

## Komentáre

Pre prehľadnejší obsah zdrojového kódu je možné tak, ako aj v iných programovacích jazykoch, používať komentáre. Komentáre predstavujú zápis informácií určených pre programátora, ktoré sa v programe nezobrazujú a pri kompilácii sa ignorujú. V komentároch sa najčastejšie uvádzajú informácie k uvedenému programovému kódu, čím sa sprehľadňuje obsah zdrojového súboru a zároveň zjednodušuje pochopenie programovej logiky. Komentármi je zároveň možné deaktivovať a aktivovať jednotlivé časti programového kódu.

Keďže komentáre nie sú spracovávané, môžu obsahovať diakritiku. Vo vývojovom prostredí Lazarus sú v menu *Upraviť* k dispozícii funkcie *Zakomentovať výber*, *Odkomentovať výber* a *Toggle comment*.

V jazyku Free Pascal je možné použiť tri typy komentárov. Dva zápisy sú určené pre viacriadkové komentáre a jeden pre jednoriadkový komentár. Jednoriadkový komentár sa inicializuje zápisom znakov `//`. Viacriadkový komentár sa umiestňuje medzi dvojicu znakov `(* a *)` alebo znaky `{ a }`. Pri jednoriadkovom komentári sa text za znakmi `//` až do konca riadku nespracuje. Znak pre zápis komentára nie je možné navzájom kombinovať.

*Zápis jednoriadkového komentára prostredníctvom znakov `//`.*

```
// toto je jednoriadkový komentár
```

*Zápis viacriadkového komentára s dvojicou znakov `(* a *)`.*

```
(* toto je  
komentár na viac riadkov *)
```

*Zápis viacriadkového komentára s dvojicou znakov `{ a }`.*

```
{ toto je  
komentár na viac riadkov }
```

*Nesprávne spôsoby zápisov viacriadkového komentára.*

```
(*toto je nesprávny  
komentár na viac riadkov }
```

```
(*toto je nesprávny  
komentár na viac riadkov //
```

```
//toto je nesprávny  
komentár na viac riadkov }
```

**Príklad č.9.** Použitie rôznych typov komentárov v programe.

```
program Komentare;  
// toto je jednoriadkový komentár  
(* toto je  
komentár na viac riadkov *)  
begin  
  writeln('Ahoj svet.');  readln;  
  { toto je  
  komentár na viac riadkov }  
end.
```

Zápis komentára prostredníctvom znakov `//` je rozšírením Free Pascalu a tak nemusí byť podporovaný všetkými prekladačmi. Tento zápis nepozná Pascal ani Turbo Pascal. Norma ISO/IEC 7185:1990 uvádza len znaky `{ }`, `(* *)`. Do týchto znakov je samozrejme možné umiestniť aj komentár na jeden riadok.

*Zápis jednoriadkového komentára medzi znaky `{ }`.*

```
{ toto je komentár na jeden riadok }
```

Počet komentárov v programe nie je obmedzený. Môžu byť používané v rôznych častiach programu, všade tam, kde sú povolené biele miesta. V rámci praxe sa odporúča používať komentáre čo najčastejšie a čo najvýstižnejšie. Takto aplikované komentáre môžu byť v budúcnosti veľmi užitočnou pomocou. Komentáre prekladač ignoruje.

## Poznámka

Biele miesta sú v odbornej literatúre nazývané aj ako biele znaky alebo prázdne miesta. Ide o znaky špecifického významu, ktoré nie sú viditeľné na obrazovke ani pri tlači. Je ich ale možné zobrazíť v editore, ak editor takúto funkciu podporuje. Medzi tieto znaky patrí medzera, tabulátor a nový riadok.

## Zobrazenie výpisu pred zatvorením okna

Programy v Pascale sa spúšťajú a vykonávajú v okne príkazového riadka operačného systému. Po ukončení programu sa príkazový riadok ihneď zatvorí. Pre zobrazenie výstupu je možné použiť niekoľko zápisov alebo spustiť program v príkazovom riadku. Príkazový riadok sa spustí cez ponuku Štart a položku Spustiť. Do otvoreného okna sa napíše príkaz cmd. V príkazovom riadku sa pre zmenu adresárov používa príkaz cd.

*Príklad skoku do koreňového adresára.*

```
cd \
```

*Príklad posunu o adresárovú úroveň nižšie.*

```
cd ..
```

Zatvorenie okna s príkazovým riadkom sa vykoná príkazom exit. Pre rýchlejšiu prácu s adresármi sa používa kláves tabulátor, ktorý sa stláča po zapísaní príkazu cd a medzery. Za medzerou môžu nasledovať znaky, na základe ktorých budú zobrazované existujúce adresáre. Ak za príkazom cd nie je medzera, tabulátor nie je funkčný. Zmazanie príkazového riadka sa vykoná klávesom Esc.

Zápisom procedúry **readln** v programe program čaká na stlačenie klávesy Enter, respektíve na vloženie reťazca.

```
readln;
```

**Príklad č.10.** Použitie procedúry **readln**.

```
program Vypis;  
begin  
  writeln('Zobraz tento text.');
```

```
  readln;
```

```
end.
```

Zápisom cyklu **repeat until keypressed** program čaká na stlačenie ľubovoľnej klávesy. Podmienkou je pripojenie knižnice crt.

```
uses crt;  
repeat until keypressed;
```

## Dátový typ integer

Dátový typ integer slúži na uchovávanie celých čísiel v rozsahu od -32 768 do 32 767. Dátový typ integer v pamäti zaberá 2 bajty.

integer	2 bajty	-32 768 až 32 767
---------	---------	-------------------

Turbo Pascal okrem celočíselného typu integer podporuje aj ďalšie celočíselné dátové typy.

byte	1 bajt	0 až 255
word	2 bajty	0 až 65535
shortInt	1 bajt	-128 až 127
longInt	4 bajty	-2 147 483 648 až 2 147 483 647

Kompilátor Free Pascal podporuje aj dátové typy longword, qword a int64.

longWord	4 bajty	0 až 4294967295
qword	8 bajtov	0 až 18446744073709551615
int64	8 bajtov	-9223372036854775808 až 9223372036854775807

**Príklad č.16.** Deklarácia premennej dátového typu integer.

```
program DeklaraciaDatovehoTypuInteger;
var  cele_cislo : integer;
begin
  cele_cislo:=1024;
  writeln(cele_cislo);
  readln;
end.
```

Ak sa vykonávajú operácie len s dátovým typom integer, výsledkom bude číslo dátového typu integer, resp. niektoré z celočíselných dátových typov v závislosti od platného rozsahu. V prípade matematických operácií s celočíselnými hodnotami, ktorých výsledok je mimo rozsah príslušnej premennej, dôjde k chybe programu.

### Poznámka

Kým Free Pascal pri prekročení povoleného rozsahu deklarovaného dátového typu vypíše chybu `exitcode=201`, program Lazarus hodnoty mimo rozsah dátového typu môže spracovať. Tento jav môže sťažiť hľadanie nesprávneho fungovania programu.

## Deklarácie

Deklarácia je príkaz, ktorý priradí identifikátoru meno a miesto v pamäti. Deklarácia premennej v Pascale sa vykonáva kľúčovým slovom **var**. Za kľúčovým slovom **var** nasleduje toľko deklarácií premenných, koľko je potrebné pre chod programu. Okrem kľúčového slova **var** podporuje Pascal aj špecifický druh premenných nazývaných konštanty. Konštanty sa definujú prostredníctvom kľúčového slova **const**.

Priradenie dátového typu premennej sa vykonáva znakom **:**. Deklarácia je ukončená bodkočiarkou. Definícia konštánt sa začína kľúčovým slovom **const**, ktoré nasleduje názov konštanty, znak priradenia **=** a hodnota. Pri definícii symbolickej konštanty sa dátový typ neuvádza.

*Príklad deklarácie jednej premennej dátového typu integer.*

```
var a : integer;
```

*Príklad deklarácie viacerých premenných dátového typu integer.*

```
var  
a : integer;  
b : integer;  
c : integer;
```

*Príklad zjednodušenej deklarácie viacerých premenných rovnakého dátového typu.*

```
var a,b,c : integer;
```

*Medzi kľúčovým slovom **var** a deklaráciami môžu byť biele znaky.*

```
var  
a,b,c : integer;
```

Hodnoty premenným je možné priraďovať dvoma spôsobmi, pričom každý spôsob je možné použiť len v platnej oblasti programu. Priradenie hodnoty prostredníctvom operátora **=** je možné len v sekcii **type** alebo **var**. V hlavnom programe a podprogramoch sa hodnota priraďuje zápisom **:=**. Keďže sa pre priradenie hodnôt v hlavnom programe a podprogramoch používa zápis **:=**, porovnávanie rovnosti hodnôt sa preto vykonáva znakom **=** a nie **==** ako v iných programovacích jazykoch.

## Aritmetické operátory

Aritmetické operátory umožňujú vykonávať matematické operácie s hodnotami typu integer a real. Ak sa použije operátor **div**, výsledok operácie musí byť uložený do premennej dátového typu integer. Ak sa použije operátor **/**, výsledok operácie musí byť uložený do premennej dátového typu real.

<i>Operátor</i>	<i>Význam</i>
+	sčítanie - sčíta dve čísla, $c=a+b$ .
-	odčítanie - odčíta druhé číslo od prvého, $c=a-b$ .
<b>div</b>	celočíselné delenie - delí prvé číslo druhým, $c=a/b$ .
/	delenie reálnych čísiel - delí prvé číslo druhým, $c=a/b$ .
*	násobenie - násobí čísla, $c=a*b$ .
<b>mod</b>	modulo (zvyšok po delení) - delí prvé číslo druhým a vracia zvyšok, $c=a\%b$ .
-	ako záporné znamienko hodnoty (unárne mínus), $-a$ .
+	ako kladné znamienko hodnoty (unárne plus), $+a$ .

**Príklad č.67.** *Delenie celých a reálnych čísiel.*

```
program AritmetickeOperatory;
var a,b,c : integer;
var d : real;
begin
  a:=10;
  b:=5;
  c:=a div b;
  d:=a/b;
  writeln(c);
  writeln(d:2:2);
  readln;
end.
```

**Obrázok:** Zobrazenie príkladu.

```
2
2.00
```

## Príkaz for to

Príkaz **for to** vykoná programový kód uvedený medzi kľúčovými slovami **begin** a **end** požadovaný počet krát. Počítadlo cyklu, ktoré zabezpečuje opakovanie, má stúpajúci charakter. Slovo **to** zvýši hodnotu o číslo 1. Hodnotu príkazu **for** nie je možné v tele meniť. Ak programový kód cyklu tvorí jednoriadkový zápis, nie je potrebné ho dávať medzi kľúčové slová **begin** a **end**. Premenná, ktorá tvorí počítadlo, môže byť použitá v programovom kóde cyklu **for to** len pre výpis (prípadne zápis do súboru) alebo priradenie inej premennej.

*Formát zápisu.*

```
for premenna:=hodnota to hodnota do
begin
    programový kód;
end;
```

**Príklad č.83.** Použitie príkazu **for to**.

```
program CyklusForTo;
var cislo : integer;
begin
    cislo:=1;
    for cislo:=1 to 5 do
        begin
            writeln('Opakovanie c.: ',cislo);
        end;
    readln;
end.
```

**Obrázok:** Zobrazenie príkladu.

```
Opakovanie c.: 1
Opakovanie c.: 2
Opakovanie c.: 3
Opakovanie c.: 4
Opakovanie c.: 5
```

**Príklad č.84.** Použitie príkazu **for to** bez kľúčových slov **begin** a **end**.

```
program CyklusForTo;
var cislo : integer;
begin
    cislo:=1;
    for cislo:=1 to 5 do writeln('Opakovanie c.: ',cislo);
    readln;
end.
```