemAtPos

Metoda **ItemAtPos** vrací index položky, která leží na zadaných souřadnicích.

```
function ItemAtPos( APos: TPoint ): Integer;
```

Parametry:

APos Parametr obsahuje lokální souřadnice bodu.

Návratové hodnoty:

Metoda vrací index položky. V případě, že žádná položka neobsahuje zadaný bod, pak metoda vrací hodnotu -1.

Poznámky:

4.13. Komponenta TListView

Komponenta **TListView** je určena z zobrazení a k editaci seznamu strukturovaných položek. Na následujícím obrázku je ukázka této komponenty.

Goods	Price	
CPU Amd K7 Duron 1400	1078	-7.80%
CPU Amd K7 Duron 1600	1216	-12.48%
CPU Amd Athlon XP 1700+	1602	-10.50%
CPU Amd Athlon XP 1900+	1642	-8.20%

U komponenty **TListView** lze nastavovat několik parametrů, které ovlivňují její chování, příp. vykreslování. Lze nastavit:

- Zobrazení rámečku kolem komponenty
- Zobrazení záhlaví položek
- Nezávisle výšku záhlaví a výšku položek
- Zobrazení svislých a vodorovných oddělovacích čar
- Odsazení textu položky shora a zdola
- Označení celého řádku nebo konkrétního sloupce řádku

Zobrazované sloupce se definují pomocí spojového seznamu struktur **TListColumn**, který se vytváří pomocí funkce **NewListColumn** (viz. kapitola 4.13.2). U jednotlivých sloupců komponenty **TListView** lze nastavit:

- Text záhlaví

- Šířku
- Zarovnání textu doleva, doprava nebo na střed
- Posunutí oblasti pro zobrazení textu položky od jejího levého a pravého okraje.
- Zda lze sloupec vybrat

4.13.1. Položky a metody třídy TListView

Třída **TListView** má následující konstruktor:

```
constructor Init( const ABounds: TRect; AScrollBar: PScrollBar;
                  AColumns: PListColumn );
```

Popis parametrů konstruktoru je uveden v následující tabulce:

ABounds	Umístění a rozměry komponenty v rámci vlastníka
AScrollBar	Odkaz na přidruženou komponentu TScrollBar. Položka je inicializovaná parametrem konstruktoru a je určena pouze pro čtení.
AColumns	Odkaz na spojový seznam popisující zobrazované sloupce. Viz. kapitola 4.13.2

Komponenta **TListView** rozšiřuje báзовou třídu **TControl** o položky uvedené v následující tabulce:

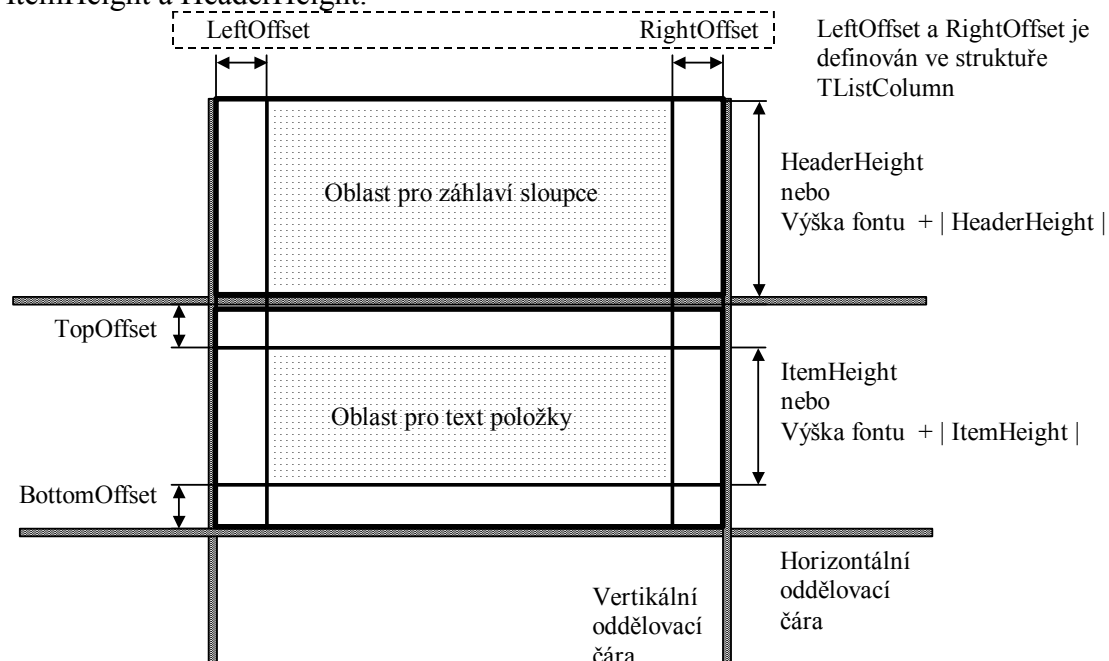
Flags	Příznaky upřesňující chování komponenty. Položka obsahuje kombinaci následujících příznaků:	
	lvfBorder	Zobrazení rámečku kolem komponenty
	lvfHorzDiv	Zobrazení horizontálních oddělovacích čar
	lvfVertDiv	Zobrazení vertikálních oddělovacích čar
	lvfHeader	Zobrazení řádku záhlaví
	lvfRowSelect	Výběr celých řádku (vs. výběr jednotlivých položek ve sloupcích)
	lvfUseEnter	Obsluha klávesy vkEnter, která generuje oznámení nmSelect.
Položka Flags je určena pouze pro čtení. Pro její nastavení použijte metodu SetFlags.		
TopOffset	Posunutí oblasti pro zobrazení textu položky od jejího horního okraje. Viz. obrázek níže. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetOffsets .	
BottomOffset	Posunutí oblasti pro zobrazení textu položky od jejího levého okraje. Viz. obrázek níže. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetOffsets .	
ItemHeight	Výška oblasti pro zobrazení textu položky. Pokud je hodnota této položky menší nebo rovna nule, pak je výška této oblasti dána výškou znaků použitého fontu zvětšená o absolutní hodnotu této položky. Položka je určena pouze pro čtení. Pro její nastavení	

	použijte metodu SetItemHeight .
HeaderHeight	Výška řádku záhlaví. Pokud je hodnota této položky menší nebo rovna nule, pak je výška této oblasti dána výškou znaků použitého fontu zvětšená o absolutní hodnotu této položky. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetItemHeight .
Columns	Odkaz na spojový seznam popisující zobrazované sloupce. Viz. kapitola 4.13.2
ScrollBar	Odkaz na přidruženou komponentu TScrollBar. Položka je inicializovaná parametrem konstruktora a je určena pouze pro čtení.
Selected	Index aktuálně vybrané položky (řádku). Položka je určena pouze pro čtení.
SelColumn	Index aktuálně vybraného sloupce.
ItemCount	Celkový počet položek (řádků). Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetItemCount .

Komponenta **TListView** rozšiřuje básovou třídu **TControl** o metodu uvedené v následující tabulce:

SetItemCount	Metoda pro nastavení celkového počtu položek (řádek)
SelectItem	Metoda pro nastavení vybrané položky (řádku) a vybraného sloupce.
SetFlags	Metoda pro nastavení položky Flags.
SetOffsets	Metoda pro nastavení položek TopOffset a BottomOffset
SetItemHeight	Metoda pro nastavení položky ItemHeight
SetHeaderHeight	Metoda pro nastavení položky HeaderHeight

Následující obrázek znázorňuje význam položek TopOffset, , BottomOffset, ItemHeight a HeaderHeight.



4.13.2. Definování sloupců komponenty TListView

Jednotlivé sloupce zobrazené v komponentě **TListView** jsou popsány pomocí spojového seznamu struktur **TListColumn** (viz. kapitola 4.13.9.3). Struktura **TListColumn** obsahuje následující položky:

Položka	Význam								
Next	Odkaz na následující strukturu ve spojovém seznamu								
Name	Název sloupce zobrazovaný v záhlaví komponenty TListView								
Width	Šířka sloupce v pixelech. Pokud je hodnota menší nebo rovna nule, pak je šířka sloupce vypočítána z šířky textu záhlaví, ke které je přičtena absolutní hodnota této položky.								
Flags	Příznaky upřesňující vlastnosti sloupce. Položka obsahuje kombinaci následujících příznaků (viz. kapitola 4.13.9.2): <table border="1" data-bbox="421 712 1353 869"> <tbody> <tr> <td>lcfLeft</td> <td>Zarovnání textu ve sloupci doleva</td> </tr> <tr> <td>lcfRight</td> <td>Zarovnání textu ve sloupci doprava</td> </tr> <tr> <td>lcfCenter</td> <td>Zarovnání textu ve sloupci na střed</td> </tr> <tr> <td>lcfSelectable</td> <td>Sloupec lze vybrat</td> </tr> </tbody> </table> <p>Příznaky lcfLeft, lcfRight a lcfCenter nelze mezi sebou kombinovat.</p>	lcfLeft	Zarovnání textu ve sloupci doleva	lcfRight	Zarovnání textu ve sloupci doprava	lcfCenter	Zarovnání textu ve sloupci na střed	lcfSelectable	Sloupec lze vybrat
lcfLeft	Zarovnání textu ve sloupci doleva								
lcfRight	Zarovnání textu ve sloupci doprava								
lcfCenter	Zarovnání textu ve sloupci na střed								
lcfSelectable	Sloupec lze vybrat								
LeftOffset	Posunutí oblasti pro zobrazení textu položky od jejího levého okraje.								
RightOffset	Posunutí oblasti pro zobrazení textu položky od jejího pravého okraje.								

K vytvoření spojového seznamu struktur **TListColumn** slouží funce **NewListColumn** (viz. kapitola 4.13.9.4) definovaná následovně:

```
function NewListColumn( const AName: string; AFlags: Word;
    AWidth: Integer; ANext: PListColumn ): PListColumn;
```

Příklad: Definování třech sloupců

```
Columns :=
  NewListColumn( 'Parameter', lcfLeft, 160,
  NewListColumn( 'Value', lcfRight or lcfSelectable, 60,
  NewListColumn( 'Unit', lcfLeft, 30,
  nil ));
```

4.13.3. Implicitní nastavení komponenty TListView

V následující tabulce jsou uvedena implicitní nastavení všech položek komponenty. Všechny ostatní položky, které nejsou v tabulce uvedené, jsou inicializovány na hodnotu 0, **nil**, apod.

Položka	Hodnota
Bounds	parametr ABounds konstrukturu
EventMask	\$FFFF
CaretSize	(1, 8)
State	sfVisible
Options	ofSelectable or ofSharedPalette or ofFirstClick
Font	fidDefault

Palette	#\$0F#\$00#\$00#\$00#\$00#\$0F#\$00#\$00#\$0F#\$00#\$0F (sdílená paleta)
Flags	lvfBorder or lvfHeader
Columns	podle parametru AColumns konstrukturu
TopOffset	1
BottomOffset	1
HeaderHeight	0
ItemHeight	0
ScrollBar	podle parametru AScrollBar konstrukturu
Selected	0
SelColumn	0
ItemCount	0

Pro komponentu **TListView** jsou definovány konstanty nastavení třídy uvedené v následující tabulce. Tyto konstanty lze předat jako parametr metodě **Customize** (viz. manuál ke knihovně Controls).

Konstanta	Význam
ccListView	Implicitní nastavení definované ovladačem použitého terminálu.

4.13.4. Události obsluhované komponentou TListView

Komponenta TListView obsluhuje vstupní události podle následující tabulky

Událost	Popis činnosti
evMouseDown evMouseRep	Kliknutí do oblasti položky způsobí její výběr.
evMouseDown evMouseDbI	Dvojklik do oblasti položky vyvolá metody DoSelect , která vyvolá oznámení nmSelect.
evBroadcast	cmScrollBarChanged Zpráva přicházející od přidruženého posuvníku (komponenty TScrollBar v položce ScrollBar) způsobí změnu položky TopItem, tj. posunutí viditelného výřezu z celkového počtu položek.
evKeyDown	vkLeft SelectItem(Selected, SelColumn - 1) vkRight SelectItem (Selected, SelColumn + 1) vkUp SelectItem (Selected - 1, SelColumn) vkDown SelectItem (Selected + 1, SelColumn) vkPageUp SelectItem(Selected - RowCount, SelColumn) vkPageDown SelectItem(Selected + RowCount, SelColumn) vkHome SelectItem (0, SelColumn) vkEnd SelectItem (ItemCount - 1, SelColumn) vkEnter Pokud je nastaven příznak lbfUseEnter v položce Flags, pak stisk klávesy vkEnter způsobí vyvolání metody DoSelect , která vyvolá oznámení nmSelect.

Pozn.: RowCount je počet zobrazovaných řádek. Klávesy vkLeft a vkRight jsou

obsluhovány pouze tehdy, jestliže lze vybrat předchozí nebo následující sloupec.

4.13.5. Oznámení generovaná komponentou TListView

Komponenta generuje následující oznámení:

Oznámení	Okamžik vyvolání	Parametry oznámení
nmEnter	Komponenta se dostane do ohniska	-
nmExit	Komponenta se dostane mimo ohnisko	-
nmHide	Volání metody Hide	-
nmShow	Volání metody Show	-
nmEnable	Volání metody Enable	-
nmDisable	Volání metody Disable	-
nmCanExit	Před opuštěním ohniska, pokud je nastaven příznak ofValidate v položce Options	Položka Accept struktury oznámení obsahuje hodnotu False. Opuštění ohniska lze zakázat nastavením této položky na hodnotu False.
nmChange	Změna vybrané položky	
nmSelect	Výběr položky	
nmGetData	Požadavek na uživatelská data	V položce Row struktury oznámení je uložen řádek. V položce Col je uložen sloupec. V položce Text je hodnota nil. Obsluha oznámení musí naplnit položku Text ukazatelem na znakový řetězec, který má být zobrazen v políčku daném položkami Row a Col. Viz. kapitola 4.13.6.

4.13.6. Data zobrazovaná komponentou TListView

Při každém požadavku na zobrazení určité položky je vyvolání oznámení nmGetData, jehož úkolem je získat text, který má být zobrazen. Oznámení nmGetData je automaticky generováno při překreslování komponenty, při výběru položku apod.

Oznámení nmGetData má následující parametry:

Položka	Obsah
Code	Konstant nmGetData
Control	Odkaz na instanci komponenty TListBox
Row	Index řádku
Col	Index sloupce
Text	Položka, která má být obsluhou oznámení vyplněna, tak aby obsahovala ukazatel na znakový řetězec.

Obsluha oznámení může vypadat např. následovně:

```

const
  MyListViewText: array[0..2, 0..5] of string[10] =
    ( ('Field_0_0', 'Field_1_0', 'Field_2_0' ),
      ('Field_0_1', 'Field_1_1', 'Field_2_1' ),
      ('Field_0_2', 'Field_1_2', 'Field_2_2' ),
      ('Field_0_3', 'Field_1_3', 'Field_2_3' ),
      ('Field_0_4', 'Field_1_4', 'Field_2_4' ),
      ('Field_0_5', 'Field_1_5', 'Field_2_5' )
    );

const
  cidMyListView = 1000;

procedure NotificationHandler( N : TNotification );
begin
  case N.Control^.Id of
    cidMyListView:
      begin
        case N.Code of
          nmGetData:
            begin
              N.Text := @MyListViewText[ N.Row, N.Col ];
              ClearNotification( N );
            end;
          end;
        end;
      end;
    end;
end;

```

Je potřeba si uvědomit, že znakový řetězec musí být platný i při opuštění procedury NotificationHandler, tzn. že ukazatel uložený do položky Text struktury oznámení se nesmí odkazovat na řetězec alokovaný na zásobníku.

4.13.7. Validace hodnoty komponenty TListView

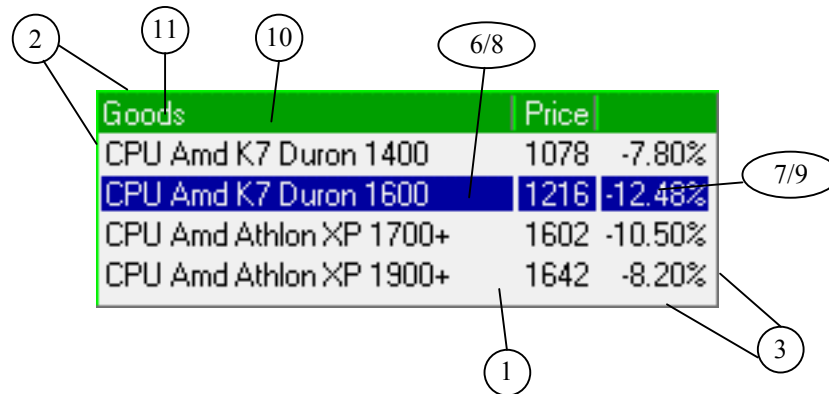
S komponentu **TListBox** lze použít všechny typy validátorů implementujících metodu **TransferOrdinal**, tj. všechny potomky třídy **TOrdinalValidator**. Hodnota validované proměnné koresponduje s vybranou položku.

4.13.8. Paleta komponenty TListView

Konstruktor komponenty **TListView** nastavuje implicitně paletu barev komponenty podle následující tabulky:

#	Barva	Popis
1	\$0F	Pozadí
2	■ \$00	Linka rámečku (vlevo a nahoře)
3	■ \$00	Linka rámečku (vpravo a dole)
4	■ \$00	Oddělovací čáry
5	■ \$00	Text položky
6	\$0F	Pozadí vybrané položky (mimo ohnisko)

7	■ \$00	Text vybrané položky (mimo ohnisko)
8	■ \$00	Pozadí vybrané položky (v ohnisku)
9	\$0F	Text vybrané položky (v ohnisku)
10	■ \$00	Pozadí záhlaví
11	\$0F	Text záhlaví



4.13.9. Reference

Třída **TListView** je potomkem třídy **TControl** popsané v dokumentaci ke knihovně Controls.

```

PListView = ^TListView;
TListView = object( TControl )
public
  Flags      : Word;
  Columns    : PListColumn;
  ItemHeight : Integer;
  TopOffset  : Integer;
  BottomOffset : Integer;
  HeaderHeight : Integer;
  TopItem    : Integer;
  Selected   : Integer;
  SelColumn  : Integer;
  ItemCount  : Integer;
  ScrollBar  : PScrollBar;

  constructor Init( const ABounds: TRect; AScrollBar: PScrollBar;
                   AColumns: PListColumn );
  destructor Done; virtual;
  procedure Paint( ACanvas: PCanvas ); virtual;
  procedure HandleEvent( var AEvent: TEvent ); virtual;
  procedure SetState( AState: Word; AEnable: Boolean ); virtual;

  procedure SetItemCount( AItemCount: Integer );
  procedure SelectItem( AItemIndex, AColIndex: Integer );
  procedure SetFlags( ASet, AReset: Word ); virtual;
  procedure SetOffsets( ATopOffset, ABottomOffset: Integer );
  procedure SetItemHeight( AHeight: Integer );
  procedure SetHeaderHeight( AHeight: Integer );
  procedure GetItemBounds( AItemIndex, AColIndex: Integer;
                          var ABounds: TRect );

  function GetRowHeight: Integer;
  function GetColWidth( AColumn: PListColumn ): Integer;
  function GetHeaderHeight: Integer;

```

```

function GetRowCount: Integer;

function GetData( AItemIndex, AColIndex:
    Integer ): PString; virtual;
procedure DoSelect; virtual;

procedure PaintItems( AItemIndex, ACount: Integer );
procedure PaintAllItems;
function ItemAtPos( APos: TPoint; var AColIndex:
    Integer ): Integer;

procedure UpdateScrollBar;
end;

```

4.13.9.1. Konstanty lvf_

Konstanty s prefixem lvf_ upřesňují chování komponenty **TListView**. Kombinace konstant je uložena v položce **Flags** této komponenty (viz. kapitola 4.13.9.6).

Identifikátor	Kód	Popis
lvfBorder	\$0001	Zobrazení rámečku kolem komponenty
lvfHorzDiv	\$0002	Zobrazení horizontálních oddělovacích čar
lvfVertDiv	\$0004	Zobrazení vertikálních oddělovacích čar
lvfHeader	\$0008	Zobrazení řádku záhlaví
lvfRowSelect	\$0010	Výběr celých řádku (vs. výběr jednotlivých položek ve sloupcích)
lvfUseEnter	\$0020	Obsluha klávesy vkEnter, která generuje oznámení nmSelect.

4.13.9.2. Konstanty lcf_

Konstanty s prefixem lcf_ upřesňují vlastnosti sloupce komponenty **TListView**. Kombinace konstant je uložena v položce **Flags** struktury **TListColumn** (viz. kapitola 4.13.9.3).

Identifikátor	Kód	Popis
lcfAlignMask	\$0003	Maska pro příznaky lcfLeft, lcfCenter a lcfRight.
lcfLeft	\$0000	Zarovnání textu sloupce doleva
lcfCenter	\$0001	Zarovnání textu sloupce na střed
lcfRight	\$0002	Zarovnání textu sloupce doprava
lcfSelectable	\$0010	Položku ve sloupci lze vybrat

Příznaky lcfLeft, lcfCenter a lcfRight nelze navzájem kombinovat.

4.13.9.3. Struktura TListColumn

Struktura **TListColumn** popisuje jeden sloupec zobrazený komponentou **TListView**. Struktury **TListColumn** jsou alokovány na zásobníku a tvoří spojový seznam.

```

PListColumn = ^TListColumn;
TListColumn = record
    Next      : PListColumn;
    Name      : PString;
    Width     : Integer;

```

```

Flags      : Word;
LeftOffset : Integer;
RightOffset : Integer;
end;
```

Popis jednotlivých položek struktury je uveden v následující tabulce:

Položka	Popis
Next	Odkaz na následující strukturu TListColumn ve spojovém seznamu. Poslední struktura obsahuje v této položce hodnotu nil .
Name	Název sloupce zobrazovaný v záhlaví. Záhlaví je viditelné pouze tehdy, jestliže je v položce Flags komponenty TListView nastaven příznak lvfHeader.
Width	Šířka sloupce v pixelech. Pokud je hodnota menší nebo rovna nule, pak je šířka sloupce vypočítána z šířky názvu záhlaví, ke které je přičtena absolutní hodnota této položky.
Flags	Položka upřesňující vlastnosti sloupce. Obsahuje kombinaci příznaků s prefixem lcf_ (viz. kapitola 4.13.9.2).
LeftOffset	Posunutí oblasti pro zobrazení textu položky od jejího levého okraje.
RightOffset	Posunutí oblasti pro zobrazení textu položky od jejího pravého okraje.

4.13.9.4. Funkce NewListColumn

Funkce **NewListColumn** slouží k alokování struktury **TListColumn** popisující jeden sloupec komponenty **TListView**.

```

function NewListColumn( const AName: string; AFlags: Word;
  AWidth: Integer; ANext: PListColumn ): PListColumn;
```

Parametry:

AName	Název sloupce zobrazovaný v záhlaví sloupce.
AFlags	Kombinace příznaků s prefixem lcf_ upřesňující vlastnosti sloupce (viz. kapitola 4.13.9.2)
AWidth	Šířka sloupce v pixelech. Pokud je hodnota parametru menší nebo rovna nule, pak je šířka sloupce vypočítána z šířky názvu záhlaví, ke které je přičtena absolutní hodnota tohoto parametru.
ANext	Odkaz na následující strukturu TListColumn nebo hodnota nil .

Návratové hodnoty:

Funkce vrací ukazatel na strukturu **TListColumn** alokovanou na hromadě a inicializovanou dle parametrů této funkce.

Poznámky:

Položky **LeftOffset** a **RightOffset** struktury **TListColumn** jsou nastaveny na implicitní

hodnotu 1.

Strukturu **TListColumn**, resp. celý spojový seznam těchto struktur lze uvolnit pomocí procedure **FreeListColumn** (viz. kapitola 4.13.9.5)

4.13.9.5. Funkce FreeColumn

Procedure **FreeListColumn** slouží k uvolnění spojového seznamu struktur **TListColumn** alokovaných metodou **NewListColumn**.

```
procedure FreeListColumn( AColumn: PListColumn );
```

Parametry:

AColumn Odkaz na první strukturu **TListColumn** ve spojovém seznamu..

Poznámky:

4.13.9.6. Položka TListView.Flags

Položka **Flags** upřesňuje chování komponenty. Obsahuje kombinaci příznaků s prefixem `lvf_` popsanych v kapitole 4.13.9.1. Položka je určena pouze pro čtení. Pro její nastavení slouží metodu **SetFlags** (viz. kapitola 4.13.9.20).

```
Flags                    : Word;
```

4.13.9.7. Položka TListView.Columns

Položka **Columns** obsahuje ukazatel na první položku spojového seznamu struktur **TListColumn** popisující zobrazované sloupce. Položka je inicializovaná parametrem konstrukturu a je určena pouze pro čtení. Destruktor komponenty uvolní spojový seznam z paměti.

```
Columns                : PListColumn;
```

4.13.9.8. Položka TListView.ItemHeight

Položka **ItemHeight** obsahuje výšku oblasti (tj. počet pixelů) pro zobrazení textu položky. Pokud je hodnota této položky menší nebo rovna nule, pak je výška této oblasti dána výškou znaků použitého fontu zvětšená o absolutní hodnotu této položky. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetItemHeight** (viz. kapitola 4.13.9.22).

4.13.9.9. Položka TListView.TopOffset

Položka **TopOffset** obsahuje počet pixelů posunutí oblasti pro zobrazení textu položky od horního okraje buňky. Viz. obrázek v kapitole 4.13.1. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetOffsets** (viz. kapitola 4.13.9.21).

```
TopOffset              : Integer;
```

4.13.9.10. Položka TListView.BottomOffset

Položka **BottomOffset** obsahuje počet pixelů posunutí oblasti pro zobrazení textu položky od pravého okraje buňky. Viz. obrázek v kapitole 4.13.1. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetOffsets** (viz. kapitola 4.13.9.21).

```
BottomOffset      : Integer;
```

4.13.9.11. Položka TListView.HeaderHeight

Položka **HeaderHeight** obsahuje výšku oblasti (tj. počet pixelů) pro zobrazení textu záhlaví položky. Pokud je hodnota této položky menší nebo rovna nule, pak je výška této oblasti dána výškou znaků použitého fontu zvětšená o absolutní hodnotu této položky. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetHeaderHeight** (viz. kapitola 4.13.9.23).

4.13.9.12. Položka TListView.TopItem

Položka **TopItem** obsahuje index první viditelné položky zobrazené v první řádce komponenty. Položka je aktualizována automaticky a je určena pouze pro čtení.

```
TopItem           : Integer;
```

4.13.9.13. Položka TListView.Selected

Položka **Selected** obsahuje index vybrané položky (řádku). Položka je aktualizována automaticky při procházení položek komponenty. Je určena pouze pro čtení. Pro její explicitní nastavení slouží metoda **SelectItem** (viz. kapitola 4.13.9.19).

```
Selected          : Integer;
```

4.13.9.14. Položka TListView.SelColumn

Položka **SelColumn** obsahuje index sloupce aktuálně vybrané položky. Položka je platná pouze tehdy pokud v položce **Flags** není nastaven příznak **lvfRowSelect**. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SelectItem** (viz. kapitola 4.13.9.19).

```
SelColumn         : Integer;
```

4.13.9.15. Položka TListView.ItemCount

Položka **ItemCount** obsahuje celkový počet zobrazovaných položek. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetItemCount** (viz. kapitola 4.13.9.18).

```
ItemCount         : Integer;
```

4.13.9.16. Položka TListView.ScrollBar

Položka **ScrollBar** obsahuje odkaz na přidruženou komponentu **TScrollBar**. Položka je inicializovaná parametrem konstruktora a je určena pouze pro čtení.


```
ScrollBar : PScrollBar;
```

4.13.9.17. Konstruktor TListView.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```
constructor Init( const ABounds: TRect; AScrollBar: PScrollBar;  
                 AColumns: PListColumn );
```

Parametry:

ABounds	Umístění a rozměry komponenty v rámci jejího vlastníka.
AScrollBar	Odkaz na přidružený posuvník, tj. komponentu TScrollBar . Parametr může mít hodnotu nil .
AColumns	Odkaz na první položku spojového seznamu struktur TListColumn vytvořených pomocí funkce NewListColumn (viz. kapitola 4.13.9.4)

Návratové hodnoty:

Konstruktor nevrací žádnou hodnotu.

Poznámky:

Konstruktor **Init** nastaví všechny položky instance na implicitní hodnoty, viz. tabulka v kapitole 4.13.3.

4.13.9.18. Metoda TListView.SetItemCount

Metoda **SetItemCount** slouží k nastavení celkového počtu zobrazovaných položek, tj. nastavení položky **ItemCount** (viz. kapitola 4.13.9.15)

```
procedure SetItemCount( AItemCount: Integer );
```

Parametry:

AItemCount	Celkový počet položek. Hodnota parametru musí být větší nebo rovna nule.
------------	--

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **PaintAllItems**.

4.13.9.19. Metoda TListView.SelectItem

Metoda **SelectItem** slouží k nastavení aktuálně vybrané položky (řádku a sloupce), tj.

nastavení položek Selected a SelColumn (viz. kapitoly 4.13.9.13 a 4.13.9.14).

```
procedure SelectItem( AItemIndex, AColIndex: Integer );
```

Parametry:

AItemIndex	Index požadovaného řádku. Hodnota parametru musí být větší nebo rovna nule a menší než je celkový počet položek (řádků).
AColIndex	Index požadovaného sloupce. Hodnota parametru musí být větší nebo rovna nule a menší než je celkový počet sloupců.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

4.13.9.20. Metoda TListView.SetFlags

Metoda **SetFlags** slouží k nastavení položky Flags (viz. kapitola 4.13.9.20)

```
procedure SetFlags( ASet, AReset: Word ); virtual;
```

Parametry:

AFlags	Kombinace příznaků s prefixem lvf_ (viz. kapitola 4.13.9.1).
--------	--

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

4.13.9.21. Metoda TListView.SetOffsets

Metoda **SetOffsets** slouží k nastavení posunutí oblasti pro zobrazení textu položky od okrajů buňky, tj. položek TopOffset a BottomOffset.

```
procedure SetOffsets( ATopOffset, ABottomOffset: Integer );
```

Parametry:

ATopOffset	Posunutí od horního okraje buňky
ABottomOffset	Posunutí od spodního okraje buňky

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

4.13.9.22. Metoda TListView.SetItemHeight

Metoda **SetItemHeight** slouží k nastavení výšky oblasti pro zobrazení textu položky, tj. nastavení položky `ItemHeight` (viz. kapitola 4.13.9.8).

```
procedure SetItemHeight( AHeight: Integer );
```

Parametry:

AHeight Parametr udává výšku oblasti (tj. počet pixelů) pro zobrazení textu položky. Pokud je hodnota tohoto parametru menší nebo rovna nule, pak je výška oblasti dána výškou znaků použitého fontu zvětšená o absolutní hodnotu této položky.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

4.13.9.23. Metoda TListView.SetHeaderHeight

Metoda **SetHeaderHeight** slouží k nastavení výšky oblasti pro zobrazení textu záhlaví sloupce, tj. nastavení položky `HeaderHeight` (viz. kapitola 4.13.9.11).

```
procedure SetHeaderHeight( AHeight: Integer );
```

Parametry:

AHeight Parametr udává výšku oblasti (tj. počet pixelů) pro zobrazení textu záhlaví sloupce. Pokud je hodnota tohoto parametru menší nebo rovna nule, pak je výška oblasti dána výškou znaků použitého fontu zvětšená o absolutní hodnotu této položky.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

4.13.9.24. Metoda TListView.GetItemBounds

Metoda **GetItemBounds** slouží ke zjištění umístění a rozměrů obdélníku pro

zobrazení textu položky v zadaném řádku a sloupci.

```
procedure GetItemBounds( AItemIndex, AColIndex: Integer;  
                        var ABounds: TRect );
```

Parametry:

AItemIndex	Index položky (řádku). Hodnota parametru musí být větší nebo rovna nule a menší, než je celkový počet položek.
AColIndex	Index sloupce. Hodnota parametru musí být větší nebo rovna nule a menší, než je celkový počet definovaných sloupců.
ABounds	Parametr, který bude naplněn umístěním a rozměry obdélníku položky.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud položka se zadaným indexem řádku a sloupce není aktuálně viditelná. Pak bude po provedení metody parametr ABounds prázdný, tj. všechny souřadnice obdélníku budou nastaveny na hodnotu nula.

4.13.9.25. Metoda TListView.GetRowHeight

Metoda **GetRowHeight** je interní metoda vracující výšku řádku v pixelech.

```
function GetRowHeight: Integer;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda vrací výšku jednoho řádku v pixelech. Výška řádku je součtem výšky oblasti pro zobrazení textu položky, posunutí textu od horního okraje (TopOffset), posunutí textu od dolního okraje (BottomOffset). Do výšky řádku je započítána případná výška horizontálního oddělovače řádků.

Poznámky:

4.13.9.26. Metoda TListView.GetColWidth

Metoda **GetColWidth** je interní metoda vracující šířku zadaného sloupce.

```
function GetColWidth( AColumn: PListColumn ): Integer;
```

Parametry:

AColumn Odkaz na strukturu **TListColumn** příslušného sloupce.

Návratové hodnoty:

Metoda vrací šířku sloupce v pixelech. Do šířky sloupce je započítána případná šířka vertikálního oddělovače sloupců.

Poznámky:

4.13.9.27. Metoda TListView.GetHeaderHeight

Metoda **GetHeaderHeight** je interní metoda vracející výšku záhlaví v pixelech.

```
function GetHeaderHeight: Integer;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda vrací výšku jednoho řádku v pixelech. Výška řádku je součtem výšky oblasti pro zobrazení textu záhlaví, posunutí textu od horního okraje (TopOffset), posunutí textu od dolního okraje (BottomOffset).

Poznámky:

4.13.9.28. Metoda TListView.GetRowCount

Metoda **GetRowCount** je interní metoda vracející počet zobrazovaných řádků.

```
function GetRowCount: Integer;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda vrací počet zobrazovaných řádku komponentou **TListView**.

Poznámky:

4.13.9.29. Metoda TListView.GetData

Metoda **GetData** vrací text položky se zadaným indexem řádku a sloupce. Metoda je volána automaticky pokaždé, když komponenta potřebuje vykreslit konkrétní položku.

```
function GetData( AItemIndex, AColIndex:  
                  Integer ): PString; virtual;
```

Parametry:

ItemIndex	Index položky (řádku). Hodnota parametru musí být větší nebo rovna nule a menší, než je celkový počet položek.
AColIndex	Index slupce. Hodnota parametru musí být větší nebo rovna nule a menší, než je celkový počet sloupců.

Návratové hodnoty:

Metoda vrací ukazatel na znakový řetězec nebo hodnotu **nil**, pokud má být položka prázdná.

Poznámky:

Metoda **GetData** je definována jako virtuální. Potomci třídy **TListView** ji mohou předefinovat.

Implicitní chování metody **GetData** definované ve třídě **TListView** je následující:

Metoda vyvolá oznámení `nmGetData` s nastaveným parametrem `Row` a `Col`, odpovídající parametrům této metody. Obsluha oznámení musí vyplnit položku `Text` oznámení a metoda **GetData** vrátí obsah této položky.

4.13.9.30. Metoda `TListView.DoSelect`

Metoda **DoSelect** je automaticky volána dvojkliku nebo při stisku klávesy `vkEnter`, pokud je v položce `Flags` nastaven příznak `lvfUseEnter`.

```
procedure DoSelect: PString; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Metoda **DoSelect** je definována jako virtuální. Potomci třídy **TListView** ji mohou předefinovat.

Přímo ve třídě **TListView** je metoda **DoSelect** implementována tak, že vyvolá oznámení `nmSelect`.

4.13.9.31. Metoda `TListView.PaintItems`

Metoda **PaintItems** překreslí obsah všech viditelných položek (řádků) v zadaném

rozsahu

```
procedure PaintItems( AItemIndex, ACount: Integer );
```

Parametry:

AItemIndex První vykreslovaná položka (řádek).

ACount Počet vykreslovaných položek.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Metoda **PaintItems** vykresluje pouze text a pozadí položek. Ze zadaného rozsahu jsou vykresleny pouze viditelné položky (řádky).

4.13.9.32. Metoda TListView.PaintAllItems

Metoda **PaintAllItems** překreslí obsah všech viditelných položek (řádků).

```
procedure PaintAllItems;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Metoda **PaintAllItems** překreslí obsah všech viditelných položek. Vykresluje pouze text položek a proto je obvykle podstatně rychlejší než metoda **Repaint** překreslující celou komponentu včetně rámečku, oddělovacích čar apod.

4.13.9.33. Metoda TListView.ItemAtPos

Metoda **ItemAtPos** vrací řádek a sloupec položky, která leží na zadaných souřadnicích.

```
function ItemAtPos( APos: TPoint; var AColIndex:  
                  Integer ): Integer;
```

Parametry:

APos Parametr obsahuje lokální souřadnice bodu.

AColIndex Odkaz na proměnnou, do které bude uložen index sloupce položky.

Návratové hodnoty:

Metoda vrací index položky. V případě, že žádná položka neobsahuje zadaný bod, pak metoda vrací hodnotu -1.

Poznámky:**4.13.9.34. Metoda TListView.UpdateScrollBar**

Metoda **UpdateScrollBar** je interní metoda pro nastavení parametrů přidruženého posuvníku podle aktuálního stavu komponenty.

```
procedure UpdateScrollBar;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:**4.14. Komponenta TListEdit**

Komponenta **TListEdit** je určena pro zobrazení a editaci seznamu položek. Komponenta **TListEdit** je potomkem **TListView** a sdílí prakticky všechny její vlastnosti.

Description	Value	
Max RPM	1400.00	rpm
Gear ratio	2.50	-
Starting ramp	5.00	s
Stopping ramp	8.00	s

U komponenty **TListEdit** lze nastavovat několik parametrů, které ovlivňují její chování, příp. vykreslování. Lze nastavit:

- Zobrazení rámečku kolem komponenty
- Zobrazení záhlaví položek
- Nezávisle výšku záhlaví a výšku položek
- Zobrazení svislých a vodorovných oddělovacích čar
- Odsazení textu položky shora a zdola
- Označení celého řádku nebo konkrétního sloupce řádku

Zobrazované sloupce se definují pomocí spojového seznamu struktur **TListColumn**, který se vytváří pomocí funkce **NewListColumn** (viz. kapitola 4.13.2). U

jednotlivých sloupců komponenty **TListEdit** lze nastavit:

- Text záhlaví
- Šířku
- Zarovnání textu doleva, doprava nebo na střed
- Posunutí oblasti pro zobrazení textu položky od jejího levého a pravého okraje.
- Zda lze sloupec vybrat

Komponenta **TListEdit** musí mít alespoň dva definované sloupce. První sloupec obsahuje obvykle popis zobrazené proměnné. Druhý sloupec slouží k zobrazení a editaci proměnné. Pokud je nadefinován třetí sloupec, pak obsahuje obvykle jednotky editované proměnné.

Jednotlivé řádky komponenty **TListEdit** jsou uloženy ve spojovém seznamu struktur **TListEditItem** (viz. kapitola 4.14.1). Každá z položek tohoto seznamu popisuje text zobrazený v prvním a třetím sloupci, validátor proměnné zobrazené ve druhé sloupci a doplňující příznaky.

Komponenta **TListEdit** je potomkem komponenty **TListView** a dědí všechny její vlastnosti uvedené v kapitolách 4.13.1 až 4.13.8. Třída **TListEdit** předefinovává dvě metody **GetData** a **DoSelect**. Metoda **GetData** se snaží načíst data ze spojového seznamu struktur **TListEditItem**, jinak použije implicitní metodu **GetData** předka, tj. načte data pomocí oznámení **nmGetData**. Metoda **DoSelect** je předefinována tak, že v případě navoleného druhého sloupce vyvolá editaci položky (metodu **DoEdit** – viz. kapitola 4.14.2.11).

4.14.1. Definice položek (řádků)

Jednotlivé řádky zobrazované komponentou **TListEdit** jsou popsány pomocí spojového seznamu struktur **TListEditItem** (viz. kapitola 4.14.2.3). Struktura **TListEditItem** obsahuje následující položky:

Položka	Význam
Next	Odkaz na následující strukturu ve spojovém seznamu
LeftText	Text zobrazený v prvním sloupci řádku.
RightText	Text zobrazený ve třetím sloupci řádku.
Flags	Kombinace příznaků s prefixem lif_ upřesňující vlastnosti řádku, viz. kapitola 4.14.2.2:
lifNotify	Místo editace se vyvolá oznámení nmNotify .
lifInlineEdit	Při editaci je vložena do místa položky komponenta TEdit .
lifInlineEnum	Při editaci je hodnota položky zvýšena o 1 (vhodné pro výčtové typy). Tento typ editace předpokládá, že položka má přiřazen validátor typu TOrdinalValidátor nebo jeho potomka.
lifReadOnly	Položku nelze editovat, je pouze zobrazována.

	Příznaky <code>lifNotify</code> , <code>lifInlineEdit</code> a <code>lifLineEnum</code> nelze mezi sebou kombinovat.
<code>HelpCtx</code>	Položka obsahující index do tabulky textů nápovědy. Jedná se o konstantu s prefixem <code>hc_</code> .
<code>Validator</code>	Validátor přiřazený řádku. Může se jednat o instanci libovolného validátoru. Položka <code>Flags</code> upřesňuje typ editace (viz. kapitola 4.14.2.2).

K vytvoření spojového seznamu struktur `TListEditItem` slouží funkce `NewListEditItem` nebo `NewListEditItemEx` (viz. kapitoly 4.14.2.4 a 4.14.2.5).

```
function NewListEditItem( ALeftText, ARightText: TRString;
  AFlags: Word; AValidator: PValidator;
  ANext: PListEditItem ): PListEditItem;
```

Příklad: Definování třech sloupců a čtyřech řádků:

```
P := New( PListEdit, Init( R, nil,
  NewListColumn( 'Description', 0, 138,
  NewListColumn( 'Value', lcfRight, 45,
  NewListColumn( '', lcfRight, 40,
  nil ))) ,
  NewListEditItem( 'Max RPM', 'rpm', lifInlineEdit,
    New( PRealValidator, Init( @A1, 0, 0, 2000, 2, 1 ) ) ),
  NewListEditItem( 'Gear ratio', '-', lifInlineEdit,
    New( PRealValidator, Init( @A2, 0, 0, 1000, 2, 1 ) ) ),
  NewListEditItem( 'Starting ramp', 's', lifInlineEdit,
    New( PRealValidator, Init( @A3, 0, 0, 1000, 2, 1 ) ) ),
  NewListEditItem( 'Stopping ramp', 's', lifInlineEdit,
    New( PRealValidator, Init( @A4, 0, 0, 1000, 2, 1 ) ) ),
  nil ))))
));
```

4.14.2. Implicitní nastavení komponenty TListEdit

V následující tabulce jsou uvedena implicitní nastavení všech položek komponenty. Všechny ostatní položky, které nejsou v tabulce uvedené, jsou inicializovány na hodnotu 0, `nil`, apod.

Položka	Hodnota
<code>Bounds</code>	parametr <code>ABounds</code> konstrukturu
<code>EventMask</code>	<code>\$FFFF</code>
<code>CaretSize</code>	(1, 8)
<code>State</code>	<code>sfVisible</code>
<code>Options</code>	<code>ofSelectable</code> or <code>ofSharedPalette</code> or <code>ofFirstClick</code>
<code>Font</code>	<code>fidDefault</code>
<code>Palette</code>	<code>#\$0F#\$00#\$00#\$00#\$00#\$0F#\$00#\$00#\$0F#\$00#\$0F</code> (sdílená paleta)
<code>Flags</code>	<code>lvfBorder</code> or <code>lvfHeader</code> or <code>lvfUseEnter</code>
<code>Columns</code>	podle parametru <code>AColumns</code> konstrukturu
<code>TopOffset</code>	1
<code>BottomOffset</code>	1

HeaderHeight	0
ItemHeight	0
ScrollBar	podle parametru AScrollBar konstrukturu
Selected	0
SelColumn	0
ItemCount	0

Pro komponentu **TListEdit** jsou definovány konstanty nastavení třídy uvedené v následující tabulce. Tyto konstanty lze předat jako parametr metodě **Customize** (viz. manuál ke knihovně Controls).

Konstanta	Význam
ccListEdit	Implicitní nastavení definované ovladačem použitého terminálu.

4.14.2.1. Reference

```

PListEdit = ^TListEdit;
TListEdit = object( TListView )
public
  Items : PListEditItem;

  constructor Init( const ABounds: TRect; AScrollBar: PScrollBar;
    AColumns: PListColumn; AEditItems: PListEditItem );
  destructor Done; virtual;

  function GetData( AItemIndex, AColIndex: Integer ):
    PRString; virtual;

  procedure DoSelect; virtual;
  procedure DoEdit( R: TRect; AItem: PListEditItem ); virtual;

  function GetItemByIndex( AItemIndex: Integer ): PListEditItem;
  function GetItemCount: Integer;
end;

```

4.14.2.2. Konstanty lif_

Konstanty s prefixem lif_ upřesňují vlastnosti řádku zobrazeného komponentou **TListView**. Kombinace konstant je uložena v položce Flags struktury **TListEditItem** (viz. kapitola 4.14.2.3).

Identifikátor	Kód	Popis
lifEditType	\$0007	Maska pro příznaky typu editace.
lifNotify	\$0000	Místo editace se vyvolá oznámení nmNotify.
lifInlineEdit	\$0001	Při editaci je vložena do místa položky komponenta TEdit .
lifInlineEnum	\$0002	Při editaci je hodnota položky zvýšena o 1 (vhodné pro výčtové typy). Tento typ editace předpokládá, že položka má přiřazen validátor typu TOrdinalValidátor nebo jeho potomka.
lifReadOnly	\$0008	Položku nelze editovat, je pouze zobrazována.

4.14.2.3. Struktura TListEditItem

Struktura **TListEditItem** popisuje jeden řádek zobrazený komponentou **TListView**. Struktury **TListEditItem** jsou alokovány na zásobníku a tvoří spojový seznam.

```
PListEditItem = ^TListEditItem;
TListEditItem = record
  Next      : PListEditItem;
  LeftText  : PRString;
  RightText : PRString;
  Validator : PValidator;
  Flags     : Word;
  HelpCtx   : Word;
end;
```

Popis jednotlivých položek struktury je uveden v následující tabulce:

Položka	Popis
Next	Odkaz na následující strukturu TListEditItem ve spojovém seznamu. Poslední struktura obsahuje v této položce hodnotu nil .
LeftText	Text zobrazený v prvním sloupci řádku.
RightText	Text zobrazený ve třetím sloupci řádku.
Validator	Validátor přiřazený řádku. Může se jednat o instanci libovolného validátoru. Položka Flags upřesňuje typ editace (viz. kapitola 4.14.2.2).
Flags	Kombinace příznaků s prefixem lif_ upřesňující vlastnosti řádku, viz. kapitola 4.14.2.2
HelpCtx	Položka obsahující index do tabulky textů nápovědy. Jedná se o konstantu s prefixem hc_ .

4.14.2.4. Funkce NewListEditItem

Funkce **NewListEditItem** slouží k alokování struktury **TListEditItem** popisující jeden řádek komponenty **TListEdit**.

```
function NewListEditItem( ALeftText, ARightText: TRString;
  AFlags: Word; AValidator: PValidator;
  ANext: PListEditItem ): PListEditItem;
```

Parametry:

ALeftText	Text zobrazovaný v prvním sloupci.
ARightText	Text zobrazovaný ve třetím sloupci.
AFlags	Kombinace příznaků s prefixem lif_ upřesňující vlastnosti řádku, viz. kapitola 4.14.2.2
AValidator	Validátor přiřazený řádku. Může se jednat o instanci libovolného validátoru. Položka Flags upřesňuje typ editace (viz. kapitola 4.14.2.2).
ANext	Odkaz na následující strukturu TListEditItem nebo hodnota nil .

Návratové hodnoty:

Funkce vrací ukazatel na strukturu **TEditItem** alokovanou na hromadě a inicializovanou dle parametrů této funkce.

Poznámky:

Položka HelpCtx struktury **TListEditItem** je touto funkcí nastavena na implicitní hodnotu nula.

Strukturu **TListEditItem**, resp. celý spojový seznam těchto struktur lze uvolnit pomocí procedure **FreeListEditItem**(viz. kapitola 4.13.9.5)

4.14.2.5. Funkce NewListEditItemEx

Funkce **NewListEditItemEx** slouží k alokování struktury **TListEditItem** popisující jeden řádek komponenty **TListEdit**.

```
function NewListEditItemEx( ALeftText, ARightText: TRString;
  AFlags: Word; AHelpCtx: Word; AValidator: PValidator;
  ANext: PListEditItem ): PListEditItem;
```

Parametry:

ALeftText	Text zobrazovaný v prvním sloupci.
ARightText	Text zobrazovaný ve třetím sloupci.
AFlags	Kombinace příznaků s prefixem lif_ upřesňující vlastnosti řádku, viz. kapitola 4.14.2.2
AHelpCtx	Položka obsahující index do tabulky textů nápovědy. Jedná se o konstantu s prefixem hc_.
AValidator	Validátor přiřazený řádku. Může se jednat o instanci libovolného validátoru. Položka Flags upřesňuje typ editace (viz. kapitola 4.14.2.2).
ANext	Odkaz na následující strukturu TListEditItem nebo hodnota nil .

Návratové hodnoty:

Funkce vrací ukazatel na strukturu **TEditItem** alokovanou na hromadě a inicializovanou dle parametrů této funkce.

Poznámky:

Položka HelpCtx struktury **TListEditItem** je touto funkcí nastavena na implicitní hodnotu nula.

Strukturu **TListEditItem**, resp. celý spojový seznam těchto struktur lze uvolnit pomocí procedure **FreeListEditItem**(viz. kapitola 4.13.9.5)

4.14.2.6. Procedura FreeListEditItem

Procedura **FreeListEditItem** slouží k uvolnění spojového seznamu struktur **TListEditItem** alokovaných metodou **NewListEditItem**.

```
procedure FreeListEditItem( AItem: PListEditItem );
```

Parametry:

AItem Odkaz na první strukturu **TListEditItem** ve spojovém seznamu..

Poznámky:

4.14.2.7. Položka TListEdit.Items

Položka **Items** obsahuje ukazatel na první položku spojového seznamu struktur **TListEditItem** popisujícího jednotlivé editované položky. Položka je inicializovaná parametrem konstrukturu a je určena pouze pro čtení.

```
Item : PListEditItem;
```

4.14.2.8. Konstruktor TListEdit.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```
constructor Init( const ABounds: TRect; AScrollBar: PScrollBar;  
    AColumns: PListColumn; AEditItems: PListEditItem );
```

Parametry:

ABounds Umístění a rozměry komponenty v rámci jejího vlastníka.
AScrollBar Odkaz na přidružený posuvník, tj. komponentu **TScrollBar**.
 Parametr může mít hodnotu **nil**.
AColumns Odkaz na první položku spojového seznamu struktur **TListColumn** vytvořených pomocí funkce **NewListColumn** (viz. kapitola 4.13.9.4)
AEditItems Odkaz na první položku spojového seznamu struktur **TListEditItem** vytvořených pomocí funkce **NewListEditItem** nebo **NewListEditItemEx** (viz. kapitola 4.14.1).

Návratové hodnoty:

Konstruktor nevrací žádnou hodnotu.

Poznámky:

Konstruktor **Init** nastaví všechny položky instance na implicitní hodnoty, viz. tabulka

v kapitole 4.13.3.

Spojové seznamy `AColumns` a `AEditItems` jsou automaticky uvolněny voláním destrukturu.

4.14.2.9. Metoda `TListEdit.GetData`

Metoda **GetData** vrací text položky se zadaným indexem řádku a sloupce. Metoda je volána automaticky pokaždé, když komponenta potřebuje vykreslit konkrétní položku.

```
function GetData( AItemIndex, AColIndex:
    Integer ): PString; virtual;
```

Parametry:

<code>AItemIndex</code>	Index položky (řádku). Hodnota parametru musí být větší nebo rovna nule a menší, než je celkový počet položek.
<code>AColIndex</code>	Index sloupce. Hodnota parametru musí být větší nebo rovna nule a menší, než je celkový počet sloupců.

Návratové hodnoty:

Metoda vrací ukazatel na znakový řetězec nebo hodnotu **nil**, pokud má být položka prázdná.

Poznámky:

Metoda **GetData** je definována jako virtuální. Implicitní chování metody **GetData** definované ve třídě **TListView** je následující:

Metoda vyvolá oznámení `nmGetData` s nastaveným parametrem `Row` a `Col`, odpovídající parametrům této metody. Obsluha oznámení musí vyplnit položku `Text` oznámení a metoda **GetData** vrátí obsah této položky.

Třída **TListEdit** předefinovává implicitní chování této metody následovně:

Pokud je číslo řádku v rozsahu 0 až počet editovaných položek mínus jedna a zároveň číslo sloupce je v rozsahu 0 až 2. Pak je text položky definován strukturou **TListEditItem**. Do prvního sloupce je vložena hodnota `LeftText`, do třetího sloupce hodnota `RightText` a text prostředního sloupce je naplněn textovou reprezentací hodnoty dané položkou `Validator`.

Pokud je číslo řádku nebo sloupce mimo rozsah uvedený výše, pak se metoda **GetData** chová jako ve třídě **TListView**, tj. vyvolává oznámení `nmGetData`.

4.14.2.10. Metoda `TListEdit.DoSelect`

Metoda **DoSelect** je automaticky volána dvojkliku nebo při stisku klávesy `vkEnter`, pokud je v položce `Flags` nastaven příznak `lvfUseEnter`.

```
procedure DoSelect: PString; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Metoda **DoSelect** je definována jako virtuální. Potomci třídy **TListView** ji mohou předefinovat.

Přímo ve třídě **TListEdit** je metoda **DoSelect** implementována tak, že zavolá oznámení **nmSelect**. Pokud je vybraná položka a sloupec editovaná hodnota s přiřazeným validátorem, pak metoda namísto vyvolání oznámení **nmSelect**, zavolá metodu **DoEdit**.

4.14.2.11. Metoda TListEdit.DoEdit

Metoda **DoEdit** provede na základě příznaků **lif_** uvedených v položce **Flags** struktury **TListEditItem** editaci pomocí validátoru položky.

```
procedure DoEdit( R: TRect; AItem: PListEditItem ); virtual;
```

Parametry:

R	Obdélník v souřadnicích vlastníka komponenty TListEdit , který určuje pozici a rozměry buňky pro zobrazení položky.
AItem	Odkaz na editovanou položku.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Potomci mohou tuto metodu předefinovat a doplnit další způsoby editace.

4.14.2.12. Metoda TListEdit.GetItemByIndex

Metoda **GetItemByIndex** vrací ukazatel na položku ve spojovém seznamu struktur **TListEditItem** podle zadaného indexu.

```
function GetItemByIndex( AItemIndex: Integer ): PListEditItem;
```

Parametry:

AItemIndex	Index položky. Položky jsou číslovány od nuly.
------------	--

Návratové hodnoty:

Metoda vrací ukazatel na strukturu **TListEditItem**. Pokud je index mimo rozsah seznamu, metoda vrací hodnotu **nil**.

Poznámky:**4.14.2.13. Metoda TListEdit.GetItemCount**

Metoda **GetItemCount** vrací počet položek ve spojovém seznamu struktur **TListEditItem**.

```
function GetItemCount: Integer;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda vrací počet položek.

Poznámky:

Počet položek nemusí nutně odpovídat počtu zobrazovaných řádků. Počet zobrazovaných řádků (**ItemCount**), lze nezávisle nastavit pomocí metody **SetItemCount**. Data řádků s indexem větším než je počet položek jsou získávány pomocí oznámení **nmGetData** (jako u komponenty **TListView**)

4.15. Komponenta TNavigator

Komponenta **TNavigator** je lišta umístěna obvykle ve spodní části displeje, na které je umístěno jedno nebo více funkčních tlačítek (panely). Obvykle tato lišta slouží k přepínání (navigaci) mezi stránkami aplikace. Na následujících obrázcích jsou zobrazeny ukázky této komponenty



U komponenty **TNavigator** lze nastavovat několik parametrů, které ovlivňují její chování, příp. vykreslování. Lze nastavit:

- Zobrazení rámečku kolem komponenty

U jednotlivých panelů (tlačítek navigátoru) lze nastavit tyto parametry:

- Vypisovaný text nebo vykreslovaná bitmapa
- Zarovnání textu doleva, doprava nebo na střed.
- Zobrazení oddělovací linky za panelem
- Přiřazenou klávesu (tj. kód kbXXX)
- Akci, která se má provést při kliknutí panelu, nebo stisku přiřazené klávesy

Při kliknutí panelu, nebo při stisku přiřazené klávesy panelu komponenta **TNavigator** může vyvolat následující akce:

- Generování oznámení `nmClick`
- Skok na definovanou stránku komponenty `TPageControl`
- Volání definované stránky komponenty `TPageControl`
- Návrat do předchozí stránky komponenty `TPageControl`

4.15.1. Položky a metody třídy TNavigator

Třída **TNavigator** má následující konstruktor:

```
constructor Init( const ABounds: TRect; APages: TPageControl;
  APanels: TNavPanel );
```

Popis parametrů konstruktoru je uveden v následující tabulce:

ABounds	Umístění a rozměry komponenty v rámci vlastníka
APanels	Odkaz na spojový seznam struktur <code>TNavPanel</code> (viz. kapitola 4.15.2) popisujících zobrazené panely. Položka je inicializována parametrem konstruktoru a je určena pouze pro čtení.
APages	V případě, že některý z panelů provádí při stisku přepnutí stránky, pak je potřeba aby tento parametr obsahoval odkaz na příslušnou komponentu TPageControl . Jinak může být nastaven na hodnotu nil .

Komponenta **TNavigator** rozšiřuje základní třídu **TControl** o položky uvedené v následující tabulce:

Flags	Položka upřesňující chování komponenty. Obsahuje kombinaci následujících příznaků: <table border="1" data-bbox="459 1713 1353 1758"> <tr> <td><code>nafBorder</code></td> <td>Zobrazení rámečku kolem komponenty</td> </tr> </table> Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetFlags .	<code>nafBorder</code>	Zobrazení rámečku kolem komponenty
<code>nafBorder</code>	Zobrazení rámečku kolem komponenty		
Panels	Odkaz na spojový seznam struktur <code>TNavPanel</code> (viz. kapitola 4.15.2) popisujících zobrazené panely. Položka je inicializována parametrem konstruktoru a je určena pouze pro čtení.		
Pages	V případě, že některý z panelů provádí při stisku přepnutí stránky, pak je potřeba aby tato položka obsahovala odkaz na příslušnou		

	komponentu TPageControl . Položka je inicializována parametrem konstrukturu a je určena pouze pro čtení.
--	---

Komponenta **TNavigator** rozšiřuje báзовou třídu **TControl** o metodu uvedené v následující tabulce:

SetFlags	Metoda pro nastavení položky Flags.
-----------------	-------------------------------------

4.15.2. Vytvoření seznamu panelů komponenty TNavigator

Panely zobrazované komponentou **TNavigator** jsou popsány pomocí spojového seznamu struktur **TNavPanel** (viz. kapitola 4.15.8.3). Struktura **TNavPanel** obsahuje následující položky:

Položka	Význam																						
Next	Odkaz na následující strukturu ve spojovém seznamu																						
Flags	Příznaky upřesňující vlastnosti sloupce. Položka obsahuje kombinaci následujících příznaků (viz. kapitola 4.15.8.2): <table border="1"> <tr> <td>npfCall</td> <td>Při kliknutí na panel je provedeno volání metody CallPage komponenty TPageControl odkazovanou položkou Pages komponenty s parametrem Param.</td> </tr> <tr> <td>npfGoto</td> <td>Při kliknutí na panel je provedeno volání metody GotoPage komponenty TPageControl odkazovanou položkou Pages komponenty s parametrem Param.</td> </tr> <tr> <td>npfReturn</td> <td>Při kliknutí na panel je provedeno volání metody ReturnPage komponenty TPageControl odkazovanou položkou Pages komponenty.</td> </tr> <tr> <td>npfNext</td> <td>Při kliknutí na panel je provedeno volání metody GotoNextPage komponenty TPageControl odkazovanou položkou Pages komponenty.</td> </tr> <tr> <td>npfPrev</td> <td>Při kliknutí na panel je provedeno volání metody GotoPrevPage komponenty TPageControl odkazovanou položkou Pages komponenty.</td> </tr> <tr> <td>npfNotify</td> <td>Při kliknutí na panel je generování oznámení nmClick.</td> </tr> <tr> <td>npfBitmap</td> <td>Místo textu je v panelu zobrazena bitmapa specifikovaná položkou Bitmap.</td> </tr> <tr> <td>npfDivider</td> <td>Zobrazení oddělovací linky za panelem</td> </tr> <tr> <td>npfAlignLeft</td> <td>Zarovnání textu nebo bitmapy doleva</td> </tr> <tr> <td>npfAlignRight</td> <td>Zarovnání textu nebo bitmapy doprava</td> </tr> <tr> <td>npfHidden</td> <td>Panel je skrytý, není vykreslován a nelze na něj kliknout</td> </tr> </table>	npfCall	Při kliknutí na panel je provedeno volání metody CallPage komponenty TPageControl odkazovanou položkou Pages komponenty s parametrem Param.	npfGoto	Při kliknutí na panel je provedeno volání metody GotoPage komponenty TPageControl odkazovanou položkou Pages komponenty s parametrem Param.	npfReturn	Při kliknutí na panel je provedeno volání metody ReturnPage komponenty TPageControl odkazovanou položkou Pages komponenty.	npfNext	Při kliknutí na panel je provedeno volání metody GotoNextPage komponenty TPageControl odkazovanou položkou Pages komponenty.	npfPrev	Při kliknutí na panel je provedeno volání metody GotoPrevPage komponenty TPageControl odkazovanou položkou Pages komponenty.	npfNotify	Při kliknutí na panel je generování oznámení nmClick.	npfBitmap	Místo textu je v panelu zobrazena bitmapa specifikovaná položkou Bitmap.	npfDivider	Zobrazení oddělovací linky za panelem	npfAlignLeft	Zarovnání textu nebo bitmapy doleva	npfAlignRight	Zarovnání textu nebo bitmapy doprava	npfHidden	Panel je skrytý, není vykreslován a nelze na něj kliknout
npfCall	Při kliknutí na panel je provedeno volání metody CallPage komponenty TPageControl odkazovanou položkou Pages komponenty s parametrem Param.																						
npfGoto	Při kliknutí na panel je provedeno volání metody GotoPage komponenty TPageControl odkazovanou položkou Pages komponenty s parametrem Param.																						
npfReturn	Při kliknutí na panel je provedeno volání metody ReturnPage komponenty TPageControl odkazovanou položkou Pages komponenty.																						
npfNext	Při kliknutí na panel je provedeno volání metody GotoNextPage komponenty TPageControl odkazovanou položkou Pages komponenty.																						
npfPrev	Při kliknutí na panel je provedeno volání metody GotoPrevPage komponenty TPageControl odkazovanou položkou Pages komponenty.																						
npfNotify	Při kliknutí na panel je generování oznámení nmClick.																						
npfBitmap	Místo textu je v panelu zobrazena bitmapa specifikovaná položkou Bitmap.																						
npfDivider	Zobrazení oddělovací linky za panelem																						
npfAlignLeft	Zarovnání textu nebo bitmapy doleva																						
npfAlignRight	Zarovnání textu nebo bitmapy doprava																						
npfHidden	Panel je skrytý, není vykreslován a nelze na něj kliknout																						
KeyCode	Kód klávesy přiřazený panelu (kb_). Při stisku této klávesy se provede akce spojená s panelem.																						
Param	Doplňující parametr akce panelu. Obsahuje buď identifikátor stránky (pro akce npfGoto, npfCall) nebo kód, který se pošle v oznámení																						

	nmClick (akce npfNotify).
Width	Šířka panelu v pixelech. Pokud je hodnota menší nebo rovna nule, pak je šířka sloupce vypočítána z šířky textu příp. bitmapy, ke které je přičtena absolutní hodnota této položky.
Bitmap	Ukazatel na bitmapu vykreslanou v panelu. Položka je platná, pokud je v položce Flags nastaven příznak npfBitmap.
Text	Ukazatel na text zobrazovaný v panelu. Položka je platná, pokud v položce Flags není nastaven příznak npfBitmap.

K vytvoření spojového seznamu panelů slouží následující funkce:

Funkce **NewTextPanel** (viz. kapitola 4.15.8.4) alokuje na hromadě strukturu TNavPanel pro panel zobrazující text.

```
function NewTextPanel( const AText: string;
  AKeyCode: Word; AWidth: Integer; AFlags: Word; AParam: Word;
  ANext: PNavPanel ): PNavPanel;
```

Funkce **NewBitmapPanel** (viz. kapitola 4.15.8.5) alokuje na hromadě strukturu TNavPanel pro panel zobrazující bitmapu.

```
function NewBitmapPanel( ABitmap: PBitmap; AKeyCode: Word;
  AWidth: Integer; AFlags: Word; AParam: Word;
  ANext: PNavPanel) : PNavPanel;
```

Příklad: Vytvoření komponenty **TNavigator** se dvěma panely.

```
Navigator := New( PNavigator, Init( R, PageCtrl,
  NewTextPanel( 'Zpět', kbEsc, -4, npfReturn or npfDivider, 0,
  NewTextPanel( 'Setup', Ord( 'S' ), -4,
    npfCallPage or npfDivider, cidSetupPage,
  nil )) ));
```

4.15.3. Implicitní nastavení komponenty TNavigator

V následující tabulce jsou uvedena implicitní nastavení všech položek komponenty. Všechny ostatní položky, které nejsou v tabulce uvedené, jsou inicializovány na hodnotu 0, **nil**, apod.

Položka	Hodnota
Bounds	parametr ABounds konstrukturu
EventMask	\$FFFF
CaretSize	(1, 8)
State	sfVisible
Options	ofSharedPalette or ofPostProcess
Font	fidDefault
Palette	#\$00#\$0F#\$0F#\$0F (sdílená paleta)
Panels	podle parametru APanels konstrukturu
Pages	pdole parametru APages konstrukturu

Pro komponentu **TNavigator** jsou definovány konstanty nastavení třídy uvedené v následující tabulce. Tyto konstanty lze předat jako parametr metodě **Customize** (viz. manuál ke knihovně Controls).

Konstanta	Význam
ccNavigator	Implicitní nastavení definované ovladačem použitého terminálu.

4.15.4. Události obsluhované komponentou TNavigator

Komponenta **TNavigator** obsluhuje vstupní události podle následující tabulky:

Událost	Popis činnosti
evMouseDown	Vyvolání akce spojené s příslušným panelem.
evKeyDown	Klávesy nadefinované ve spojovém seznamu struktur TNavPanel Vyvolání akce spojené s příslušným panelem.

4.15.5. Oznámení generovaná komponentou TNavigator

Komponenta generuje následující oznámení:

Oznámení	Okamžik vyvolání	Parametry oznámení
nmHide	Volání metody Hide	-
nmShow	Volání metody Show	-
nmEnable	Volání metody Enable	-
nmDisable	Volání metody Disable	-
nmClick	Kliknutí do oblasti panelu s nastaveným příznakem npfNotify	Položka ItemId struktury oznámení obsahuje hodnoty položky Param struktury TNavPanel. Položka LParam obsahuje odkaz na strukturu TNavPanel.

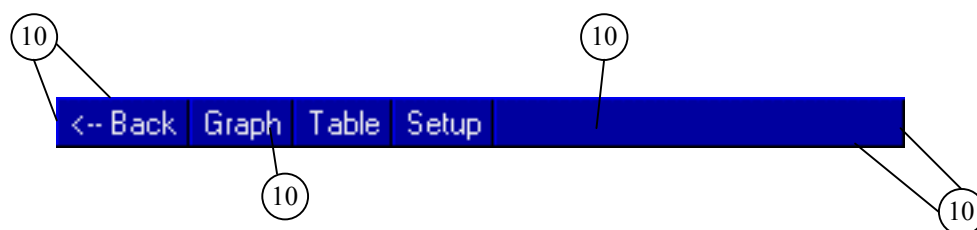
4.15.6. Validace hodnoty komponenty TNavigator

S komponentu **TNavigator** nelze použít žádný typ validátoru.

4.15.7. Paleta komponenty TNavigator

Konstruktor komponenty **TNavigator** nastavuje implicitně paletu barev komponenty podle následující tabulky:

#	Barva	Popis
1	\$00	Pozadí
2	■ \$0F	Linka rámečku (vlevo a nahoře)
3	■ \$0F	Linka rámečku (vpravo a dole)
4	■ \$0F	Text



4.15.8. Reference

Třída **TNavigator** je potomkem třídy **TControl** popsané v dokumentaci ke knihovně Controls.

```

PNavigator = ^TNavigator;
TNavigator = object( TControl )
public
  Panels : PNavPanel;
  Pages  : PPageControl;
  Flags  : Word;

  constructor Init( const ABounds: TRect;
    APages: PPageControl; APanels: PNavPanel );
  destructor Done; virtual;
  procedure Paint( ACanvas: PCanvas ); virtual;
  procedure HandleEvent( var AEvent: TEvent ); virtual;
  function PanelAtPos( AMouse: TPoint ): PNavPanel; virtual;
  procedure ClickPanel( APanel: PNavPanel ); virtual;
  procedure SetFlags( ASet, AReset: Word ); virtual;
  function FindPanel( AParam: Word ): PNavPanel;
end;

```

4.15.8.1. Konstanty naf_

Konstanty s prefixem naf_ upřesňují chování komponenty **TNavigator**. Kombinace konstant je uložena v položce Flags této komponenty (viz. kapitola 4.15.8.1).

Identifikátor	Kód	Popis
nafBorder	\$0001	Zobrazení rámečku kolem komponenty

4.15.8.2. Konstanty npf_

Konstanty s prefixem npf_ upřesňují vlastnosti panelu komponenty **TNavigator**. Kombinace konstant je uložena v položce Flags struktury TNavPanel (viz. kapitola 4.15.8.3).

Identifikátor	Kód	Popis
npfActionMask	\$000F	Maska pro příznaky akce při kliknutí na panel
npfCall	\$0000	Při kliknutí na panel je provedeno volání metody CallPage komponenty TPageControl odkazovanou položkou Pages komponenty s parametrem Param.
npfGoto	\$0001	Při kliknutí na panel je provedeno volání metody GotoPage komponenty TPageControl odkazovanou položkou Pages

		komponenty s parametrem Param.
npfReturn	\$0002	Při kliknutí na panel je provedeno volání metody ReturnPage komponenty TPageControl odkazovanou položkou Pages komponenty.
npfNext	\$0003	Při kliknutí na panel je provedeno volání metody GotoNextPage komponenty TPageControl odkazovanou položkou Pages komponenty.
npfPrev	\$0004	Při kliknutí na panel je provedeno volání metody GotoPrevPage komponenty TPageControl odkazovanou položkou Pages komponenty.
npfNotify	\$0005	Při kliknutí na panel je generování oznámení nmClick.
npfBitmap	\$0010	Místo textu je v panelu zobrazena bitmapa specifikovaná položkou Bitmap.
npfDivider	\$0020	Zobrazení oddělovací linky za panelem
npfAlignLeft	\$0040	Zarovnání textu nebo bitmapy doleva
npfAlignRight	\$0080	Zarovnání textu nebo bitmapy doprava
npfHidden	\$0100	Panel je skrytý, není vykreslován a nelze na něj kliknout (při stisku přiřazené klávesy je provedena požadovaná akce)
npfDisabled	\$0200	Panel je zakázaný, je viditelný, ale nefunkční.

Příznaky npfCall až npfNotify nelze mezi sebou kombinovat. Pokud není uveden ani jeden z příznak npfAlignLeft, npfAlignRight, pak je text panelu zarovnán na střed.

4.15.8.3. Struktura TNavPanel

Struktura **TNavPanel** popisuje vlastnosti jednoho panelu komponenty **TNavigator**. Struktury **TNavPanel** tvoří spojový seznam alokovaný na hromadě.

```

PNavPanel = ^TNavPanel;
TNavPanel = record
  RefCount : Integer;
  Next     : PNavPanel;
  Flags    : Word;
  KeyCode  : Word;
  Width    : Integer;
  Param    : Word;
  case Integer of
    0: (Text : PString);
    1: (Bitmap : PBitmap);
  end;

```

Význam jednotlivých položek struktury **TNavPanel** je uveden v následující tabulce:

Položka	Význam
RefCount	Položka obsahuje počet referencí (odkazů) na tuto strukturu. Položka umožňuje, aby struktura panelu mohla být sdílená mezi různými komponentami TNavigator. Položka je nastavována automaticky.
Next	Odkaz na následující strukturu ve spojovém seznamu
Flags	Příznaky upřesňující vlastnosti sloupce. Položka obsahuje kombinaci příznaků s prefixem npf_ (viz. kapitola 4.15.8.2)
KeyCode	Kód klávesy přiřazený panelu (konstanta s prefixem kb_ nebo ASCII kód). Při stisku této klávesy se provede akce spojená s panelem.

Param	Doplňující parametr akce panelu. Obsahuje buď identifikátor stránky (pro akce npfGoto, npfCall) nebo kód, který se pošle v oznámení nmClick (akce npfNotify).
Width	Šířka panelu v pixelech. Pokud je hodnota menší nebo rovna nule, pak je šířka sloupce vypočítána z šířky textu příp. bitmapy, ke které je přičtena absolutní hodnota této položky.
Bitmap	Ukazatel na bitmapu vykreslanou v panelu. Položka je platná, pokud je v položce Flags nastaven příznak npfBitmap.
Text	Ukazatel na text zobrazovaný v panelu. Položka je platná, pokud v položce Flags není nastaven příznak npfBitmap.

4.15.8.4. Funkce NewTextPanel

Funkce **NewTextPanel** slouží k alokování struktury **TNavPanel** popisující jeden panel komponenty **TNavigator** obsahující text.

```
function NewTextPanel( const AText: string; AKeyCode: Word;
  AWidth: Integer; AFlags: Word; AParam: Word;
  ANext: PNavPanel ): PNavPanel;
```

Parametry:

AText	Zobrazovaný text v panelu.
AKeyCode	Kód klávesy (tj. konstanta kb_ nebo ASCII kód). Při stisku této klávesy se provede akce svázaná s panelem.
AWidth	Šířka sloupce v pixelech. Pokud je hodnota parametru menší nebo rovna nule, pak je šířka sloupce vypočítána z šířky textu, ke které je přičtena absolutní hodnota tohoto parametru.
AFlags	Kombinace příznaků s prefixem npf_ upřesňující vlastnosti sloupce a akci prováděnou při stisku klávesy dané parametrem AKeyCode nebo při kliknutí na panel (viz. kapitola 4.15.8.2)
AParam	Parametr upřesňující akci svázanou s panelem (viz. kapitola 4.15.8.2)
ANext	Odkaz na následující strukturu TNavPanel nebo hodnota nil

Návratové hodnoty:

Funkce vrací ukazatel na strukturu **TNavPanel** alokovanou na hromadě a inicializovanou dle parametrů této funkce.

Poznámky:

Strukturu **TNavPanel**, resp. celý spojový seznam těchto struktur lze uvolnit pomocí procedure **FreePanel** (viz. kapitola 4.15.8.6)

4.15.8.5. Funkce NewBitmapPanel

Funkce **NewBitmapPanel** slouží k alokování struktury **TNavPanel** popisující jeden panel komponenty **TNavigator** obsahující bitmapu.

```
function NewBitmapPanel( ABitmap: PBitmap; AKeyCode: Word;
```



```
AWidth: Integer; AFlags: Word; AParam: Word;  
ANext: PNavPanel) : PNavPanel;
```

Parametry:

ABitmap	Bitmapa zobrazovaná v panelu.
AKeyCode	Kód klávesy (tj. konstanta kb_ nebo ASCII kód). Při stisku této klávesy se provede akce svázaná s panelem.
AWidth	Šířka sloupce v pixelech. Pokud je hodnota parametru menší nebo rovna nule, pak je šířka sloupce vypočítána z šířky textu, ke které je přičtena absolutní hodnota tohoto parametru.
AFlags	Kombinace příznaků s prefixem npf_ upřesňující vlastnosti sloupce a akci prováděnou při stisku klávesy dané parametrem AKeyCode nebo při kliknutí na panel (viz. kapitola 4.15.8.2)
AParam	Parametr upřesňující akci svázanou s panelem (viz. kapitola 4.15.8.2)
ANext	Odkaz na následující strukturu TNavPanel nebo hodnota nil

Návratové hodnoty:

Funkce vrací ukazatel na strukturu **TNavPanel** alokovanou na hromadě a inicializovanou dle parametrů této funkce.

Poznámky:

Strukturu **TNavPanel**, resp. celý spojový seznam těchto struktur lze uvolnit pomocí procedure **FreePanel** (viz. kapitola 4.15.8.6)

4.15.8.6. Procedure FreePanel

Procedure **FreePanel** slouží k uvolnění spojového seznamu struktur **TNavPanel** alokovaných funkcemi **NewTextPanel** nebo **NewBitmapPanel**.

```
procedure FreePanel( APanel: PNavPanel );
```

Parametry:

APanel Odkaz na první strukturu **TNavPanel** ve spojovém seznamu..

Poznámky:

4.15.8.7. Položka TNavigator.Panels

Položka **Panels** obsahuje odkaz na spojový seznam struktur **TNavPanel** (viz. kapitola 4.15.2) popisujících zobrazené panely. Položka je inicializována parametrem konstruktoru a je určena pouze pro čtení.

```
Panels : PNavPanel;
```

4.15.8.8. Položka TNavigator.Pages

Položka **Pages** obsahuje odkaz na instanci komponenty **TPageControl**. V případě, že některý z panelů provádí při stisku přepnutí stránky, je volána příslušná metoda této komponenty. V případě, že žádný z panelů neprovádí přepnutí stránky, může být tato položka nastavena na hodnotu **nil**. Položka je inicializována parametrem konstruktoru a je určena pouze pro čtení.

```
Pages : TPageControl;
```

4.15.8.9. Položka TNavigator.Flags

Položka **Flags** upřesňuje chování komponenty. Obsahuje kombinaci příznaků s prefixem `naf_` popsaných v kapitole 4.15.8.1. Položka je určena pouze pro čtení. Pro její nastavení slouží metodu **SetFlags** (viz. kapitola 4.15.8.14).

```
Flags : Word;
```

4.15.8.10. Konstruktore TNavigator.Init

Konstruktore **Init** provádí inicializaci instance třídy.

```
constructor Init( const ABounds: TRect; APages: TPageControl;
  APanels: TNavPanel );
```

Parametry:

ABounds	Umístění a rozměry komponenty v rámci jejího vlastníka.
APages	V případě, že některý z panelů provádí při stisku přepnutí stránky, pak je potřeba aby tento parametr obsahoval odkaz na příslušnou komponentu TPageControl . Pokud ani jeden panel neprovádí přepnutí stránky, pak může mít hodnotu nil .
APanels	Odkaz na první položku spojového seznamu struktur TNavPanel vytvořených pomocí funkce NewTextPanel nebo NewBitmapPanel (viz. kapitola 4.15.8.4 nebo 4.15.8.5)

Návratové hodnoty:

Konstruktore nevrací žádnou hodnotu.

Poznámky:

Konstruktore **Init** nastaví všechny položky instance na implicitní hodnoty, viz. tabulka v kapitole 4.15.3.

4.15.8.11. Destruktore TNavigator.Done

Destruktore **Done** provádí uvolnění prostředku alokovaných konstruktorem instance..

```
destructor Done; virtual;
```

Parametry:

Destruktor nemá žádné parametry.

Návratové hodnoty:

Destruktor nevrací žádnou hodnotu.

Poznámky:

Destruktor **Done** uvolní spojový seznam panelů alokovaný na hromadě.

4.15.8.12. Metoda TNavigator.PanelAtPos

Metoda **PanelAtPos** vrací ukazatel na panel, tj. strukturu TNavPanel, který obsahuje zadané souřadnice.

```
function PanelAtPos( AMouse: TPoint ): PNavPanel; virtual;
```

Parametry:

AMouse Parametr obsahuje lokální souřadnice bodu.

Návratové hodnoty:

Metoda vrací ukazatel na panel, tj. strukturu TNavPanel. Pokud zadané souřadnice se nenacházejí v žádném z panelů, metoda vrací hodnotu **nil**.

Poznámky:

4.15.8.13. Metoda TNavigator.ClickPanel

Metoda **ClickPanel** provádí akci svázanou s panelem.

```
procedure ClickPanel( APanel: PNavPanel ); virtual;
```

Parametry:

APanel Ukazatel na strukturu panelu, tj. strukturu **TNavPanel**.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Metoda **ClickPanel** je automaticky volána při stisku klávesy přiřazené panelu nebo při kliknutí do oblasti panelu. Potomci třídy mohou tuto metodu předefinovat.

4.15.8.14. Metoda TNavigator.SetFlags

Metoda **SetFlags** slouží k nastavení položky **Flags** (viz. kapitola 4.15.8.9)

```
procedure SetFlags( ASet, AReset: Word ); virtual;
```

Parametry:

AFlags Kombinace příznaků s prefixem **naf_** (viz. kapitola 4.15.8.1).

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

4.15.8.15. Metoda TNavigator.FindPanel

Metoda **FindPanel** slouží k vyhledání struktury panelu podle hodnoty její položky **Param**.

```
function FindPanel( AParam: Word ): PNavPanel;
```

Parametry:

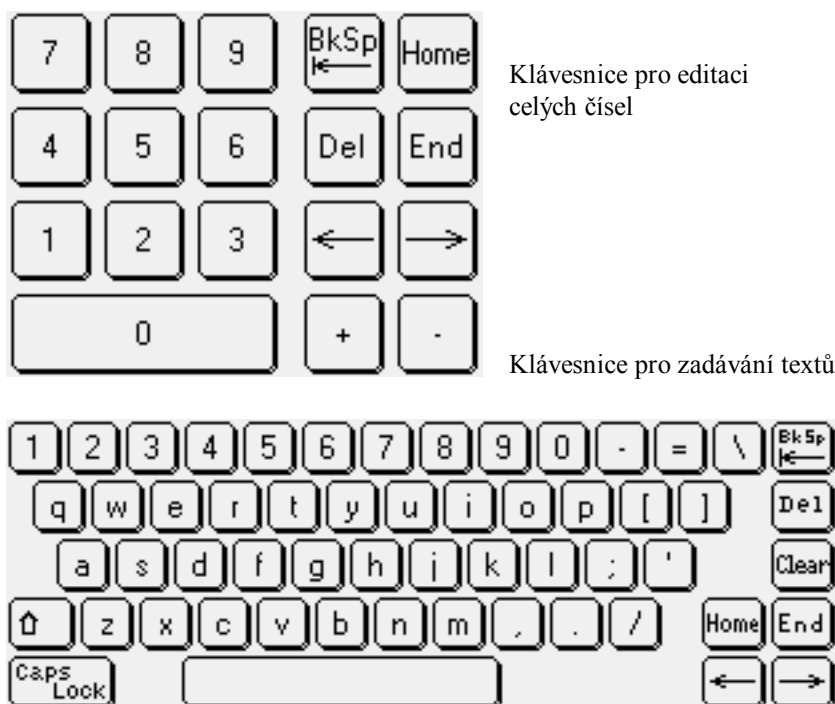
AParam Hodnota parametru **AParam** zadaná při vytvoření panelu funkce **NewTextPanel** nebo **NewBitmapPanel** (viz. kapitoly 4.15.8.4, 4.15.8.5).

Návratové hodnoty:

Funkce vrací ukazatel na nalezenou strukturu **TNavPanel**. V případě, že parametr **AParam** neodpovídá položce **Param** ani jedné struktury v seznamu, metoda vrací hodnotu **nil**.

4.16. Komponenta TKeypad

Komponenta **TKeypad** implementuje virtuální klávesnici pro terminály s dotykovým panelem. Na následujících obrázcích jsou zobrazeny ukázky této komponenty.



Komponenta **TKeypad** je implementována univerzálně. Komponentě **TKeypad** je nutné předhodit definici klávesnice, která je umístěna mimo tuto komponentu, obvykle ve speciální knihovně poplatné konkrétnímu typu terminálu (T11mkpad, T11ckpad, T33mkpad apod.). Jsou definovány následující typy klávesnic:

- Klávesnice pro zadávání anglických textů
- Klávesnice pro zadávání českých textů
- Klávesnice pro zadávání textů s přepínačem Anglická/Česká
- Klávesnice pro zadávání hexadecimálních čísel
- Klávesnice pro zadávání decimálních celých čísel
- Klávesnice pro zadávání desetinných čísel
- Klávesnice pro zadávání desetinných čísel (bez znaku E)
- Klávesnice pro zadávání speciálních hodnot, jako je datum, čas apod.
- Klávesnice se dvěma tlačítky Ok a Storno

U komponenty **TKeypad** lze nastavovat několik parametrů, které ovlivňují její chování, příp. vykreslování. Lze nastavit:

- Zobrazení rámečku kolem komponenty
- Rozvržení kláves primární a sekundární klávesnice

Součástí každé klávesnice mohou být dvě speciální tlačítka Shift a Alt (příp. tlačítka ShiftLock a AltLock). Na základě stavu těchto dvou tlačítek může mít každé další tlačítko až čtyři různé funkce. Dále klávesnice může obsahovat tlačítko LangLock, tj. tlačítko primárně určené pro přepínání mezi dvěma jazykovými mutacemi klávesnice.

U každého tlačítka klávesnice lze nezávisle nastavit tyto vlastnosti:

- Umístění a rozměry v rámci komponenty

- Bitmapa stisknutého a uvolněného tlačítka
- Zda tlačítko reaguje na stisk nebo na uvolnění
- Akce při stisku tlačítka

Při stisku tlačítka se mohou provést tyto akce:

- Emulace stisku zadané klávesy
- Vyvolání oznámení `nmClick`
- Ukončení modálního stavu

4.16.1. Položky a metody třídy `TKeyPad`

Třída **`TKeyPad`** má následující konstruktor:

```
constructor Init( const ABounds: TRect );
```

Popis parametrů konstruktoru je uveden v následující tabulce:

ABounds	Umístění komponenty v rámci vlastníka. Rozměry klávesnice jsou automaticky dopočítány při vložení komponenty do skupiny vlastníka.
----------------	--

Komponenta **`TKeyPad`** rozšiřuje báзовou třídu **`TControl`** o položky uvedené v následující tabulce:

Flags	Položka upřesňující chování komponenty. Obsahuje kombinaci příznaků <code>kpf</code> (viz. kapitola 4.16.8.1): <table border="1" data-bbox="454 1220 1348 1265"> <tr> <td><code>kpfBorder</code></td> <td>Zobrazení rámečku kolem komponenty</td> </tr> </table> Položka je určena pouze pro čtení. Pro její nastavení použijte metodu <code>SetFlags</code> .	<code>kpfBorder</code>	Zobrazení rámečku kolem komponenty
<code>kpfBorder</code>	Zobrazení rámečku kolem komponenty		
PriButtons	Definice primární klávesnice. Odkaz na spojový seznam struktur <code>TKeyPadButton</code> (viz. kapitola 4.16.8.7). Položka je určena pouze pro čtení. Pro její nastavení použijte metodu <code>SetButtons</code> .		
SecButtons	Definice sekundární klávesnice. Odkaz na spojový seznam struktur <code>TKeyPadButton</code> (viz. kapitola 4.16.8.7). Položka je určena pouze pro čtení. Pro její nastavení použijte metodu <code>SetButtons</code> .		
BitmapGetter	Odkaz na funkci vracející ukazatel na bitmapy jednotlivých tlačítek.		
AltState	Aktuální stav kláves <code>Shift</code> , <code>Alt</code> a <code>LangLock</code> . Viz. příznaky <code>kpa_</code> v kapitole 4.16.8.3. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu <code>SetAltState</code> .		

Komponenta **`TKeyPad`** rozšiřuje báзовou třídu **`TControl`** o metodu uvedené v následující tabulce:

SetButtons	Metoda pro nastavení rozvržení kláves primární a sekundární klávesnice.
SetFlags	Metoda pro nastavení položky <code>Flags</code> .
SetAltState	Metoda pro nastavení položky <code>AltState</code> .

4.16.2. Definice rozvržení kláves komponenty TKeypad

Komponenta **TKeypad** může najednou spravovat dvě různá rozvržení kláves nazvaná primární rozvržení a sekundární rozvržení. Přepínání mezi těmito rozvrženími zajišťuje speciální klávesa LangLock (viz. dále)

Rozvržení klávesnice je spojový seznam struktur **TKeypadButton**. Každá struktura **TKeypadButton** obsahuje následující položky:

Položka	Význam																				
Next	Odkaz na další strukturu ve spojovém seznamu																				
Flags	<p>Položka upřesňuje chování tlačítka. Obsahuje kombinaci následujících příznaků:</p> <table border="1"> <tbody> <tr> <td>kpbfKeyCode</td> <td>Při stisku tlačítka se emuluje stisk klávesy dané položkami KeyCode, KeyCodeSh, KeyCodeAlt nebo KeyCodeShAlt podle aktuálního stavu kláves Shift a Alt.</td> </tr> <tr> <td>kpbfNotify</td> <td>Při stisku se generuje oznámení nmClick.</td> </tr> <tr> <td>kpbfEndModal</td> <td>Při stisku tlačítka se provede ukončení modálního stavu metodou EndModal. Důvod ukončení modálního stavu je dán položkami KeyCode, KeyCodeSh, KeyCodeAlt nebo KeyCodeShAlt podle aktuálního stavu kláves Shift a Alt.</td> </tr> <tr> <td>kpbfShift</td> <td>Tlačítko funguje jako klávesa Shift.</td> </tr> <tr> <td>kpbfShiftLock</td> <td>Tlačítko funguje jako klávesa Shift-Lock (Caps-Lock)</td> </tr> <tr> <td>kpbfAlt</td> <td>Tlačítko funguje jako klávesa Alt.</td> </tr> <tr> <td>kpbfAltLock</td> <td>Tlačítko funguje jako klávesa Alt-Lock.</td> </tr> <tr> <td>kpbfLangLock</td> <td>Tlačítko funguje jako přepínač mezi primární a sekundární klávesnicí.</td> </tr> <tr> <td>kpbfPush</td> <td>Pokud je nastaven tento příznak, tlačítko reaguje na stisk. V opačném případě reaguje na uvolnění.</td> </tr> <tr> <td>kpbfChar</td> <td>Pokud je nastaven tento příznak, pak je uvnitř tlačítka vykreslen znak daný položkami KeyCode, KeyCodeSh, KeyCodeAlt nebo KeyCodeShAlt, podle aktuálního stavu kláves Shift a Alt.</td> </tr> </tbody> </table> <p>Příznaky kpbfKeyCode až kpbfLangLock uvedené v tabulce výše nelze mezi sebou kombinovat.</p>	kpbfKeyCode	Při stisku tlačítka se emuluje stisk klávesy dané položkami KeyCode, KeyCodeSh, KeyCodeAlt nebo KeyCodeShAlt podle aktuálního stavu kláves Shift a Alt.	kpbfNotify	Při stisku se generuje oznámení nmClick.	kpbfEndModal	Při stisku tlačítka se provede ukončení modálního stavu metodou EndModal. Důvod ukončení modálního stavu je dán položkami KeyCode, KeyCodeSh, KeyCodeAlt nebo KeyCodeShAlt podle aktuálního stavu kláves Shift a Alt.	kpbfShift	Tlačítko funguje jako klávesa Shift.	kpbfShiftLock	Tlačítko funguje jako klávesa Shift-Lock (Caps-Lock)	kpbfAlt	Tlačítko funguje jako klávesa Alt.	kpbfAltLock	Tlačítko funguje jako klávesa Alt-Lock.	kpbfLangLock	Tlačítko funguje jako přepínač mezi primární a sekundární klávesnicí.	kpbfPush	Pokud je nastaven tento příznak, tlačítko reaguje na stisk. V opačném případě reaguje na uvolnění.	kpbfChar	Pokud je nastaven tento příznak, pak je uvnitř tlačítka vykreslen znak daný položkami KeyCode, KeyCodeSh, KeyCodeAlt nebo KeyCodeShAlt, podle aktuálního stavu kláves Shift a Alt.
kpbfKeyCode	Při stisku tlačítka se emuluje stisk klávesy dané položkami KeyCode, KeyCodeSh, KeyCodeAlt nebo KeyCodeShAlt podle aktuálního stavu kláves Shift a Alt.																				
kpbfNotify	Při stisku se generuje oznámení nmClick.																				
kpbfEndModal	Při stisku tlačítka se provede ukončení modálního stavu metodou EndModal. Důvod ukončení modálního stavu je dán položkami KeyCode, KeyCodeSh, KeyCodeAlt nebo KeyCodeShAlt podle aktuálního stavu kláves Shift a Alt.																				
kpbfShift	Tlačítko funguje jako klávesa Shift.																				
kpbfShiftLock	Tlačítko funguje jako klávesa Shift-Lock (Caps-Lock)																				
kpbfAlt	Tlačítko funguje jako klávesa Alt.																				
kpbfAltLock	Tlačítko funguje jako klávesa Alt-Lock.																				
kpbfLangLock	Tlačítko funguje jako přepínač mezi primární a sekundární klávesnicí.																				
kpbfPush	Pokud je nastaven tento příznak, tlačítko reaguje na stisk. V opačném případě reaguje na uvolnění.																				
kpbfChar	Pokud je nastaven tento příznak, pak je uvnitř tlačítka vykreslen znak daný položkami KeyCode, KeyCodeSh, KeyCodeAlt nebo KeyCodeShAlt, podle aktuálního stavu kláves Shift a Alt.																				
Bounds	Umístění a rozměry tlačítka v rámci komponenty TKeypad.																				
UpBitmap	Identifikátor bitmapy uvolněného tlačítka																				
DownBitmap	Identifikátor bitmapy stisknutého tlačítka																				
KeyCode	Kód klávesy kb_ emulované při stisku tlačítka																				
KeyCodeSh	Kód klávesy kb_ emulované při stisku tlačítka a současném držení klávesy Shift																				

KeyCodeAlt	Kód klávesy kb_ emulované při stisku tlačítka a současném držení klávesy Alt
KeyCodeShAlt	Kód klávesy kb_ emulované při stisku tlačítka a současném držení kláves Shift a Alt

K definici kláves slouží následující funkce:

Funkce **NewKeypadButton** (viz. kapitola 4.16.8.4) alokuje a inicializuje na hromadě strukturu TKeypadButton.

```
function NewKeypadButton( ALeft, ATop, ARight, ABottom: Integer;
  AFlags: Word; AUpBitmap, ADownBitmap: Integer;
  AKeyCode, AKeyCodeSh, AKeyCodeAlt, AKeyCodeShAlt: Word;
  ANext: PKeypadButton ): PKeypadButton;
```

Funkce **NewStdKeypadButton** (viz. kapitola 4.16.8.5) je jednodušší alternativa k funkci NewKeypadButton. Nastavuje položky Bounds, UpBitmap, KeyCode. Položku Flags nastavuje na hodnotu kpbfPush. Ostatní položky nechává nulové.

```
function NewStdKeypadButton( ALeft, ATop, ARight, ABottom: Integer;
  ABitmap: Integer; AKeyCode: Integer;
  ANext: PKeypadButton ): PKeypadButton;
```

4.16.3. Implicitní nastavení komponenty TKeypad

V následující tabulce jsou uvedena implicitní nastavení všech položek komponenty. Všechny ostatní položky, které nejsou v tabulce uvedené, jsou inicializovány na hodnotu 0, **nil**, apod.

Položka	Hodnota
Bounds	parametr ABounds konstrukturu
EventMask	evMouseDown or evMouseRep or evMouseMove or evMouseDown or evMouseUp
CaretSize	(1, 8)
State	sfVisible
Options	ofSharedPalette
Font	fidDefault
Palette	#\$0F#\$00#\$00#\$00 (sdílená paleta)
AltState	0
Flags	0

Pro komponentu **TKeypad** jsou definovány konstanty nastavení třídy uvedené v následující tabulce. Tyto konstanty lze předat jako parametr metodě **Customize** (viz. manuál ke knihovně Controls).

Konstanta	Význam
ccKeypadAlphaEn	Klávesnice pro zadávání anglických textů
ccKeypadAlphaCz	Klávesnice pro zadávání českých textů
ccKeypadAlphaEnCz	Klávesnice pro zadávání textů s přepínačem

	Anglická/Česká
ccKeypadNumHex	Klávesnice pro zadávání hexadecimálních čísel
ccKeypadNumDec	Klávesnice pro zadávání decimálních celých čísel
ccKeypadNumReal	Klávesnice pro zadávání desetinných čísel
ccKeypadNumSmallReal	Klávesnice pro zadávání desetinných čísel (bez znaku E)
ccKeypadNumSpec	Klávesnice pro
ccKeypadOkCancel	Klávesnice se dvěma tlačítky Ok a Storno

4.16.4. Události obsluhované komponentou TKeypad

Komponenta **TKeypad** obsluhuje vstupní události podle následující tabulky

Událost	Popis činnosti
evMouseDown evMouseRep evMouseDown	Kliknutí do oblasti některého z tlačítek vyvolá akci s ním spojenou, tj. buď emulaci klávesy, ukončení modálního stavu nebo vyvolání oznámení nmClick.

4.16.5. Oznámení generovaná komponentou TKeypad

Komponenta generuje následující oznámení:

Oznámení	Okamžik vyvolání	Parametry oznámení
nmHide	Volání metody Hide	-
nmShow	Volání metody Show	-
nmEnable	Volání metody Enable	-
nmDisable	Volání metody Disable	-
nmClick	Kliknutí na tlačítko označené příznakem kpbfNotify	Parametr WParam struktury oznámení obsahuje hodnotu položku KeyCode, KeyCodeSh, KeyCodeAlt nebo KeyCodeShAlt podle aktuálního stavu tlačítek Shift a Alt.

4.16.6. Validace hodnoty komponenty TKeypad

S komponentu **TKeypad** nelze použít žádný typ validátoru.

4.16.7. Paleta komponenty TKeypad

Konstruktor komponenty **TKeypad** nastavuje implicitně paletu barev komponenty podle následující tabulky:

#	Barva	Popis
1	\$0F	Pozadí
2	■ \$00	Linka rámečku (vlevo a nahoře)

3 ■ \$00 Linka rámečku (vpravo a dole)

4 ■ \$00 Text tlačítek

4.16.8. Reference

Třída **TKeypad** je potomkem třídy **TControl** popsané v dokumentaci ke knihovně Controls.

```

PKeypad = ^TKeypad;
TKeypad = object( TControl )
public
  PriButtons      : PKeypadButton;
  SecButtons      : PKeypadButton;
  Flags           : Word;
  BitmapGetter   : TBitmapGetter;
  AltState       : Word;

  constructor Init( const ABounds: TRect );
  destructor Done; virtual;
  procedure HandleEvent( var AEvent: TEvent ); virtual;
  procedure Paint( ACanvas: PCanvas ); virtual;
  procedure SizeLimits( var AMin, AMax: TPoint ); virtual;
  procedure PaintButton( AButton: PKeypadButton; ADown: Boolean );
  procedure ClickButton( AButton: PKeypadButton ); virtual;
  function GetButtonBitmap( AId: Integer ): PBitmap; virtual;
  procedure SetButtons( APriButtons, ASecButtons: PKeypadButton );
  procedure SetFlags( ASet, AReset: Word ); virtual;
  procedure SetAltState( AAltState: Word );
  function IsButtonDown( AButton: PKeypadButton ): Boolean;
  function GetButtonKeyCode( AButton: PKeypadButton ): Word;
  function GetButtons: PKeypadButton;
end;

```

4.16.8.1. Konstanty kpf_

Konstanty s prefixem **kpf_** upřesňují chování komponenty **TKeypad**. Kombinace konstant je uložena v položce **Flags** této komponenty (viz. kapitola 4.16.8.10).

Identifikátor	Kód	Popis
kpfBorder	\$0001	Zobrazení rámečku kolem komponenty

4.16.8.2. Konstanty kpbf_

Konstanty s prefixem **kpbf_** upřesňují vlastnosti tlačítka klávesnice **TKeypad**. Kombinace konstant je uložena v položce **Flags** struktury **TKeypadButton** popisující tlačítko klávesnice (viz. kapitola 4.16.8.7).

Identifikátor	Kód	Popis
kpbfActionMask	\$000F	Maska příznaků akce tlačítka
kpbfKeyCode	\$0000	Při stisku tlačítka se emuluje stisk klávesy dané položkami KeyCode , KeyCodeSh , KeyCodeAlt nebo KeyCodeShAlt podle aktuálního stavu kláves Shift a Alt .
kpbfNotify	\$0001	Při stisku se generuje oznámení nmClick .

kpbfEndModal	\$0002	Při stisku tlačítka se provede ukončení modálního stavu metodou EndModal. Důvod ukončení modálního stavu je dán položkami KeyCode, KeyCodeSh, KeyCodeAlt nebo KeyCodeShAlt podle aktuálního stavu kláves Shift a Alt.
kpbfShift	\$0003	Tlačítko funguje jako klávesa Shift.
kpbfShiftLock	\$0004	Tlačítko funguje jako klávesa Shift-Lock (Caps-Lock)
kpbfAlt	\$0005	Tlačítko funguje jako klávesa Alt.
kpbfAltLock	\$0006	Tlačítko funguje jako klávesa Alt-Lock.
kpbfLangLock	\$0007	Tlačítko funguje jako přepínač mezi primární a sekundární klávesnicí.
kpbfPush	\$0010	Pokud je nastaven tento příznak, tlačítko reaguje na stisk. V opačném případě reaguje na uvolnění.
kpbfChar	\$0020	Pokud je nastaven tento příznak, pak je uvnitř tlačítka vykreslen znak daný položkami KeyCode, KeyCodeSh, KeyCodeAlt nebo KeyCodeShAlt, podle aktuálního stavu kláves Shift a Alt.

Příznaky akce kpbfKeyCode až kbfLangLock nelze mezi sebou kombinovat, vždy může být uveden pouze jeden z těchto příznaků.

4.16.8.3. Konstanty kpa_

Konstanty s prefixem kpa_ definují klávesy určené k přepínání funkcí ostatních kláves, tj. klávesy Shift, Shift-Lock, Alt, Alt-Lock a Lang-Lock. Kombinace těchto konstant je uložena v položce AltState komponenty TKeypad, která udžuje stav těchto speciálních kláves (viz. kapitola 4.16.8.12).

Identifikátor	Kód	Popis
kpaShift	\$01	Tlačítko Shift
kpaShiftLock	\$02	Tlačítko Shift-Lock
kpaAlt	\$04	Tlačítko Alt
kpaAltLock	\$08	Tlačítko Alt-Lock
kpaLangLock	\$10	Tlačítko Lang-Lock

4.16.8.4. Funkce NewKeypadButton

Funkce **NewKeypadButton** slouží k alokování struktury **TKeypadButton** popisující jedno tlačítko komponenty **TKeypad**.

```
function NewKeypadButton( ALeft, ATop, ARight, ABottom: Integer;
  AFlags: Word; AUpBitmap, ADownBitmap: Integer;
  AKeyCode, AKeyCodeSh, AKeyCodeAlt, AKeyCodeShAlt: Word;
  ANext: PKeypadButton ): PKeypadButton;
```

Parametry:

ALeft	Souřadnice X levého horního rohu tlačítka.
ATop	Souřadnice Y levého horního rohu tlačítka.
ARight	Souřadnice X pravého horního rohu tlačítka.

ABottom	Souřadnice Y pravého horního rohu tlačítka.
AFlags	Kombinace příznaků s prefixem kpbf_ upřesňující vlastnosti tlačítka a akci prováděnou při stisku tlačítka (viz. kapitola 4.16.8.2)
AUpBitmap	Číselný identifikátor podkladové bitmapy uvolněného tlačítka.
ADownBitmap	Číselný identifikátor podkladové bitmapy stisknutého tlačítka.
AKeyCode	Kód emulované klávesy při uvolnění klávesách Shift a Alt.
AKeyCodeSh	Kód emulované klávesy při stisknutí klávese Shift a uvolnění klávese Alt.
AKeyCodeAlt	Kód emulované klávesy při uvolnění klávese Shift a stisknutí klávese Alt.
AKeyCodeShAlt	Kód emulované klávesy při stisknutí klávesách Shift a Alt.
ANext	Odkaz na následující strukturu TKeypadButton nebo hodnota nil .

Návratové hodnoty:

Funkce vrací ukazatel na inicializovanou strukturu **TKeypadButton** alokovanou na hromadě.

Poznámky:

V závislosti na akci tlačítka specifikované parametrem AFlags mohou parametry AKeyCode, AKeyCodeSh, AKeyCodeAlt a AKeyCodeShAlt obsahovat různé hodnoty.

Akce	Význam parametrů AKeyCode, AKeyCodeSh, ...
kpbfKeyCode	Kód klávesy, tj. konstanta s prefixem kb_ nebo případně ASCII kód znaku.
kpbfNotify	Identifikátor posílaný ve struktuře oznámení v položce WParam.
kpbfEndModal	Důvod ukončení modální stavu, tj. konstant s prefixem mr_, např. mrOk nebo mrCancel.
ostatní	Parametry jsou nevýznamné

Strukturu **TKeypadButton**, resp. celý spojový seznam těchto struktur lze uvolnit pomocí procedure **FreeKeypadButton** (viz. kapitola 4.16.8.6)

4.16.8.5. Funkce NewStdKeypadButton

Funkce **NewStdKeypadButton** slouží k alokování struktury **TKeypadButton** popisující jedno tlačítko komponenty **TKeypad**. Jedná se o jednodušší alternativu k funkci **NewKeypadButton** (viz. kapitola 4.16.8.4).

```
function NewStdKeypadButton( ALeft, ATop, ARight, ABottom: Integer;
  ABitmap: Integer; AKeyCode: Integer;
  ANext: PKeypadButton ): PKeypadButton;
```

Parametry:

ALeft Souřadnice X levého horního rohu tlačítka.

A <code>Top</code>	Souřadnice Y levého horního rohu tlačítka.
A <code>Right</code>	Souřadnice X pravého horního rohu tlačítka.
A <code>Bottom</code>	Souřadnice Y pravého horního rohu tlačítka.
A <code>Bitmap</code>	Číselný identifikátor podkladové bitmapy tlačítka.
A <code>KeyCode</code>	Kód emulované klávesy při uvolnění klávesách Shift a Alt.
A <code>Next</code>	Odkaz na následující strukturu <code>TKeypadButton</code> nebo hodnota nil .

Návratové hodnoty:

Funkce vrací ukazatel na inicializovanou strukturu **TKeypadButton** alokovanou na hromadě.

Poznámky:

Funkce alokuje strukturu `TKeypadButton` a podle svých parametrů nastavuje položky `Bounds`, `UpBitmap`, `DownBitmap` a `KeyCode`. Položku `Flags` naplní hodnotou `kpbfKeycode` **or** `kpbfPush`. Ostatní položky ponechá nulové.

Strukturu **TKeypadButton**, resp. celý spojový seznam těchto struktur lze uvolnit pomocí procedure **FreeKeypadButton** (viz. kapitola 4.16.8.6)

4.16.8.6. Procedura `FreeKeypadButton`

Procedure **FreeKeypadButton** slouží k uvolnění spojového seznamu struktur **TKeypadButton** alokovaných funkcemi **NewKeypadButton** nebo **NewStdKeypadButton**.

```
procedure FreeKeypadButton( AButton: PKeypadButton );
```

Parametry:

A <code>Button</code>	Odkaz na první strukturu TKeypadButton ve spojovém seznamu..
-----------------------	---

Poznámky:4.16.8.7. Struktura `TKeypadButton`

Struktura **TKeypadButton** popisuje vlastnosti jednoho tlačítka klávesnice. Celý popis klávesnice komponenty **TKeypad** je tvořen spojovým seznamem struktur **TKeypadButton** alokovaných na hromadě.

```
PKeypadButton = ^TKeypadButton;
TKeypadButton = record
  RefCount      : Integer;
  Next          : PKeypadButton;
  Flags         : Word;
  Bounds        : TRect;
  UpBitmap      : Integer;
```

```

DownBitmap : Integer;

case Integer of
  0 : ( KeyCode      : Word;
        KeyCodeSh   : Word;
        KeyCodeAlt  : Word;
        KeyCodeShAlt : Word );
end;

```

Jednotlivé položky struktury **TKeypadButton** popisuje následující tabulka:

Položka	Význam
Next	Odkaz na další strukturu ve spojovém seznamu
Flags	Položka upřesňuje chování tlačítka. Obsahuje kombinaci příznaků kb_ (viz. kapitola 4.16.8.2).
Bounds	Umístění a rozměry tlačítka v rámci komponenty TKeypad .
UpBitmap	Identifikátor bitmapy uvolněného tlačítka
DownBitmap	Identifikátor bitmapy stisknutého tlačítka
KeyCode	Kód klávesy kb_ emulované při stisku tlačítka
KeyCodeSh	Kód klávesy kb_ emulované při stisku tlačítka a současném držení klávesy Shift
KeyCodeAlt	Kód klávesy kb_ emulované při stisku tlačítka a současném držení klávesy Alt
KeyCodeShAlt	Kód klávesy kb_ emulované při stisku tlačítka a současném držení kláves Shift a Alt

4.16.8.8. Položka TKeypad.PriButtons

Položka **PriButtons** obsahuje odkaz na spojový seznam struktur TKeypadButton popisujících primární klávesnici. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetButtons** (viz. kapitola 4.16.8.18).

```
PriButtons : PKeypadButton;
```

4.16.8.9. Položka TKeypad.SecButtons

Položka **SecButtons** obsahuje odkaz na spojový seznam struktur TKeypadButton popisujících primární klávesnici. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetButtons** (viz. kapitola 4.16.8.18).

4.16.8.10. Položka TKeypad.Flags

Položka **Flags** upřesňuje chování komponenty. Obsahuje kombinaci příznaků s prefixem kpf_ popsanych v kapitole 4.16.8.1. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetFlags** (viz. kapitola 4.16.8.19).

```
Flags : Word;
```

4.16.8.11. Položka TKeypad.BitmapGetter

Položka **BitmapGetter** obsahuje odkaz na funkci vracející ukazatel na bitmapu tlačítka na základě číselného identifikátoru. Tato funkce je volána vždy, když komponenta vykresluje určité tlačítko, parametr této funkce je buď hodnota UpBitmap

nebo `DownBitmap` (podle stavu tlačítka) struktury **TKeypadButton**.

```
BitmapGetter : TBitmapGetter;
```

4.16.8.12. Položka TKeypad.AltState

Položka **AltState** udržuje stav speciálních tlačítek Shift, Shift-Lock, Alt, Alt-Lock a Lang-Lock. Položka obsahuje kombinaci příznaků `kpa_` (viz. kapitola 4.16.8.3). Položka je určena pouze pro čtení. Pro její nastavení slouží metoda `SetAltState` (viz. kapitola 4.16.8.20)

```
AltState      : Word;
```

4.16.8.13. Konstruktor TKeypad.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```
constructor Init( const ABounds: TRect );
```

Parametry:

ABounds Umístění komponenty v rámci jejího vlastníka. Velikost komponenty je automaticky dopočítána z rozložení tlačítek a nastavena při vložení komponenty do skupiny vlastníka.

Návratové hodnoty:

Konstruktor nevrací žádnou hodnotu.

Poznámky:

Konstruktor **Init** nastaví všechny položky instance na implicitní hodnoty, viz. tabulka v kapitole 4.16.3.

4.16.8.14. Destruktor TKeypad.Done

Destruktor **Done** provádí uvolnění prostředku alokovaných konstruktorem instance..

```
destructor Done; virtual;
```

Parametry:

Destruktor nemá žádné parametry.

Návratové hodnoty:

Destruktor nevrací žádnou hodnotu.

Poznámky:

Destruktor **Done** uvolní spojový seznam struktur **TKeypadButton**, na který se

odkazují položky **PriButtons** a **SecButtons**.

4.16.8.15. Metoda TKeypad.PaintButton

Metoda **PaintButton** vykresluje jedno tlačítko v zadaném stavu, tj. stisknuté nebo uvolněné.

```
procedure PaintButton( AButton: PKeypadButton; ADown: Boolean );
```

Parametry:

AButton	Odkaz na strukturu TKeypadButton tlačítka klávesnice.
ADown	Pokud je parametr ADown nastaven na hodnotu True, tlačítko je vykresleno jako stisknuté. V opačném případě je vykresleno jako uvolněné.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

4.16.8.16. Metoda TKeypad.ClickButton

Metoda **ClickButton** provádí akci svázanou se stiskem tlačítka.

```
procedure ClickButton( AButton: PKeypadButton ); virtual;
```

Parametry:

AButton	Odkaz na strukturu TKeypadButton tlačítka klávesnice.
---------	--

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Metoda **ClickButton** je automaticky volána při stisku tlačítka klávesnice. Potomci třídy mohou tuto metodu předefinovat.

4.16.8.17. Metoda TKeypad.GetButtonBitmap

Metoda **GetButtonBitmap** ukazuje na strukturu podkladové bitmapy tlačítka.

```
function GetButtonBitmap( AId: Integer ): PBitmap; virtual;
```

Parametry:

AId	Číselný identifikátor bitmapy uvedený v položkách UpBitmap
-----	--

nebo DownBitmap struktury **TKeypadButton**.

Návratové hodnoty:

Metoda vrací ukazatel na strukturu bitmapy nebo hodnotu **nil**.

Poznámky:

Metoda **GetButtonBitmap** je automaticky volána při vykreslování tlačítek. Potomci třídy mohou tuto metodu předefinovat.

Přímo ve třídě **TKeypad** je tato metoda implementována tak, že volá funkce danou položkou **BitmapGetter** (viz. kapitola 4.16.8.11)

4.16.8.18. Metoda **TKeypad.SetButtons**

Metoda **SetButtons** slouží k nastavení primárního a sekundárního rozložení kláves , tj položek **PriButtons** a **SecButton** (viz. kapitoly 4.16.8.8 a 4.16.8.9).

```
procedure SetButtons( APriButtons, ASecButtons: PKeypadButton );
```

Parametry:

APriButtons	Odkaz na spojový seznam struktur TKeypadButton primární klávesnice.
ASecButtons	Odkaz na spojový seznam struktur TKeypadButton sekundární klávesnice.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

4.16.8.19. Metoda **TKeypad.SetFlags**

Metoda **SetFlags** slouží k nastavení položky **Flags** (viz. kapitola 4.16.8.10)

```
procedure SetFlags( ASet, AReset: Word ); virtual;
```

Parametry:

AFlags	Kombinace příznaků s prefixem kpf_ (viz. kapitola 4.16.8.1).
--------	---

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

4.16.8.20. Metoda TKeypad.SetAltState

Metoda **SetAltState** slouží k nastavení stavu speciálních kláves Shift, Shift-Lock, Alt, Alt-Lock a Lang-Lock, tj položky AltState, (viz. kapitola 4.16.8.12)

```
procedure SetAltState( AAltState: Byte ); virtual;
```

Parametry:

AAltState Kombinace příznaků s prefixem kpa_ (viz. kapitola 4.16.8.3).

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení tlačítek komponenty.

4.16.8.21. Metoda TKeypad.IsButtonDown

Metoda **IsButtonDown** zjišťuje zda je zadané tlačítko stisknuto.

```
function IsButtonDown( AButton: PKeypadButton ): Boolean;
```

Parametry:

AButton Odkaz na strukturu TKeypadButton popisující tlačítko.

Návratové hodnoty:

Metoda vrací aktuální stav tlačítka. Pokud je stisknuto vrací hodnotu True.

Poznámky:

4.16.8.22. Metoda TKeypad.GetButtonKeyCode

Metoda **GetButtonKeyCode** vrací kód klávesy emulované při stisku tlačítka zadaného tlačítka platný pro aktuální stav kláves Shift a Alt.

```
function GetButtonKeyCode( AButton: PKeyPadButton ): Word;
```

Parametry:

AButton Odkaz na strukturu **TKeypadButton** tlačítka.

Návratové hodnoty:

Metoda **GetButtonKeyCode** vrací jednu z položek `KeyCode`, `KeyCodeSh`, `KeyCodeAlt`, `KeyCodeShAlt` podle aktuálního stavu tlačítek Shift a Alt.

Poznámky:

4.16.8.23. Metoda `TKeypad.GetButtons`

Metoda **GetButtons** vrací odkaz na spojový seznam struktur **TKeypadButton** aktuálního rozložení tlačítek klávesnice.

```
function GetButtons: PkeypadButton;
```

Parametry:

Metoda nemá žádné parametry.

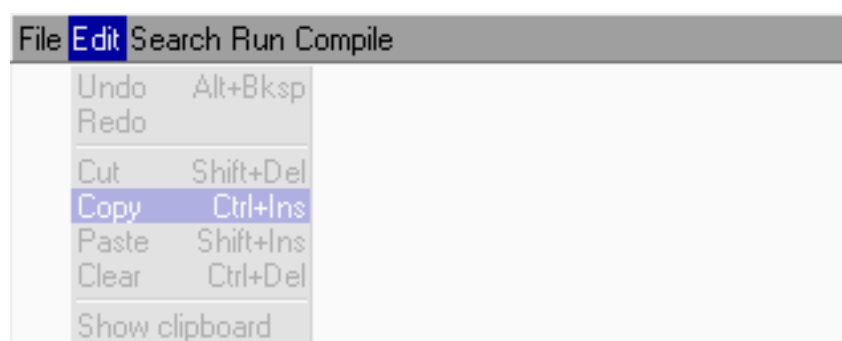
Návratové hodnoty:

Pokud je v položce `AltState` vynulován příznak `kpaLangLock`, tj. tlačítko Lang-Lock není stisknuto, vrací metoda hodnotu položky `PriButtons` (primární rozložení kláves). V opačném případě vrací hodnotu položky `SecButtons` (sekundární rozložení kláves)

Poznámky:

4.17. Komponenta `TMenuBar`

Komponenta **TMenuBar** je lišta s položkami kořenového menu. Tato lišta je obvykle umístěna v horní části displeje terminálu. Na následujícím obrázku je ukázka této komponenty.



Komponenta **TMenuBar** je určena pro terminály s dotykovým panelem, lze ji však s omezením použít i pro terminály vybavené pouze klávesnicí.

Při procházení menu dochází k dynamickému vytváření a rušení komponent **TMenuBar** (viz. kapitola 4.18) pro jednotlivá podmenu.

U komponenty **TMenuBar** lze nastavovat několik parametrů, které ovlivňují její

chování, příp. vykreslování. Lze nastavit:

- Zobrazení rámečku kolem komponenty

U jednotlivých položek zobrazených komponentou **TMenuBar** lze nastavit tyto vlastnosti:

- Text položky
- Šířku položky
- Akci, která se má provést při kliknutí panelu, nebo stisku přiřazené klávesy

Na položky kořenového menu zobrazené komponentou **TMenuBar** obvykle navazují podmenu. Toto pravidlo však není absolutní, tyto položky mohou mimo jiné vyvolat následující akce:

- Generování oznámení `nmClick`
- Skok na definovanou stránku komponenty **TPageControl**
- Volání definované stránky komponenty **TPageControl**
- Návrat do předchozí stránky komponenty **TPageControl**
- Ukončení modálního stavu

4.17.1. Položky a metody třídy TMenuBar

Třída **TMenuBar** má následující konstruktor:

```
constructor Init( const ABounds: TRect; AMenu: PMenu );
```

Popis parametrů konstruktoru je uveden v následující tabulce:

ABounds	Umístění a rozměry komponenty v rámci vlastníka
AMenu	Odkaz na strukturu stromového menu alokovanou na hromadě. Viz. kapitola 4.17.2.

Komponenta **TMenuBar** vychází ze třídy **TCustomMenu**, která rozšiřuje básovou třídu **TControl** mimo jiné o položky uvedené v následující tabulce (detailní popis těchto položek se nachází v dokumentaci ke knihovně Controls)

Flags	Položka upřesňuje chování komponenty. Obsahuje kombinaci následujících příznaků.	
	mfMenuBar	Příznak určující zda se jedná o komponentu TMenuBar . Pokud tento příznak není nastaven, pak se jedná o komponentu TMenuBox nebo jejího potomka. Tento příznak musí být vždy nastaven.
	mfBorder	Zobrazení rámečku kolem komponenty
	Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetFlags .	
Pages	V případě, že některá z položek menu provádí při výběru přepnutí	

	stránky, pak je potřeba aby tento parametr obsahoval odkaz na příslušnou komponentu TPageControl , ve které k přepnutí dojde. Jinak může být nastaven na hodnotu nil .
Menu	Položka obsahuje odkaz na první úroveň menu (tj. strukturu TMenu). Položka je inicializovaná parametrem konstruktoru a je určena pouze pro čtení.
Selected	Položka obsahuje odkaz na vybranou položku menu (tj. strukturu TMenuItem). Položka je nastavována automaticky při procházení menu a je určena pouze pro čtení.

Komponenta **TMenuPopup** rozšiřuje básovou třídu **TControl** o metodu uvedené v následující tabulce:

SetFlags	Metoda pro nastavení položky Flags .
-----------------	---

4.17.2. Vytvoření struktury menu

Stromové menu je tvořena pomocí struktur **TMenu** (podmenu) a **TMenuItem** (položka menu) alokovaných na hromadě. K vytvoření stromového menu slouží funkce **NewMenu**, **NewMenuItem**, **NewSubMenu**, které jsou detailně popsány v dokumentaci ke knihovně **Controls** v kapitole ke komponentě **TCustomMenu**.

4.17.3. Implicitní nastavení komponenty **TMenuBar**

V následující tabulce jsou uvedena implicitní nastavení všech položek komponenty. Všechny ostatní položky, které nejsou v tabulce uvedené, jsou inicializovány na hodnotu 0, **nil**, apod.

Položka	Hodnota
Bounds	parametr ABounds konstruktoru
EventMask	\$FFFF
CaretSize	(1, 8)
State	sfVisible
Options	ofSharedPalette or ofPostProcess
Font	fidDefault
Palette	#\$0F#\$00#\$00#\$00#\$00#\$0F (sdílená paleta)
Menu	podle parametru AMenu konstruktoru
Flags	mfMenuBar

Pro komponentu **TMenuBar** jsou definovány konstanty nastavení třídy uvedené v následující tabulce. Tyto konstanty lze předat jako parametr metodě **Customize** (viz. manuál ke knihovně **Controls**).

Konstanta	Význam
ccMenuBar	Implicitní nastavení definované ovladačem použitého terminálu.

4.17.4. Události obsluhované komponentou TMenuBar

Komponenta **TMenuBar** obsluhuje vstupní události dvojím způsobem, podle toho zda je nebo není v modálním stavu. Pokud komponenta **TMenuBar** není v modálním stavu obsluhuje události podle následující tabulky:

Událost		Popis činnosti
evMouseDown		Kliknutí do oblasti menu způsobí spuštění menu v modálním stavu.
evKeyDown	vkMenu	Stisk klávesy vkMenu způsobí spuštění menu v modálním stavu.

Pokud je komponenta v modálním stavu, tj. při procházení menu. Pak se vstupní události obsluhují jinak, jak je uvedeno v následující tabulce:

Událost		Popis činnosti
evMouseDown evMouseUp evMouseMove		Procházení menu. Akce svázaná s položkou menu je aktivována při uvolnění tlačítka myši.
evKeyDown	vkLeft	Výběr předchozí položky menu
	vkRight	Výběr následující položky menu
	vkDown	Rozbalení podmenu aktuální položky
	vkHome	Výběr první položky menu
	vkEnd	Výběr poslední položky menu
	vkEnter	Aktivace akce spojené s položkou menu
	vkEsc	Ukončení modálního stavu menu

4.17.5. Oznámení generovaná komponentou TMenuBar

Komponenta generuje následující oznámení:

Oznámení	Okamžik vyvolání	Parametry oznámení
nmEnter	Komponenta se dostane do ohniska	-
nmExit	Komponenta se dostane mimo ohnisko	-
nmHide	Volání metody Hide	-
nmShow	Volání metody Show	-
nmEnable	Volání metody Enable	-
nmDisable	Volání metody Disable	-
nmClick	Výběr položky menu označené příznakem mifNotify	Položka ItemId struktury oznámení obsahuje hodnotu položky Id struktury TMenuItem .

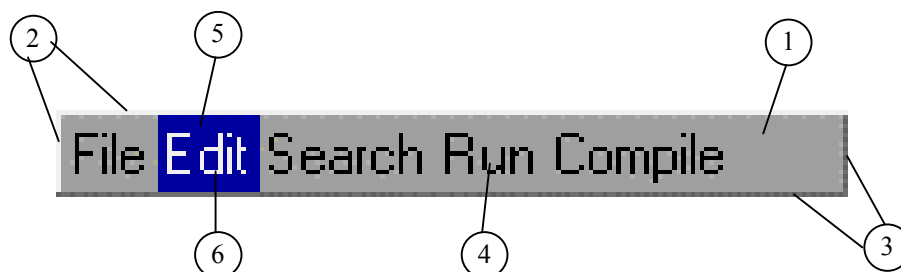
4.17.6. Validace hodnoty komponenty TMenuBar

S komponentu **TMenuBar** nelze použít žádný typ validátoru.

4.17.7. Paleta komponenty TMenuBar

Konstruktor komponenty **TMenuBar** nastavuje implicitně paletu barev komponenty podle následující tabulky:

#	Barva	Popis
1	\$0F	Pozadí
2	■ \$00	Linka rámečku (vlevo a nahoře)
3	■ \$00	Linka rámečku (vpravo a dole)
4	■ \$00	Text
5	■ \$00	Pozadí vybrané položky
6	\$0F	Text vybrané položky



4.17.8. Reference

Třída **TMenuBar** je potomkem třídy **TCustomMenu** popsané v dokumentaci ke knihovně Controls.

```

PMenuBar = ^TMenuBar;
TMenuBar = object( TCustomMenu )
public
  constructor Init( const ABounds: TRect; AMenu: PMenu );
  destructor Done; virtual;
  procedure Paint( ACanvas: PCanvas ); virtual;
  procedure PaintItems( AItem1, AItem2: PMenuItem ); virtual;
  procedure GetItemRect( AItem: PMenuItem;
    var AR: TRect ); virtual;
  function NewSubControl( const AItemBounds: TRect; AMenu: PMenu;
    AParentMenu: PCustomMenu ): PCustomMenu; virtual;

end;
```

4.17.8.1. Konstruktor TMenuBar.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```

constructor Init( const ABounds: TRect; AMenu: PMenu );
```

Parametry:

ABounds Umístění a rozměry komponenty v rámci jejího vlastníka.
AMenu Odkaz na první úroveň menu, tj. strukturu **TMenu** vytvořenou pomocí funkce **NewMenu** (viz. kapitola 4.17.2)

Návratové hodnoty:

Konstruktor nevrací žádnou hodnotu.

Poznámky:

Konstruktor **Init** nastaví všechny položky instance na implicitní hodnoty, viz. tabulka v kapitole 4.17.3.

4.17.8.2. Destruktor TMenuBar.Done

Destruktor **Done** provádí uvolnění prostředku alokovaných konstruktorem instance..

```
destructor Done; virtual;
```

Parametry:

Destruktor nemá žádné parametry.

Návratové hodnoty:

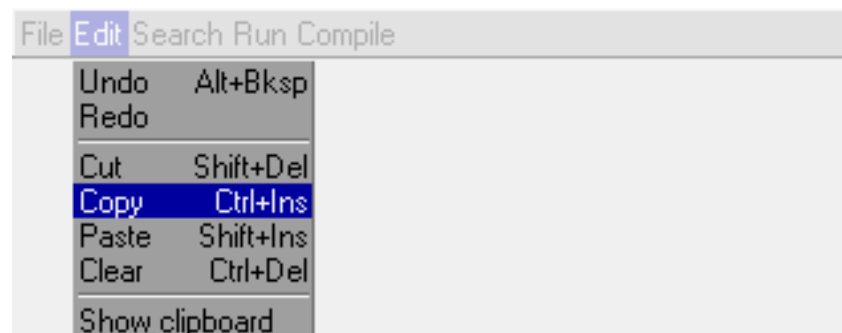
Destruktor nevrací žádnou hodnotu.

Poznámky:

Destruktor **Done** uvolní strukturu menu alokovanou na hromadě.

4.18. Komponenta TMenuBar

Komponenta **TMenuBar** je komponenta zobrazující položky menu v řádcích pod sebou, jak ukazuje následující obrázek:



Komponenta **TMenuBar** vychází ze třídy **TCustomMenu** (popsané v dokumentaci ke knihovně Controls) a dědí většinu jejich vlastností.

Aplikace nikdy nevytváří instance komponenty **TMenuBar** přímo. Komponenty

TMenuBar jsou dynamicky vytvářeny a rušeny při procházení menu vytvořeného pomocí komponenty **TMenuBar** nebo **TMenuPoup**.

Pro vytvoření menu celé aplikace použijte komponentu **TMenuBar** (viz. kapitola 4.17). Pro vyvolání samostatného menu, tzv. popup menu, použijte komponentu **TMenuPopup** (viz. kapitola 4.19)

4.18.1. Reference

Třída **TMenuBar** je potomkem třídy **TCustomMenu** popsané v dokumentaci ke knihovně Controls.

```

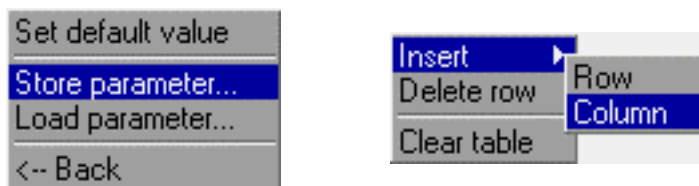
PMenuBar = ^TMenuBar;
TMenuBar = object( TCustomMenu )
public
    ItemHeight : Integer;

    constructor Init( const ABounds: TRect; AMenu: PMenu;
        AParentMenu: PCustomMenu );
    procedure Paint( ACanvas: PCanvas ); virtual;
    procedure PaintItems( AItem1, AItem2: PMenuItem ); virtual;
    procedure GetItemRect( AItem: PMenuItem;
        var AR: TRect ); virtual;
    procedure SizeLimits( var AMin, AMax: TPoint ); virtual;
    function NewSubControl( const AItemBounds: TRect; AMenu: PMenu;
        AParentMenu: PCustomMenu ): PCustomMenu; virtual;
end;

```

4.19. Komponenta TMenuPopup

Komponenta **TMenuPopup** implementuje tzv. popup menu. Popup menu může být kdykoli z aplikace manuálně vyvoláno a zobrazeno v libovolném místě na displeji. Na následujících obrázcích jsou zobrazeny ukázky této komponenty.



Komponenta **TMenuPopup** je určena pro terminály s dotykovým panelem, lze ji však s omezením použít i pro terminály vybavené pouze klávesnicí.

U komponenty **TMenuPopup** lze nastavovat několik parametrů, které ovlivňují její chování, příp. vykreslování. Lze nastavit:

- Zobrazení rámečku kolem komponenty

U jednotlivých položek zobrazených komponentou **TMenuPopup** lze nastavit tyto vlastnosti:

- Text položky zarovnaný k levému okraji menu
- Text položky zarovnaný k pravému okraji menu
- Akci, která se má provést při kliknutí panelu, nebo stisku přiřazené klávesy

Při výběru položky nebo při stisku přiřazené klávesy panelu komponenta **TMenuPopup** může vyvolat následující akce:

- Generování oznámení `nmClick`
- Skok na definovanou stránku komponenty **TPageControl**
- Volání definované stránky komponenty **TPageControl**
- Návrat do předchozí stránky komponenty **TPageControl**
- Ukončení modálního stavu

4.19.1. Položky a metody třídy TMenuPopup

Třída **TMenuPopup** má následující konstruktor:

```
constructor Init( const ABounds: TRect; AMenu: PMenu );
```

Popis parametrů konstruktoru je uveden v následující tabulce:

ABounds	Umístění komponenty v rámci vlastníka. Rozměry klávesnice jsou automaticky dopočítány při vložení komponenty do skupiny vlastníka.
AMenu	Odkaz na strukturu stromového menu alokovanou na hromadě. Viz. kapitola 4.19.2.

Komponenta **TMenuPopup** rozšiřuje báзовou třídu **TControl** o položky uvedené v následující tabulce:

Flags	Položka upřesňuje chování komponenty. Obsahuje kombinaci následujících příznaků.	
	mfMenuBar	Příznak určující zda se jedná o komponentu TMenuBar . Pokud tento příznak není nastaven, pak se jedná o komponentu TMenuBox nebo jejího potomka. Tento příznak nesmí být nastaven.
	mfBorder	Zobrazení rámečku kolem komponenty
	Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetFlags .	
Pages	V případě, že některá z položek menu provádí při výběru přepnutí stránky, pak je potřeba aby tento parametr obsahoval odkaz na příslušnou komponentu TPageControl , ve které k přepnutí dojde. Jinak může být nastaven na hodnotu nil .	
Menu	Položka obsahuje odkaz na první úroveň menu (tj. strukturu TMenu). Položka je inicializovaná parametrem konstruktoru a je určena pouze pro čtení.	
Selected	Položka obsahuje odkaz na vybranou položku menu (tj. strukturu	

	TMenuItem). Položka je nastavována automaticky při procházení menu a je určena pouze pro čtení.
--	---

Komponenta **TMenuPopup** rozšiřuje básovou třídu **TControl** o metodu uvedené v následující tabulce:

SetFlags	Metoda pro nastavení položky Flags.
-----------------	-------------------------------------

4.19.2. Vytvoření struktury menu

Stromové menu je tvořena pomocí struktur **TMenu** (podmenu) a **TMenuItem** (položka menu) alokovaných na hromadě. K vytvoření stromového menu slouží funkce **NewMenu**, **NewMenuItem**, **NewSubMenu**, které jsou detailně popsány v dokumentaci ke knihovně Controls v kapitole ke komponentě **TCustomMenu**.

4.19.3. Vytvoření a inicializace komponenty TPopupMenu

Komponenta **TMenuPopup** nefunguje jako běžná komponenta, vždy je potřeba ji spustit v modálním stavu pomocí metody **TGroup.ExecControl**:

Příklad:

```
var
  MyPopup : PMenuPopup;
  R       : TRect;

{ Umístění levého horního rohu komponenty, rozměry jsou dopočítány
  { automaticky }

R.Assign( 0, 0, 0, 0 );

{ Inicializace instance komponenty menu }
MyPopup := New( PMenuPopup, Init( R,
  NewMenu( ... )
));

{ Vložení komponenty do skupiny }
MyGroup^.Insert( MyPopup );

..
..
..

{ Vyvolání popup menu na zadanych souradnicich }
MyPopup^.Popup( 50, 60 );
```

4.19.4. Implicitní nastavení komponenty TMenuPopup

V následující tabulce jsou uvedena implicitní nastavení všech položek komponenty. Všechny ostatní položky, které nejsou v tabulce uvedené, jsou inicializovány na hodnotu 0, **nil**, apod.

Položka	Hodnota
Bounds	parametr ABounds konstrukturu
EventMask	\$FFFF
CaretSize	(1, 8)
State	0
Options	ofSelectable or ofSharedPalette
Font	fidDefault
Palette	#\$0F#\$00#\$00#\$00#\$00#\$0F (sdílená paleta)

Pro komponentu TMenuPopup jsou definovány konstanty nastavení třídy uvedené v následující tabulce. Tyto konstanty lze předat jako parametr metodě **Customize** (viz. manuál ke knihovně Controls).

Konstanta	Význam
ccMenuPopup	Implicitní nastavení definované ovladačem použitého terminálu.

4.19.5. Události obsluhované komponentou TMenuPopup

Komponenta **TMenuPopup** obsluhuje vstupní události podle tabulky níže. Tyto události jsou obsluhovány pouze v případě, že je komponenta v modálním stavu.

Událost	Popis činnosti
evMouseDown evMouseUp evMouseMove	Procházení menu. Akce svázaná s položkou menu je aktivována při uvolnění tlačítka myši.
evKeyDown	
vkLeft, vkUp	Výběr předchozí položky menu
vkRight, vkDown	Výběr následující položky menu
vkHome	Výběr první položky menu
vkEnd	Výběr poslední položky menu
vkEnter	Aktivace akce spojené s položkou menu
vkEsc	Ukončení modálního stavu menu

4.19.6. Oznámení generovaná komponentou TMenuPopup

Komponenta generuje následující oznámení:

Oznámení	Okamžik vyvolání	Parametry oznámení
nmEnter	Komponenta se dostane do ohniska	-
nmExit	Komponenta se dostane mimo ohnisko	-
nmHide	Volání metody Hide	-
nmShow	Volání metody Show	-
nmEnable	Volání metody Enable	-

nmDisable	Volání metody Disable	-
nmClick	Výběr položky menu označené příznakem mifNotify	Položka ItemId struktury oznámení obsahuje hodnotu položky Id struktury TMenuItem .

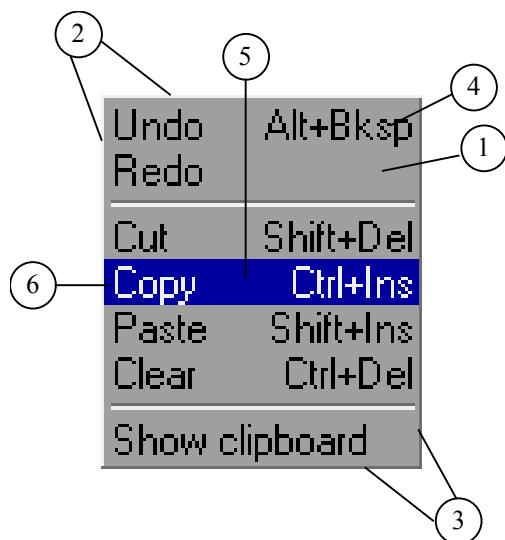
4.19.7. Validace hodnoty komponenty TMenuPopup

S komponentu **TMenuPopup** nelze použít žádný typ validátoru.

4.19.8. Paleta komponenty TMenuPopup

Konstruktor komponenty **TMenuPopup** nastavuje implicitně paletu barev komponenty podle následující tabulky:

#	Barva	Popis
1	\$0F	Pozadí
2	■ \$00	Linka rámečku (vlevo a nahoře)
3	■ \$00	Linka rámečku (vpravo a dole)
4	■ \$00	Text
5	■ \$00	Pozadí vybrané položky
6	\$0F	Text vybrané položky



4.19.9. Reference

Třída **TMenuPopup** je potomkem třídy **TCustomMenu** popsané v dokumentaci ke knihovně **Controls**.

```

PMenuPopup = ^TMenuPopup;
TMenuPopup = object( TMenuBox )
public
  constructor Init( const ABounds: TRect; AMenu: PMenu );
  destructor Done; virtual;
  procedure HandleEvent( var AEvent: TEvent ); virtual;
  procedure Popup( X, Y: Integer );
end;

```

4.19.9.1. Konstruktor TMenuPopup.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```
constructor Init( const ABounds: TRect; AMenu: PMenu );
```

Parametry:

ABounds	Umístění a rozměry komponenty v rámci jejího vlastníka.
AMenu	Odkaz na první úroveň menu, tj. strukturu TMenu vytvořenou pomocí funkce NewMenu (viz. kapitola 4.19.2)

Návratové hodnoty:

Konstruktor nevrací žádnou hodnotu.

Poznámky:

Konstruktor **Init** nastaví všechny položky instance na implicitní hodnoty, viz. tabulka v kapitole 4.19.4.

Konstruktor **Init** resetuje příznak sfVisible položky State. Komponenta není tedy po vytvoření a vložení do skupiny viditelná. Komponenta se stane viditelnou až po zavolání metody **Popup** (viz. kapitola 4.19.9.3).

4.19.9.2. Destruktor TMenuPopup.Done

Destruktor **Done** provádí uvolnění prostředku alokovaných konstruktorem instance..

```
destructor Done; virtual;
```

Parametry:

Destruktor nemá žádné parametry.

Návratové hodnoty:

Destruktor nevrací žádnou hodnotu.

Poznámky:

Destruktor **Done** uvolní strukturu menu alokovanou na hromadě.

4.19.9.3. Procedure TMenuPopup.Popup

Procedura **Popup** provádí vyvolání komponenty popup menu v zadaném místě.

```
procedure Popup( X, Y: Integer );
```

Parametry:

X	X-ová souřadnice levého horního rohu komponenty.
Y	Y-ová souřadnice levého horního rohu komponenty.

Návratové hodnoty:

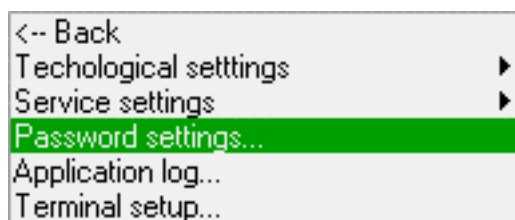
Metoda nevrací žádnou hodnotu.

Poznámky:

Metoda **Popup** provádí zobrazí komponentu na zadaných souřadnicích a spustí ji v modálním režimu. Návrat se provede až po výběru jedné z položek menu nebo po stisku klávesy vkEsc. Těsně před návratem je provedena akce spojená s vybranou položkou.

4.20. Komponenta TMenuOverlay

Komponenta **TMenuOverlay** je určena pro zobrazení stromového menu. Na následujícím obrázku je ukázka této komponenty.



Komponenta zobrazuje vždy jednu úroveň menu. Pokud je počet položek úrovně menu větší než je počet zobrazených řádek, pak lze se zobrazeným výřezem menu rolovat nahoru a dolů.

Komponenta **TMenuOverlay** je určena pro všechny typy terminálu.

U komponenty **TMenuOverlay** lze nastavovat několik parametrů, které ovlivňují její chování, příp. vykreslování. Lze nastavit:

- Zobrazení rámečku kolem komponenty

U jednotlivých položek menu lze nastavit tyto parametry:

- Text položky zarovnaný k levému okraji menu
- Text položky zarovnaný k pravému okraji menu
- Akci, která se má provést při kliknutí panelu, nebo stisku přiřazené klávesy

Při kliknutí panelu, nebo při stisku přiřazené klávesy panelu komponenta **TMenuOverlay** může vyvolat následující akce:

- Generování oznámení nmClick

- Skok na definovanou stránku komponenty TPageControl
- Volání definované stránky komponenty TPageControl
- Návrat do předchozí stránky komponenty TPageControl
- Ukončení modálního stavu

4.20.1. Položky a metody třídy TMenuOverlay

Třída **TMenuOverlay** má následující konstruktor:

```
constructor Init( const ABounds: TRect;
  AScrollBar: PCustomScrollBar; AMenu: PMenu );
```

Popis parametrů konstruktoru je uveden v následující tabulce:

ABounds	Umístění a rozměry komponenty v rámci vlastníka
AScrollBar	Odkaz na přidružený posuvník, tj. inicializovanou komponentu TScrollBar , která bude zobrazovat aktuálně viditelný výřez ze seznamu položek v jedné úrovni menu. Parametr může mít hodnotu nil .
AMenu	Odkaz na strukturu stromového menu alokovanou na hromadě. Viz. kapitola 4.20.2.

Komponenta **TMenuOverlay** rozšiřuje básovou třídu TControl o položky uvedené v následující tabulce:

Flags	Položka upřesňuje chování komponenty. Obsahuje kombinaci následujících příznaků:	
	<table border="1"> <tr> <td>mofBorder</td> <td>Zobrazení rámečku kolem komponenty</td> </tr> </table> Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetFlags .	mofBorder
mofBorder	Zobrazení rámečku kolem komponenty	
ItemHeight	Výška oblasti pro zobrazení textu položky menu. Pokud je hodnota této položky menší nebo rovna nule, pak je výška této oblasti dána výškou znaků použitého fontu zvětšená o absolutní hodnotu této položky. Položka je určena pouze pro čtení. Pro její nastavení použijte metodu SetItemHeight .	
Pages	V případě, že některá z položek menu provádí při výběru přepnutí stránky, pak je potřeba aby tento parametr obsahoval odkaz na příslušnou komponentu TPageControl , ve které k přepnutí dojde. Jinak může být nastaven na hodnotu nil .	
Menu	Položka obsahuje odkaz na první úroveň menu (tj. strukturu TMenu). Položka je inicializovaná parametrem konstruktoru a je určena pouze pro čtení.	
Current	Položka obsahuje odkaz na aktuálně zobrazenou úroveň menu (tj. strukturu TMenu). Položka je nastavována automaticky při procházení menu a je určena pouze pro čtení.	
Selected	Položka obsahuje odkaz na vybranou položku menu (tj. strukturu TMenuItem). Položka je nastavována automaticky při procházení menu a je určena pouze pro čtení.	
ScrollBar	Položka obsahuje odkaz na přidruženou komponentu TScrollBar .	

	Položka je nastavována parametrem konstrukturu a je určena pouze pro čtení.
--	---

Komponenta **TMenuOverlay** rozšiřuje báзовou třídu **TControl** o metodu uvedené v následující tabulce:

SetFlags	Metoda pro nastavení položky Flags
SetItemHeight	Metoda pro nastavení položky ItemHeight

4.20.2. Vytvoření struktury menu

Stromové menu je tvořena pomocí struktur **TMenu** (podmenu) a **TMenuItem** (položka menu) alokovaných na hromadě. K vytvoření stromového menu slouží funkce **NewMenu**, **NewMenuItem**, **NewSubMenu**, které jsou detailně popsány v dokumentaci ke knihovně Controls v kapitole ke komponentě **TCustomOverlayMenu**.

4.20.3. Implicitní nastavení komponenty TMenuOverlay

V následující tabulce jsou uvedena implicitní nastavení všech položek komponenty. Všechny ostatní položky, které nejsou v tabulce uvedené, jsou inicializovány na hodnotu 0, **nil**, apod.

Položka	Hodnota
Bounds	parametr ABounds konstrukturu
EventMask	\$FFFF
CaretSize	(1, 8)
State	sfVisible
Options	ofSelectable or ofSharedPalette or ofFirstClick
Font	fidDefault
Palette	#\$00 (sdílená paleta)
Menu	podle parametru AMenu konstrukturu
Current	kopie Menu
TopItem	kopie Current^.Items
Selected	kopie Current^.Items
Flags	0
Pages	nil
ItemHeight	0
ScrollBar	podle parametr AMenu konstrukturu

Pro komponentu **TMenuOverlay** jsou definovány konstanty nastavení třídy uvedené v následující tabulce. Tyto konstanty lze předat jako parametr metodě **Customize** (viz. manuál ke knihovně Controls).

Konstanta	Význam
ccMenuOverlay	Implicitní nastavení definované ovladačem použitého terminálu.

4.20.4. Události obsluhované komponentou TMenuOverlay

Komponenta **TMenuOverlay** obsluhuje vstupní události podle následující tabulky

Událost		Popis činnosti
evMouseDown		Výběr položky menu
evMouseDown		Provedení akce svázané s položkou menu
evKeyDown	vkLeft	Návrat do předchozí úrovně menu.
	vkRight	Provedení akce svázané s položkou menu.
	vkUp	Výběr předchozí položky
	vkDown	Výběr následující položky
	vkHome	Výběr první položky menu
	vkEnd	Výběr poslední položky menu
	vkEnter	Provedení akce svázané s položkou menu
	vkEsc	Návrat do předchozí úrovně menu

Klávesy vkLeft a vkEsc nejsou komponentou obsluhovány v případě podmenu první úrovně (tj. kořenového menu).

4.20.5. Oznámení generovaná komponentou TMenuOverlay

Komponenta generuje následující oznámení:

Oznámení	Okamžik vyvolání	Parametry oznámení
nmEnter	Komponenta se dostane do ohniska	-
nmExit	Komponenta se dostane mimo ohnisko	-
nmHide	Volání metody Hide	-
nmShow	Volání metody Show	-
nmEnable	Volání metody Enable	-
nmDisable	Volání metody Disable	-
nmCanExit	Před opuštěním ohniska, pokud je nastaven příznak ofValidate v položce Options	Položka Accept struktury oznámení obsahuje hodnotu False. Opuštění ohniska lze zakázat nastavením této položky na hodnotu False.
nmClick	Výběr položky označené příznakem mifNotify.	Položka ItemId struktury oznámení obsahuje hodnotu položky Id struktury TMenuItem .

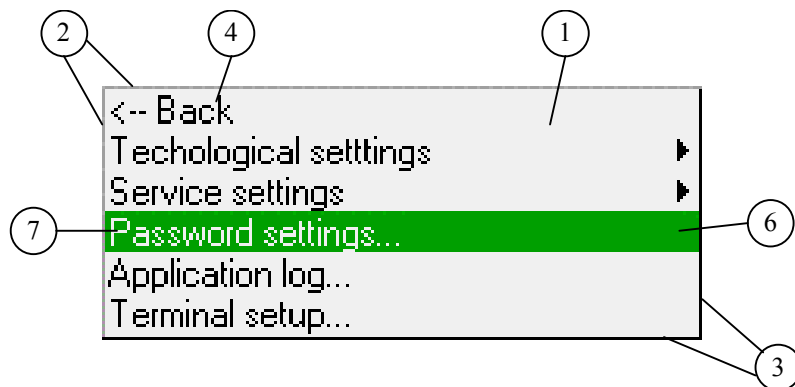
4.20.6. Validace hodnoty komponenty TMenuOverlay

S komponentu **TMenuOverlay** nelze použít žádný typ validátoru.

4.20.7. Paleta komponenty TMenuOverlay

Konstruktor komponenty **TMenuOverlay** nastavuje implicitně paletu barev komponenty podle následující tabulky:

#	Barva	Popis
1	\$0F	Pozadí
2	■ \$00	Linka rámečku (vlevo a nahoře)
3	■ \$00	Linka rámečku (vpravo a dole)
4	■ \$00	Text položky (zarovnaný k levému okraji)
5	■ \$00	Text položky (zarovnaný k pravému okraji)
6	■ \$00	Pozadí vybrané položky
7	■ \$00	Text vybrané položky (zarovnaný k levému okraji)
8	■ \$00	Text vybrané položky (zarovnaný k pravému okraji)



4.20.8. Reference

Třída **TMenuOverlay** je potomkem třídy **TCustomMenuOverlay** popsané v dokumentaci ke knihovně Controls.

```

PMenuOverlay = ^TMenuOverlay;
TMenuOverlay = object( TCustomMenuOverlay )
public
    Flags          : Word;
    ItemHeight     : Integer;

    constructor Init( const ABounds: TRect;
                     AScrollBar: PCustomScrollBar; AMenu: PMenu );

    procedure Paint( ACanvas: PCanvas ); virtual;
    procedure PaintItems( AItem1, AItem2: PMenuItem ); virtual;
    function GetRowHeight: Integer; virtual;
    function GetRowCount: Integer; virtual;
    function GetItemAtPos( APos: TPoint ): PMenuItem; virtual;
    procedure SetFlags( ASet, AReset: Word ); virtual;
    procedure SetItemHeight( AItemHeight: Integer );
end;

```

4.20.8.1. Konstanty mof_

Konstanty s prefixem `mof_` upřesňují chování komponenty **TMenuOverlay**. Kombinace konstant je uložena v položce `Flags` této komponenty (viz. kapitola 4.20.8.2).

Identifikátor	Kód	Popis
<code>mofBorder</code>	<code>\$0001</code>	Zobrazení rámečku kolem komponenty

4.20.8.2. Položka TMenuOverlay.Flags

Položka **Flags** upřesňuje chování komponenty. Obsahuje kombinaci příznaků s prefixem `mof_` popsanych v kapitole 4.20.8.1. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetFlags** (viz. kapitola 4.16.8.19).

```
Flags          : Word;
```

4.20.8.3. Položka TMenuOverlay.ItemHeight

Položka **ItemHeight** obsahuje výšku oblasti (tj. počet pixelů) položky menu. Pokud je hodnota této položky menší nebo rovna nule, pak je výška této oblasti dána výškou znaků použitého fontu zvětšená o absolutní hodnotu této položky. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetItemHeight** (viz. kapitola 4.20.8.6).

```
ItemHeight     : Integer;
```

4.20.8.4. Konstruktor TMenuOverlay.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```
constructor Init( const ABounds: TRect; AScrollBar: PCustomScrollBar;
  AMenu: PMenu );
```

Parametry:

ABounds Umístění a rozměry komponenty v rámci jejího vlastníka.
AScrollBar Odkaz na přidružený posuvník, tj. komponentu **TScrollBar**. Parametr může mít hodnotu **nil**.
AMenu Odkaz na první úroveň menu, tj. strukturu **TMenu** vytvořenou pomocí funkce **NewMenu** (viz. kapitola 4.19.2)

Návratové hodnoty:

Konstruktor nevrací žádnou hodnotu.

Poznámky:

Konstruktor **Init** nastaví všechny položky instance na implicitní hodnoty, viz. tabulka v kapitole 4.19.4.

Struktura menu daná parametrem AMenu je uvolněna při volání destrukturu **Done**.

4.20.8.5. Metoda TMenuOverlay.SetFlags

Metoda **SetFlags** slouží k nastavení položky Flags (viz. kapitola 4.20.8.2)

```
procedure SetFlags( ASet, AReset: Word ); virtual;
```

Parametry:

AFlags Kombinace příznaků s prefixem mof_ (viz. kapitola 4.20.8.1).

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.

4.20.8.6. Metoda TMenuOverlay.SetItemHeight

Metoda **SetItemHeight** slouží k nastavení výšky položky menu, tj. nastavení položky ItemHeight (viz. kapitola 4.20.8.3).

```
procedure SetItemHeight( AHeight: Integer );
```

Parametry:

AHeight Parametr udává výšku položky menu (tj. počet pixelů). Pokud je hodnota tohoto parametru menší nebo rovna nule, pak je výška oblasti dána výškou znaků použitého fontu zvětšená o absolutní hodnotu této položky.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Pokud je to nutné, metoda provede překreslení komponenty pomocí metody **Repaint**.