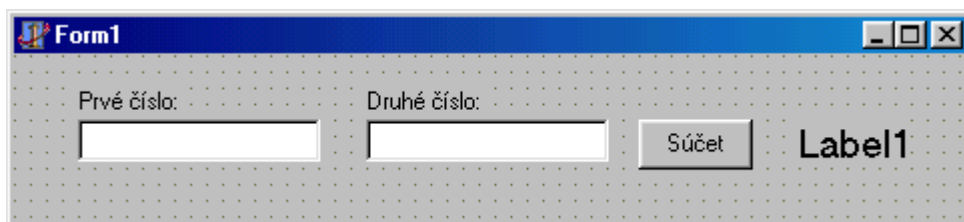


Kalkulačka

Kalkulačka patrí k jednoduchým programom, na ktorých sa dajú vysvetliť a pochopiť základné vlastnosti aplikácií v Delphi.

Vytvorme teda najprv aplikáciu, ktorá bude sčítavať dve celé čísla.



Ako prvé potrebujeme zrejme vytvoriť prostredie. Mohlo by byť podobné tomu na obrázku.. Vložené texty sú objekty *Label*, ktorým sme zmenili vlastnosť *caption* – napísali sme do nej text, ktorý chceme mať zobrazený.

Textové polia na zadávanie čísel sú objekty *Edit*, ktoré síce po vložení obsahujú vísaný text (*Edit1*, *Edit2*...), ale ten sme zrušili jeho vymazaním vo vlastnosti *Text*. Tlačidlo je *Button* so zmenenou vlastnosťou *Caption*, no a na zobrazenie výsledku použijeme opäť *Label*. Jeho obsah môžeme cez *Caption* prepísať na 0, alebo pokojne nechať taký ako je.

Všimnime si jednu vec: text, ktorý sa má zobrazovať v objekte sa zvyčajne nastavuje pomocou vlastnosti *Caption* (*Label*, *Button*, *Form*,...), prípadne v niekoľko málo prípadoch vo vlastnosti *Text* (*Edit*).

Po vytvorení prostredia nám zostáva už napísať “len” obslužný kód pre vykonanie operácie.

Prvá otázka: kedy sa má sčítavanie spustiť?

Prvá odpoveď: zrejme keď stlačíme tlačidlo.

Druhá otázka: ako povieme, že kód sa má vykonať až vtedy, keď sa tlačidlo stlačí?

Druhá odpoveď: zapísaním kódu do udalosti pri kliknutí (*onClick*). Nastaviť, aby sa kód naozaj vykonal práve vtedy možno dvoma spôsobmi (uviedli sme ich už aj vyššie): dvojklikom na tlačidlo (Delphi nás prepne do kódu, kde sa automaticky vytvorí procedúra s názvom *Button1Click*) alebo prepnutím sa v *Object inspectore* na záložku udalostí (*Events*) a dvojklikom do položky *OnClick* (vytvorí sa presne tá istá procedúra).

V oboch prípadoch je teda výsledok rovnaký...

Tretia a posledná otázka: ako napísať kód na vykonanie sčítavania?

Tretia a posledná odpoveď: veľmi jednoducho – pomocou jazyka Pascal a vlastností vložených komponentov.

Samozrejme, táto odpoveď potrebuje trochu presnejšie vysvetlenie. V prvom rade si treba uvedomiť, že v Delphi naozaj nejde o nič iné ako o kombináciu štandardného Pascalu (so všetkými jeho kladmi i záporami) a vytvorených objektov – komponentov. Každý z komponentov má kdesi skrytý svoj zdrojový kód a ten je zvyčajne aspoň pre nami zatiaľ používané komponenty) tiež napísaný v Pascale.

Pri programovaní teda používame zápis deklarácie, príkazov i procedúr rovnako ako v Pascale.

No a o tom budú aj nasledovné riadky.

Potrebujeme sčítať dve celé čísla. Prvým krokom by teda mohla byť ich deklarácia v tele automaticky vytvorenej procedúry:

```
procedure TForm1.Button1Click(Sender:TObject)
var a,b,vysledok:integer

begin

end;
```

Do premennej *a* načítame prvé číslo, do premennej *b* druhé. *Vysledok* bude obsahovať ich súčet. Pred samotným odštartovaním je však potrebné zdôrazniť, že akýkoľvek reťazec (je jedno či text

alebo číslo), ktorý sa zapisuje alebo číta z *Editu* je automaticky a nevyhnutne typu *string*. Z tohto dôvodu potrebujeme pred uložením zadanej hodnoty vykonať prevod zo *stringu* na celé číslo. Používa sa na to funkcia *StrToInt* (string na integer). Zápis potom bude vyzeráť nasledovne:

```
procedure TForm1.Button1Click(Sender:TObject)
var a,b,vysledok:integer

begin
  a:=StrToInt(Edit1.Text);
  b:=StrToInt(Edit2.Text);

end;
```

Do premennej *a* vložíme číslo, ktoré sa nachádza v *Edit1*. Vlastnosť, ktorá nás informuje o tom, čo je v *Editu* zapísané je *Text*. Prečítaný obsah je typu *string*, pre ktorý (ako si iste pamätáme z Pascalu) sčítovanie nefunguje. pomocou funkcie *StrToInt* ho prevedieme na číslo a uložíme do premennej *a*. Podobne druhé číslo prejde do premennej *b*.

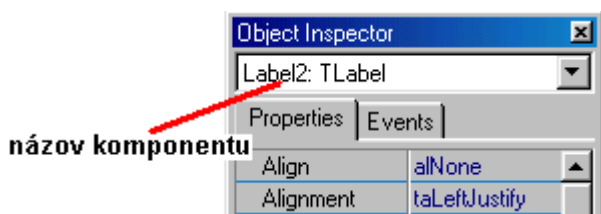
```
procedure TForm1.Button1Click(Sender:TObject)
var a,b,vysledok:integer

begin
  a:=StrToInt(Edit1.Text);
  b:=StrToInt(Edit2.Text);
  vysledok:=a+b;
  Label1.Caption:=IntToStr(vysledok);
end;
```

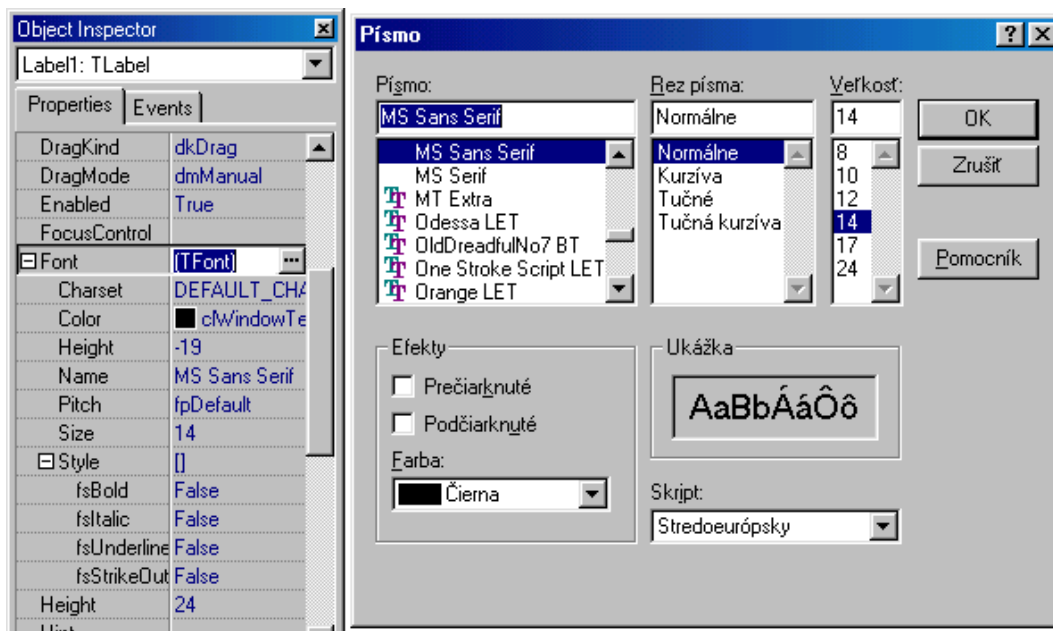
Súčet vykonáva klasické priradenie a výsledok sa zobrazí v *Labeli*. Zobrazovaný text *Labelu* je uložený vo vlastnosti *Caption*, ktorá je opäť typu *string*. Číslo na *string* prevedieme pomocou obrátenej funkcie – *IntToStr*.

A výsledok je na svete...

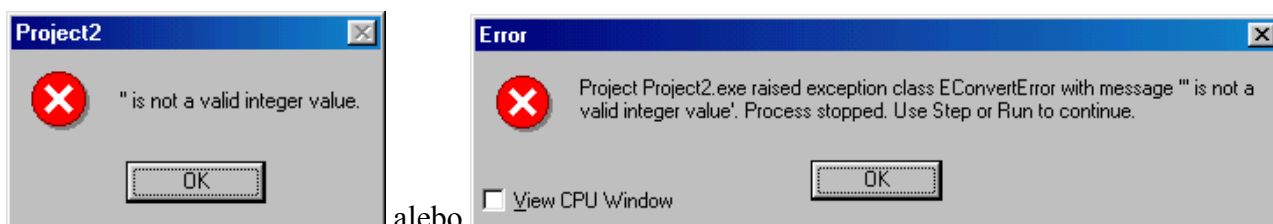
V prípade, že váš *Label* sa nevolá *Label1*, treba jeho názov upraviť podľa svojho. Názov nájdete vo vlastnosti *Name* alebo rýchlejšie v hornej časti *Object inspectoru* po kliknutí na príslušný objekt.



Možno ste si všimli, že *Label* s výsledkom je na našom obrázku podstatne väčší ako *labely* s popisným textom k *Editom*. Je to spôsobené zväčšením písma – zmenou fontu. Prakticky každý objekt obsahujúci text má k dispozícii vlastnosť *Font*, ktorá po dvojkliku na pravej strane zobrazí štandardný *FontDialog* alebo po rozbalení dovoľuje meniť vlastnosti písma po položkách.



Na záver by to chcelo upozorniť, že funkcia *StrToInt* je pomerne hlúpa – ak sa pokúsite previesť na číslo prázdny text alebo neplatné číslo (písmená, medzery...) – vyhodí chybu a preruší vykonávanie programu. Nedivte sa preto, ak sa v prípade nesprávneho čísla zobrazí hláška:



alebo

Informuje len o tom, že zadaný reťazec nie je celé číslo. Vyriešiť tento problém možno napr. pomocou procedúry *Val* známej už z Pascalu.

Vytvorte ďalšie tri tlačidlá pre súčin, rozdiel a podiel (môže byť celočíselné delenie pomocou div a mod) alebo desatinné (v takom prípade bude výsledkom reálne číslo, na čo treba myslieť pri deklarácii i pri prevode na string – používa sa funkcia FloatToStr).

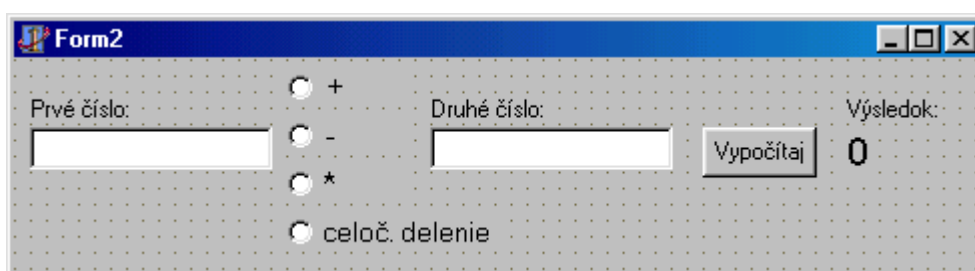
Postup je analogický ako v predchádzajúcom prípade, mení sa len znamienko pri výpočte.



RadioButton

Upravte kalkulačku tak, aby výpočet spúšťalo jediné tlačidlo, a aby sa operácia dala vykonať podľa nastavenia v RadioButtone.

Formulár môže vyzeráť napr. nasledovne:



Prázdné krúžky sú *RadioButton*y, ktoré majú tú vlastnosť, že len jeden z nich môže byť zapnutý (bude obsahovať čierny kruh). Momentálne nie je zapnutý ani jeden, ak však chceme nastaviť niektorú z operácií ako primárnu (automaticky sa zaškrtnie po spustení), môžeme tak urobiť pomocou vlastnosti *Checked*. V prípade, že je táto nastavená na *False*, *RadioButton* zaškrtnutý nie je, ak sa nastaví na *True*, zaškrtnie sa. Nastavenie (zaškrtnutie) ktoréhokoľvek *RadioButonu* na formulári má za následok vypnutie všetkých ostatných (naraz môže byť vybraný len jeden).

Pokiaľ by ste potrebovali mať vybraných naraz viac položiek, môžete použiť *CheckBox*, *RadioGroup* alebo *Panel*.

Nastavme teraz ako primárne zvolenú operáciu súčtu a pustime sa do písania kódu. Jediná udalosť, ktorú potrebujeme obslúžiť je opäť kliknutie na tlačidlo.

V prvom rade prečítame čísla zadané v *Editoch*, potom zistíme, ktorý *RadioButton* je vybraný, podľa neho vykonáme výpočet a napokon výsledok vložíme do *Labelu*.

```
procedure TForm1.Button1Click(Sender:TObject)
var a,b,vysledok,zvysok:integer

begin
  a:=StrToInt(Edit1.Text);
  b:=StrToInt(Edit2.Text);
  if RadioButton1.Checked then vysledok:=a+b;
  if RadioButton2.Checked then vysledok:=a-b;
  if RadioButton3.Checked then vysledok:=a*b;
  if RadioButton4.Checked then begin
    vysledok:=a div b;
    zvysok:=a mod b;
  end;

  Label1.Caption:=IntToStr(vysledok);
  if RadioButton4.Checked then
    Label1.Caption:= Label1.Caption + ', zvysok: ' + IntToStr(zvysok);
end;
```

Pre prvé tri *radiobuttony* by malo byť všetko jasné – ak sú zaškrtnuté (čo vyjadruje ich vlastnosť *checked* nastavená na *true*), tak výsledok vypočítame príslušným spôsobom. Ak sme zvolili celočíselné delenie, potrebujeme okrem výsledku zistiť aj zvyšok.

Pre výpis v prvých troch prípadoch stačí priradiť výsledok do *Labelu*. Pokiaľ sme celočíselne delili, potrebujeme vypísať aj zvyšok. K výsledku už zobrazenému v *Labeli* pridáme ďalší textový reťazec: *zvysok*: a prevedenie jeho hodnoty na *string*. Výsledný reťazec potom môže vyzeráť napr. takto:

2, zvysok: 5

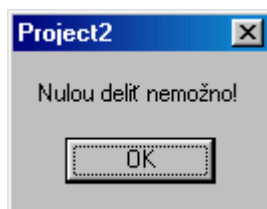
Všetko je na prvý pohľad v poriadku, ak však pri delení zadáme ako deliteľa nulu, aplikácia nám vyhodí chybu. Na ošetrenie by postačila podmienka a v prípade nuly výpis upozornenia do *labelu*. Ak však chceme dodržať štandardy, patrí sa v takomto prípade vypísať upozornenie do osobitného okna.

Jedným z najjednoduchších spôsobov je použitie funkcie *ShowMessage*, ktorá zobrazí v osobitnom okne text. Kód môže vyzeráť takto:

```
...
if RadioButton4.Checked then begin
  if b=0 then begin
    ShowMessage('Nulou deliť nemožno');
    exit;
  end
  vysledok:=a div b;
  zvysok:=a mod b;
end;
...
```

V prípade zadania nuly teda zobrazíme správu a príkazom *exit* opustíme procedúru – nevykonajú sa teda žiadne operácie nasledujúce za týmto príkazom (výpočet, výpis).

Správa bude vyzerat' podobne ako táto:



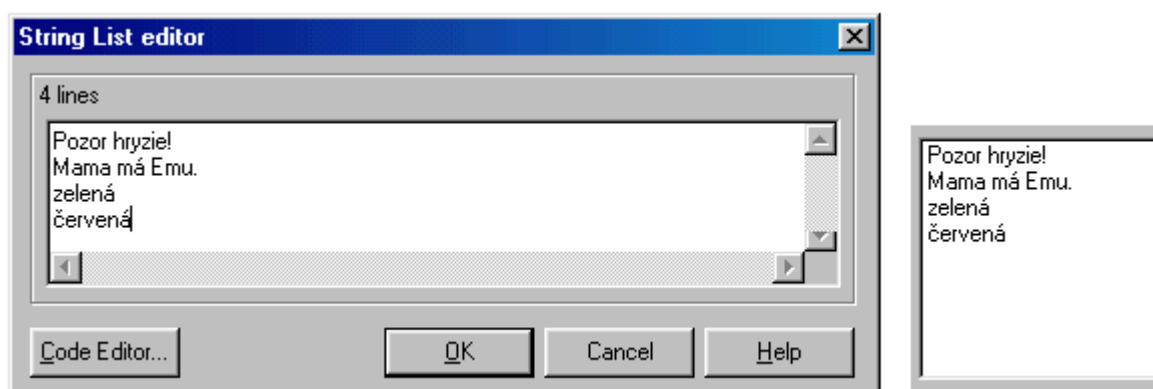
Pridajte do tejto aplikácie komponent, v ktorom budete evidovať všetky doposiaľ prevedené výpočty



ListBox

Na riešenie tohto problému môžeme použiť niekoľko rozličných komponentov. Najjednoduchším je asi použitie *ListBoxu*. Ide vlastne o akýsi zoznam, do ktorého môžeme prakticky neobmedzene pridávať položky a vieme v ňom listovať.

Skúsme si ho najprv predstaviť všeobecne a neskôr ho použijeme na vyriešenie zadanej úlohy. Text uvedený v jednotlivých riadkoch *listboxu* je zapísaný v jeho položke *Items*. Nejde o súvislý text (ako sme doposiaľ videli pri *edite* alebo *labeli*), ale o akýsi zoznam riadkov. Dvojklikom na *Items* pre *Listbox* v *Object inspectore* zobrazíme editor, v ktorom ich môžeme zadať. Texty, ktoré napíšeme pod seba sa v komponente zobrazujú ako osobitné riadky.



Výsledok vidíme okamžite (už v dobe návrhu) po odsúhlasení v listboxe umiestnenom na formulári. Po spustení aplikácie máme štandardne možnosť vyznačiť jednu položku (item) listboxu kliknutím na riadok.

Okrem naplnenia položkami v návrhu sa oveľa častejšie môžeme stretnúť s napĺňaním počas behu programu. Zápis je nasledovný:

```
Listbox1.Items.Add('Toto je môj text');
```

Inými slovami do *listboxu1* novú položku pridaj - s textom (...). Pri operáciách s položkami je na prvom mieste vždy názov listboxu, s ktorým ideme pracovať. Nasleduje informácia o tom, že ideme manipulovať s jeho položkami (*items*) a napokon uvedieme operáciu, ktorú chceme vykonať.

Pochopiť tento prístup je pomerne dôležité, pretože podobne sa pracuje aj s mnohými ďalšími komponentmi (*ComboBox*, *Memo*, *RadioGroup*...).

Operácie s *listboxom* možno zhrnúť nasledovne:

operácia	zápis	poznámka
pridať položku	<code>listbox1.Items.Add('moj text')</code>	

operácia	zápis	poznámka
získať text i-tej položky	<code>polozka:=listbox1.Items.Strings[i];</code>	<i>polozka</i> je typu string
odstrániť i-tu položku	<code>listbox1.Items.Delete(i);</code>	
počet položiek v listboxe	<code>pocet:=listbox1.Items.Count</code>	<i>pocet</i> je typu integer
vymazať všetky položky z listboxu	<code>listbox1.Items.Clear;</code>	
zistiť na ktorej položke listboxu je nastavený kurzor	<code>cislo:=listbox1.ItemIndex;</code>	<i>cislo</i> je typu integer pri výbere prvej položky je hodnota 0, druhej 1 atď. Pokiaľ nie je vybraná žiadna položka, je hodnota nastavená na -1
vypísať text na vybranej položke	<code>mytext:=listbox1.Items.Strings[Listbox1.ItemIndex]</code>	ide vlastne o kombináciu z dvoch predchádzajúcich riadkov. <i>ItemIndex</i> vráti poradové číslo položky, ktorá je aktívna a <i>Items.Strings[]</i> vráti text, ktorý je v aktívnej položke uložený.
zistiť, či sa daná položka v listboxe už nachádza a ak áno, vrátiť jej poradové číslo	<code>porcislo:=listbox1.Items.IndexOf('moj text');</code>	<i>porcislo</i> vráti poradové číslo položky, ktorá obsahuje zadaný text. V prípade, že sa takáto položka v listboxe nenachádza, vráti -1.

Vráťme sa konečne k zadaniu úlohy, ktoré od nás vyžaduje viesť históriu vykonaných výpočtov. Ak si úlohu rozoberieme slovne, stačí vlastne po každom prebehnutí výpočtu zapísať do listboxu čísla, s ktorými manipulujeme, operáciu medzi nimi a prípadne výsledok.

Riešenie by mohlo vyzerat' nasledovne:

```
procedure TForm1.Button1Click(Sender:TObject)
var a,b,vysledok,zvysok:integer;
    riadok:string;

begin
  a:=StrToInt(Edit1.Text);
  b:=StrToInt(Edit2.Text);
  riadok:=Edit1.Text; // *** zapamatom si prve zadane cislo

  // *** podla zadanych operacii pridam za prve cislo znamienko matematickej operacie
  if RadioButton1.Checked then begin vysledok:=a+b; riadok:=riadok+'+'; end;
  if RadioButton2.Checked then begin vysledok:=a-b; riadok:=riadok+'-'; end;
  if RadioButton3.Checked then begin vysledok:=a*b; riadok:=riadok+'*'; end;
  if RadioButton4.Checked then begin
    vysledok:=a div b;
    zvysok:=a mod b;
    riadok:=riadok+' div ';
  end;

  riadok:=riadok+Edit2.Text; // *** za druhu operaciu pridam druhe zadane cislo
  Label1.Caption:=IntToStr(vysledok);
  if RadioButton4.Checked then
    Label1.Caption:= Label1.Caption + ', zvysok: ' + IntToStr(zvysok);

  // *** aby som nemusel znovu robit vypocty a prevody jednoducho prevezmem vysledok
  // *** zapisany v Labeli
  riadok:=riadok+'='+Label1.Caption;

  // *** a ziskany riadok napokon pridam do listboxu
  ListBox1.Items.Add(riadok);
end;
```

Vysvetlenie je uvedené priamo v zdrojovom kóde.