

```

if podmienka then begin
    //príkazy ak podmienka platí;
end
else begin
    //príkazy ak podmienka neplatí;
end;

```

```

if podmienka then 1 príkaz
else 1 príkaz;

```

```

case x of
  1 : begin
    //príkazy, ktoré sa majú vykonať,
    ak je hodnota x rovná 1;
  end;
  2 : begin
    //príkazy, ktoré sa majú vykonať,
    ak je hodnota x rovná 2;
  end;
  3 : begin
    //príkazy, ktoré sa majú vykonať,
    ak je hodnota x rovná 3;
  end;
  4 : begin
    //príkazy, ktoré sa majú vykonať,
    ak je hodnota x rovná 4;
  end;
end;

```

```

case x of
  1 : begin
    //príkazy, ktoré sa majú vykonať,
    ak je hodnota x rovná 1;
  end;
  2 : begin
    //príkazy, ktoré sa majú vykonať,
    ak je hodnota x rovná 2;
  end;
  3 : begin
    //príkazy, ktoré sa majú vykonať,
    ak je hodnota x rovná 3;
  end
  else begin
    //príkazy, ktoré sa majú vykonať,
    ak je hodnota x iná;
  end;
end;

```

<pre> var o : integer; begin o:= 0; o:= o + 1; end; </pre>	<pre> var o : integer; begin o:= 0; inc(o); end; </pre>	<pre> var u : integer; begin u:= 0; u:= u - 1; end; </pre>	<pre> var u : integer; begin u:= 0; dec(u); end; </pre>
<pre> var o : integer; begin o:= 0; o:= o + 12; end; </pre>	<pre> var o : integer; begin o:= 0; inc(o,12); end; </pre>	<pre> var u : integer; begin u:= 0; u:= u - 12; end; </pre>	<pre> var u : integer; begin u:= 0; dec(u,12); end; </pre>

Typ integer

Ordinálny typ (každá hodnota má svojho predchodcu aj nasledovníka).

Každá celočíselná premenná typu `integer` je uložená v 2 B (16 b) a môže nadobúdať hodnoty v rozsahu $\langle -32\,768, 32\,768 \rangle$.

Typ interval

Je odvodený (vytvorený) z nejakého už existujúceho ordinálneho typu. V pamäti zaberá menej miesta.

DEKLARÁCIA:

```
type MojInterval = -1..10;  
var o : MojInterval;  
begin  
  o := 4;  
end;
```

- určíme minimálnu a maximálnu konštantu tohto nového typu,
- zápis `-1..10` znamená, že vytvárame podtyp celých čísel a tento nový typ obsahuje všetky konštanty z intervalu $\langle -1, 10 \rangle$,
 - ak by sme do premennej `o` priradili nejakú inú hodnotu (napr. 15), program spadne,
- zo svojho „nadradeného“ typu preberá všetky vlastnosti a operácie,
- je tiež ordinálny typ.

Ak máme vlastný typ interval s rozsahom $-10 \dots 5$ podľa princípu kódovania celých čísel sa pre hodnoty v intervale vyhradí 8 bitov na ukladanie hodnôt (1 bit pre kódovanie znamienka, 7 bitov pre čísla v rozsahu od -128 až 127).

Ak do premennej `MojTyp` nastavíme hodnotu, ktorá **nepatrí do nami definovaného intervalu**, ale táto hodnota **patrí do intervalu -128 až 127** (t.j. je z **intervalu hodnôt, ktoré môžeme uložiť do vyhradených bitov**) kompilátor nás na túto skutočnosť upozorní iba v správach v podobe **warningu**.

```
procedure TForm1.Button1Click(Sender: TObject);  
type MojTyp = -10..5;  
var a:MojTyp;  
begin  
  a:=12;  
  Image1.Canvas.TextOut(10,10,IntToStr(a));  
end;
```

Button1

12

Správy

Compile Project, Target: C:\Users\Miska\AppData\Local\Temp\project1.exe: Úspešné, Warnings: 1

▲ unit1.pas(44,6) Warning: range check error while evaluating constants (12 must be between -10 and 5)

Ak do premennej `MojTyp` nastavíme hodnotu, ktorá **nepatrí do nami definovaného intervalu, ani do intervalu -128 až 127** (t.j. je z **intervalu hodnôt, ktoré môžeme uložiť do vyhradených bitov**) kompilátor nás na túto skutočnosť upozorní v správach v podobe **warningu** a **hodnota premennej pretečie** – t.j. je v nej iné číslo.

```

procedure TForm1.Button1Click(Sender: TObject);
type MojTyp = -10..5;

var a:MojTyp;

begin
  a:=-200;
  Image1.Canvas.TextOut(10,10,IntToStr(a));
end;

```

Button1

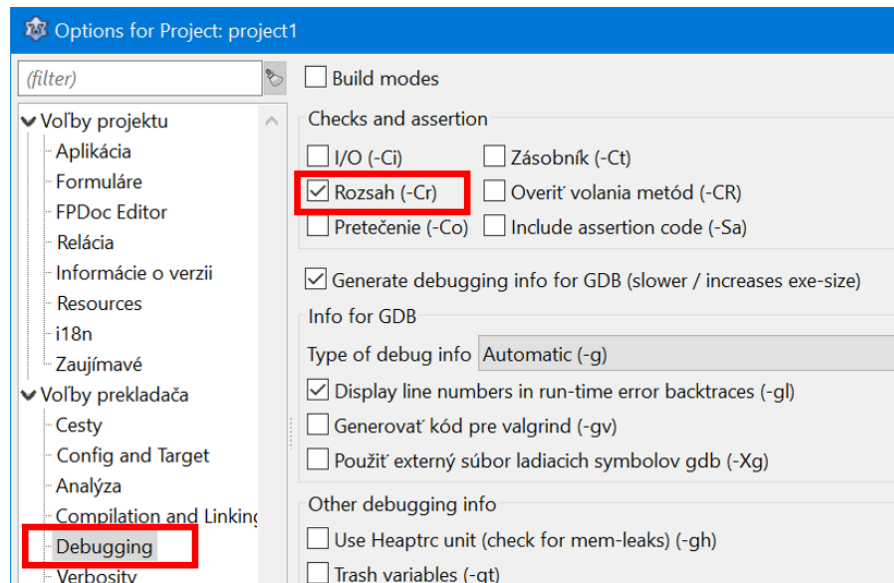
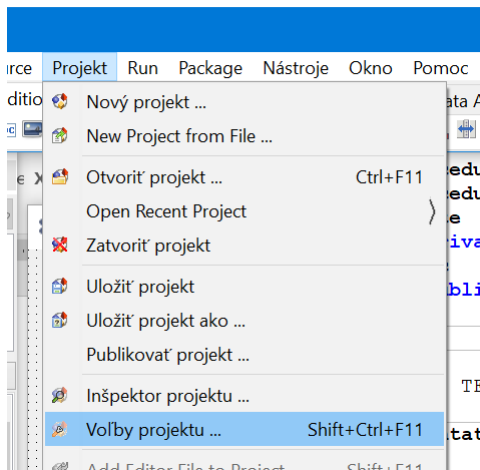
56

Správy

Compile Project, Target: C:\Users\Miska\AppData\Local\Temp\project1.exe: Úspešné, Warnings: 1

unit1.pas(41,6) Warning: range check error while evaluating constants (-200 must be between -10 and 5)

Aby sme nemuseli dávať pozor na hodnoty, ktoré do premenných vkladáme, vieme pre každý projekt nastaviť kontrolu rozsahu pre kompilátor. Ak do premennej nastavíme číslo, ktoré nie je v rozsahu program padne.



```

procedure TForm1.Button1Click(Sender: TObject);
type MojTyp = -10..5;

var a:MojTyp;

begin
  a:=-12;
  Image1.Canvas.TextOut(10,10,IntToStr(a));
end;

```

Typ real

Nie je ordinálny typ (preto preň nefungujú napr. `inc` a `dec`, nemôže sa použiť ani ako riadiaca premenná v cykle `for` ani v príkaze `case`).

Je nadradený nad celočíselný typ, t.j. celočíselná premenná sa automaticky konvertuje na reálny typ, keď je to treba, opačne to neplatí.

výraz	typ operandu A	typ operandu B	typ výsledku
A + B A - B A * B	integer	integer	integer
	integer	real	real
	real	integer	real
	real	real	real
A / B	integer	integer	real
	integer	real	
	real	integer	
	real	real	
A div B A mod B	integer	integer	integer

DEKLARÁCIA:

```
var cislo : real;  
begin  
    cislo:= 2.567;  
    cislo:= StrToFloat(Edit1.Text);  
    Image1.Canvas.TextOut(10,15,FloatToStr(cislo));  
end;
```

Naformátovanie výpisu desatinného čísla

Desatinné čísla majú zvyčajne veľký rozsah, ale obmedzenú presnosť

- rozsah typu `real` závisí od platformy,
- presnosť závisí od veľkosti čísla: `Single`, `Double`, `Comp`, `Extended` (zoradené od najmenej presných po najpresnejšie).

Tieto typy čísel majú „plávajúcu“ desatinnú čiarku (číslo je zapísané pomocou $\text{mantisa} \cdot 2^{\text{exponent}}$, kde mantisa aj exponent sú celé čísla, napr. číslo 0,1 má periodický dvojkový zápis $(0,0001100)_2$).

Občas dôjde k nekonečnému desatinnému zápisu hodnoty: $1,25 + 0,25 = 1,49999999999999$, preto hodnoty vieme vypísať na požadovaný (potrebný) počet desatinných miest:

```
Image1.Canvas.TextOut(10, 15, FloatToStrf(názovPremennej, ffnumber,  
presnost, početDesatinnýchMiest));
```

Pretypovanie real na integer

Aby sme z reálneho čísla dostali celé číslo, je treba toto konvertovať/zaokrúhliť na celé číslo. Na to slúžia 2 funkcie

trunc – výsledkom je celočíselná časť reálnej hodnoty

round – výsledkom je zaokrúhlená hodnota reálnej hodnoty

```
var a:integer;  
    o:real;  
o:=5.8;  
a:=trunc(o);    //a = 5  
a:=round(o);    //a = 6
```

Typ TCanvas – globálna premenná

```
var g: TCanvas;
```

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    g.Brush.Color:=clBlue;  
    g.Rectangle(10,20,30,50);  
    g.TextOut(10,30,'Naozaj to funguje.');
```

```
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
    g:=Image1.Canvas;  
end;
```

Konštrukcia with

```
with Image1.Canvas do begin  
    Brush.Color:=clBlue;  
    Rectangle(10,20,30,50);  
    TextOut(10,30,'Naozaj to funguje.');
```

```
end;
```

1. Vytvorte program, ktorý bude deliť dve čísla zadané (celé aj desatinné) v Edite.

$15 / 1,5 = 10$

delenie

15

1,5

$10 / 1,5 = 6,66666666666667$

delenie

10

1,5

2. Vytvorte program simulujúci palubný počítač auta. Po zadaní prejdenej vzdialenosti v km a doby jazdy v hodinách oznámi priemernú rýchlosť. Predpokladajte, že bol pohyb rovnomerný (na výpočet použite vzorec $v = s/t$). Rýchlosť zaokrúhlite na 3 desatinné miesta.

auto

10

1,5

Priemerná rýchlosť vášho auta bola 6,667 km/h

3. Zhotovte jednoduchú aplikáciu, ktorá bude prevádzať centimetre na metre, milimetre a decimetre (užívateľ zadá dĺžku v centimetroch a vyberie si jednotky, do ktorých sa má dĺžka premeniť (na otestovanie zadanej jednotky možno použiť príkaz: `if Edit1.Text='cm' then`)).

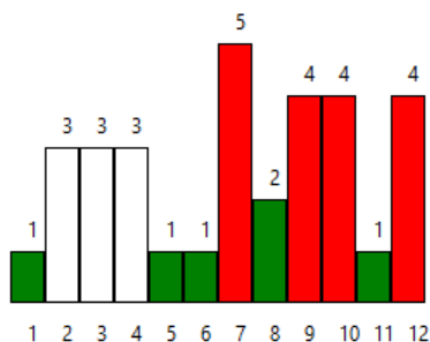
cm

1,5

mm

1,5 cm = 15 mm

4. Napíšte program, ktorý bude vymýšľať známky z písomiek pre žiakov (počet žiakov zadáme). Znamky vypíše a vykreslí stĺpcovým grafom (jednotky a dvojky sú zelené, trojky biele a štvorky a päťky sú červené). Tiež vypočítajte priemer známok (na 2 desatinné čísla) a počet jednotlivých známok v triede (na riešenie použite príkaz `case`, typ interval na známky).



priemer známok je 2,67

počet jednotiek je: 4, počet dvojok je: 1, počet trojok je: 3, počet štoriek je: 3, počet päťok je: 1

5. Adam dostával od otca pevné celočíselné vreckové N € každý týždeň (vieme ho zadať z Editu). Keďže mal dobré polročné vysvedčenie, od februára mu otec zvýšil vreckové každý týždeň o 20 centov. Napíšte program, ktorý vypočíta, koľko otec vyplatí synovi na vreckovom do konca školského roka, vypíše aj jednotlivé sumy, ktoré otec vyplatil.

```
12,2 14,4
12,4 14,6
12,6 14,8
12,8 15,0
13,0 15,2
13,2 15,4
13,4 15,6
13,6 15,8
13,8 16,0
14,0
14,2
```

Dostával vreckové 12 €, po zvýšení do konca roka mu bolo vyplatené: 282,0

6. Zostavte program, ktorý si vyžiada vek cestujúceho (typ na vek je interval od 0 do 100) a vypíše cenu cestovného lístka (cena základného lístka je 8 €). Deti a mládež do 26 rokov majú zľavu 20% z ceny. V prípade, že cestujúci má zľavu, program ešte pred samotnou cenou lístka vypíše text: „Máte nárok na zľavu“.

```
Vo veku 4 rokov máte nárok na zľavu.
Cena lístka je: 6,4 €
```

7. Napíšte program, ktorý zo zadanej prejdenej vzdialenosti vypočíta platbu taxikárovi, keď má stanovené nasledujúce sadzby:
- 0 – 19 km 0,8 € /km
 - 20 – 39 km 0,7 € /km
 - 40 – 59 km 0,65 € /km
 - 60 km a viac 0,6 € /km

Zadaná vzdialenosť môže byť iba celé číslo, ošetríte zadanie zápornej vzdialenosti (na riešenie použite príkaz case).

Viezol si sa 12 km, teraz zaplatíš 9,6 €

8. Na konci žiackej brigády (3 týždne) sa zamestnávateľ rozhodol odmeniť žiakov a k ich platu (2.5 € za hodinu) im pridal ešte odmenu. Žiakom, ktorí odpracovali viac ako 30 hodín, dal odmenu 10% z ich platu. Ostatní dostali odmenu 5% zo svojho platu. Vytvorte program, ktorému zadáme počet žiakov, vymyslí pre každého žiaka počet odpracovaných hodín, vyplatenú sumu a určí počet žiakov, ktorým bola vyplatená suma aspoň 100 €.

```
1. žiak 44 hodín = 110 €    aj s odmenou mu bolo vyplatené: 121
2. žiak 49 hodín = 122,5 €  aj s odmenou mu bolo vyplatené: 134,75
3. žiak 17 hodín = 42,5 €   aj s odmenou mu bolo vyplatené: 44,625
4. žiak 41 hodín = 102,5 €  aj s odmenou mu bolo vyplatené: 112,75
5. žiak 18 hodín = 45 €     aj s odmenou mu bolo vyplatené: 47,25
6. žiak 17 hodín = 42,5 €   aj s odmenou mu bolo vyplatené: 44,625
7. žiak 15 hodín = 37,5 €   aj s odmenou mu bolo vyplatené: 39,375
8. žiak 39 hodín = 97,5 €   aj s odmenou mu bolo vyplatené: 107,25
9. žiak 2 hodín = 5 €       aj s odmenou mu bolo vyplatené: 5,25
10. žiak 10 hodín = 25 €    aj s odmenou mu bolo vyplatené: 26,25
11. žiak 13 hodín = 32,5 €  aj s odmenou mu bolo vyplatené: 34,125
12. žiak 11 hodín = 27,5 €  aj s odmenou mu bolo vyplatené: 28,875
```

počet žiakov, ktorým bolo vyplatené viac ako 100€ je: 4

9. Napíšte program, ktorý bude stupne Celzia premieňať na stupne Kelvina a stupne Fahrenheita. Program urobte interaktívny, hodnoty sa budú prepočítavať pri každej zmene hodnoty Editu.

$$12^{\circ}C = 12 + 273,15 K$$
$$F = \frac{9 * ^{\circ}C}{5} + 32$$

° Celzia	Kelvin	° Fahrenheita
1	274,15	33,8