

POČÍTAČE A PROGRAMOVÁNÍ

Struktura programu, Proměnné,
Konstanty, Datové typy,
Podmínka If-then

Miroslav Vavroušek

Struktura programu v jazyce PASCAL

- **Hlavička programu**

- Obsahuje klíčové slovo program, název a středník

`program SectiDveCisla;`

- **Deklarační část**

- Popisuje co se bude v programu využívat

- **Implementační část**

- Popisuje co bude program dělat

Struktura programu v jazyce PASCAL

```
program SectiDveCisla;  
var  
  a, b, c: Integer;  
begin  
  ReadLN( a, b);  
  
  c:= a + b;  
  
  WriteLN(c);  
  ReadLN;  
end.
```

Hlavička programu

Deklarační část

Implementační část

Proměnná

- Proměnná uchovává informaci potřebnou pro práci programu
 - Hodnota proměnné se při vykonání programu může měnit
- Má ve svém oboru platnosti unikátní jméno
 - Případně, musí být dodržena pravidla překrývání jmen
- Každá proměnná je nějakého konkrétního datového typu

Proměnná

Deklarace

- Proměnnou, kterou chci v programu využít musím deklarovat (v deklarační části programu)

- **Deklarace**

- Deklarace proměnné obsahuje:

- Název proměnné 
 - Definiční operátor 
 - Datový typ 
 - Středník 

- Deklarovat lze i více proměnných najednou

- Názvy se oddělují se čárkou `A, B, C: Integer;`

Proměnná

Přiřazení hodnoty a inicializace proměnné

- Hodnotu proměnné mohou nastavovat pomocí příkazů (v implementační části programu)

- **Přiřazení hodnoty**

- Přiřazení hodnoty proměnné obsahuje:

- Název proměnné → C := A + B;
 - Přiřazovací operátor →
 - Vyras/hodnota →
 - Středník →
-

- Inicializace proměnné → A := 0; A := A + 1;

- Nastavení počáteční hodnoty pro další výpočty

Proměnná

Práce s hodnotou proměnné

- S hodnotu proměnné mohou pracovat pomocí jména proměnné (v implem. části programu)
- **Využití hodnoty proměnné**
 - Použiji název proměnné ve výrazu:

$C := A + B;$ Ve výrazu využívám hodnoty proměnných A a B

- Před prvním použitím hodnoty proměnné musí být hodnota proměnné definována (inicializována)

$A := 0;$ $A := A + 1;$


Konstanty

- Konstanta uchovává nějakou informaci potřebnou pro práci programu
 - Hodnotu konstanty při vykonání programu nelze měnit
 - Například: gravitační zrychlení, π , nastavení programu
- Má ve svém oboru platnosti unikátní jméno
 - Případně, musí být dodrženy pravidla překrývání jmen
- Každá proměnná je nějakého konkrétního datového typu

Konstanty

- **Přímé konstanty**

- Hodnoty zapsané přímo ve výrazu

 B := 2 * A;

- **Nepřímé konstanty**

- Pojmenované konstanty

- Deklarace nepřímé konstanty (v deklarační části)

const

MINIMUM = 1;

MAXIMUM = 500;

- Využití hodnot konstant (v implementační části)

A := MINIMUM + 20;

Datové typy v jazyku PASCAL

- Datový typ definuje:
 - Jaké data může proměnná uchovávat
 - Jaké operace lze s proměnou provádět
- Standardní datové typy (jednoduché)
 - Definované přímo v programovacím jazyku
- Uživatelské datové typy (složené)
 - Definované programátorem pomocí základních datových typů

Datové typy v jazyku PASCAL

- **Standardní datové typy (jednoduché)**

- **Celočíselné typy**

- Integer, LongInteger

- **Reálné datové typy**

- Real, Entended

- **Logické**

- Boolean

- **Znakové**

- Char, String

- **Ordinální datový typ**

- Je známé pořadí hodnot

- Například Integer, Char atd.

Datové typy v jazyku PASCAL

Identifikátor	Charakteristika	Rozsah	Ukázka deklarace a iniciace
Boolean	Booleovská hodnota	Dva stavy. True a False. (Pravda a nepravda, 1 a 0)	Deklarace: Ok: Boolean; Iniciace: Ok:= True;
Integer	Celé číslo	-32768 až 32767	Deklarace: X: Integer; Iniciace: X:= 5;
Real	Reálné číslo	$2.9 \cdot 10^{-39}$ až $1.7 \cdot 10^{38}$	Deklarace: X: Real; Iniciace: X:= 2.92;
String	Text	ASCII znaky (nemůžete používat české znaky)	Deklarace: Text: String; Iniciace: Text:= 'Ahoj';
Char	Jeden znak textu	ASCII znak	Deklarace: Znak: Char; Iniciace: Znak:= 'a';

Operátory v jazyku PASCAL

Aritmetické operátory

Identifikátor	Operace	Ukázka použití
+	Sčítání. Datový typ výsledku je určen datovými typy operandů. Pokud je alespoň jeden s operandu typu Real výsledek je také typu Real.	X := 5 + 6; //X = 11 X := 5.2 + 6; //X = 11.2 C := A + B;
-	Odčítání. Datový typ viz „+“.	X := 5 - 6; //X = -1 C := A - B;
*	Násobení. Datový typ viz „+“.	X := 5 * 6; //X = 30 C := A * B;
/	Dělení. Výsledek je typu Real.	X := 5 / 6; //X = 0.83 C := A / B;
div	Celočíselné dělení. Výsledkem je cela část čísla po dělení. Výsledek je typu Integer.	X := 47 div 6; //X = 7 X := -9 div 4; //X = -2 C := A div B;
mod	Zbytek po celočíselném dělení. Často nazýván také Modulo.	X := 47 mod 6; //X = 5 X := -9 mod 4; //X = -1 C := A mod B;

Operátory v jazyku PASCAL

Booleovské operátory

Identifikátor	Operace	Ukázka použití
not	Negace	X: = not True; //X = False B:= not A;
and	Logicky součin (oba musí platit)	X: = True and False; //X = False C:= A and B;
or	Logicky součet (alespoň jeden musí platit)	X: = True or False; //X = True C:= A or B;

Operátory v jazyku PASCAL

Porovnávací operátory

Identifikátor	Operace	Ukázka použití
>	Větší	X: = 5 > 6; //X = False C:= A > B;
<	Menší	X: = 5 < 6; //X = True C:= A < B;
>=	Větší a rovno	X: = 5 >= 6; //X = False C:= A >= B;
<=	Menší a rovno	X: = 5 <= 6; //X = True C:= A <= B;
=	Rovná se	X: = 5 = 6; //X = False C:= A = B;
<>	Nerovná se	X: = 5 <> 6; //X = True C:= A <> B;

Podmíněný příkaz

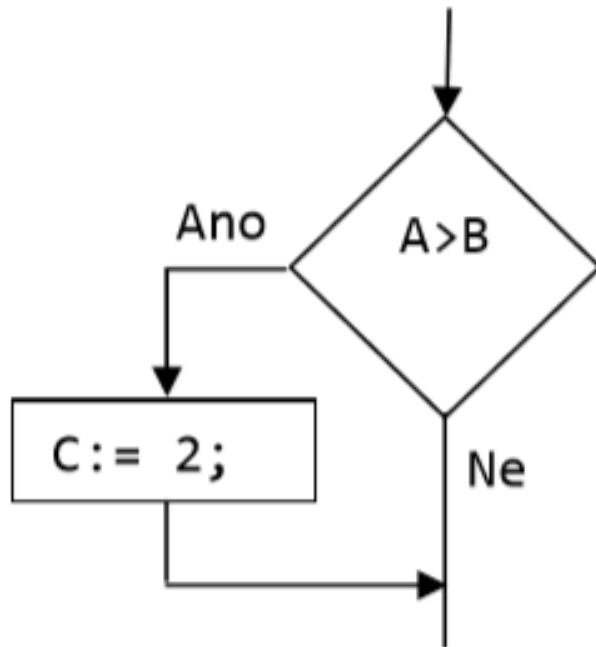
- Umožňuje větvení programu
- Umožňuje vykonávat příkazy při splnění podmínky (a vykonat jiné při nesplnění podmínky – úplná p.)
- **Podmínka If**
 - Ohraničené příkazy jsou vykonány při splnění podmínky
- **Úplná podmínka If – else**
 - Ohraničené příkazy jsou vykonány při splnění podmínky
 - Při nesplnění podmínky jsou vykonány příkazy ve větvi else

Podmíněný příkaz

Podmínka If

Podmínka if

```
if Podmínka then  
begin  
    // Pokud je podmínka splněna  
end;
```



{Pokud je A větší než B do C vlož 2}

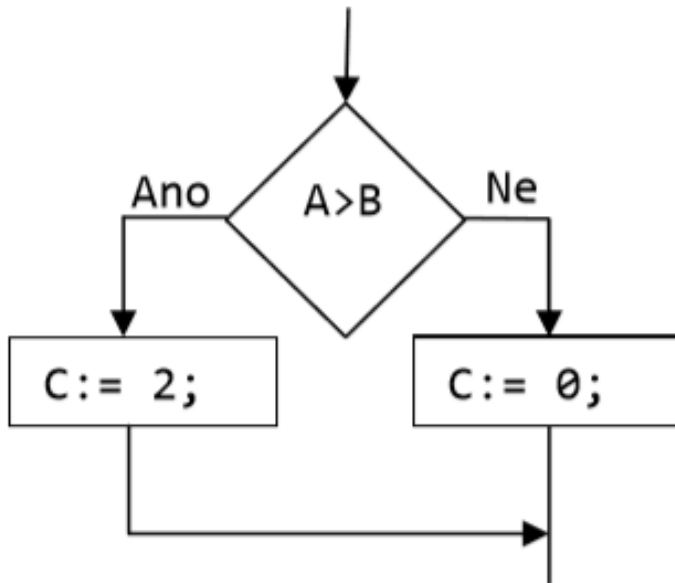
```
if A > B then  
begin  
    C := 2;  
end;
```

Podmíněný příkaz

Úplná podmínka If – else

Úplná podmínka if-else

```
if Podminka then  
begin  
    // Pokud je podmínka splněna  
end else  
begin  
    // Pokud je podmínka nesplněna  
end;
```



```
if A > B then  
begin  
    C := 2; {Pokud je A větší  
            než B do C vlož 2}  
end else  
begin  
    C := 0; {Pokud A není větší  
            než B do C vlož 0}  
end;
```