



MINISTERSTVO ŠKOLSTVÍ,  
MLÁDEŽE A TĚLOVÝCHOVY



**OP Vzdělávání  
pro konkurenceschopnost**

INVESTICE  
DO ROZVOJE  
VZDĚLÁVÁNÍ

**Střední průmyslová škola a Vyšší odborná škola technická Brno, Sokolská 1**

**Šablona: Inovace a zkvalitnění výuky prostřednictvím ICT**

**Název: Základy programování a algoritmizace úloh**

**Téma: Jednoduché příkazy jazyka Pascal**

**Autor: Ing. Hodál Jaroslav, Ph.D.**

**Číslo: VY\_32\_INOVACE\_25-07**

**Anotace:** *Materiál popisuje jednoduché příkazy tvořící základní sadu pro vytváření jednoduchých programů. Materiál je určen pro 3. a 4. ročník oboru strojírenství a technické lyceum. Vytvořeno: červenec 2012.*

# 7. Jednoduché příkazy jazyka Pascal

- jednoduché příkazy provádí jednu jednoduchou činnost a zapisují se pomocí jednoslovného názvu (viz 25-02 – Příkazy v algoritmech)
- názvy jednoduchých příkazů v jazyce Pascal obvykle vycházejí z anglického slova označujícího činnost, kterou má daný příkaz vykonávat (např. `write` umí vypsát informaci na obrazovku počítače)
- výjimkou je přiřazovací příkaz (viz 25-02)

# Volání jednoduchých příkazů

- pokud chceme, aby program provedl některý příkaz, musíme jej ve zdrojovém kódu tzv. „zavolat“
- volání jednoduchých příkazů se provádí velmi snadno pouhým zapsáním jejich názvu do zdrojového kódu.
- některé příkazy mohou potřebovat tzv. **parametry**.

# Parametry jednoduchých příkazů

- parametry představují dodatečné informace, které mohou některé příkazy potřebovat pro upřesnění toho, co mají provádět.
- parametry se obvykle uvádějí za název příkazu do kulatých závorek.

```
write ( 'Ahoj' );
```

V uvedeném příkladu požadujeme vypsání informace příkazem `write`. Parametrem je zde textový řetězec `Ahoj`, což je text, který se nakonec objeví na obrazovce počítače.

# Parametry jednoduchých příkazů

- parametrem může být i proměnná, jejíž obsah je v době psaní programu neznámý

```
write(x);
```

V tomto příkladu požadujeme bude na obrazovku vypsána aktuální informace obsažená v proměnné `x`.

- parametrů může být v některých příkazech i více
- v takovém případě se oddělují čárkami

```
write('Výsledek je ', x);
```

Tento příkaz vypíše nejprve text `Výsledek je` a pak přidá hodnotu obsaženou v proměnné `x`.

# Příkazy pro komunikaci - výstup

`write(x);`

- vypíše informaci na obrazovku
- `x` musí být proměnná nebo konstanta typu textový řetězec, textový znak, číslo (celé i reálné), logická hodnota

`writeln(x);`

- vypíše informaci na obrazovku a po vypsání odřádkuje
- `x` musí být proměnná nebo konstanta typu textový řetězec, textový znak, číslo (celé i reálné), logická hodnota
- příkaz lze použít i bez parametru – pouze odřádkuje
- `writeln; / writeln();`

# Příkazy pro komunikaci - vstup

`readln(x);`

- načte informaci z klávesnice
- `x` musí být proměnná typu textový řetězec, textový znak, číslo (celé i reálné)
- při vykonávání tohoto příkazu program pozastaví svůj chod a čeká na zadání informace od uživatele
- informace je načtena a uložena do proměnné po stisknutí klávesy ENTER
- pokud zadaná informace nebude svým typem odpovídat proměnné uvedené v parametru, dojde k chybě a program se předčasně ukončí
- příkaz lze použít i bez parametru; program pozastaví svůj běh do stisku klávesy ENTER

# Příkazy pro komunikaci

- příkaz `readln` je vhodné doplnit o předchozí výpis informací o tom, co po uživateli chceme

```
writeln('Zadejte celé číslo');  
readln(zadani);
```

- všechny příkazy pro komunikaci se používají také při práci se soubory
- pro čtení ze souborů se používá i příkaz `read`, který je však pro čtení z klávesnice nevhodný
- více o čtení dat ze souborů je uvedeno v 26-07, 26-08 a 26-09

# Matematické příkazy

`abs (x) ;`

- vypočte absolutní hodnotu z hodnoty zadané
- `x` musí být konstanta nebo proměnná typu číslo (celé i reálné)
- výsledkem je číslo typu odpovídajícího vstupní informaci
- příkaz má výsledek, který je nutno vhodným způsobem zpracovat
- příklady použití

`y := abs (x) ;`

uložení výsledku do proměnné `y`

je možné s ním dále pracovat

`writeln (abs (x) ) ;`

přímý výpis výsledku na obrazovku

nelze s ním dále pracovat

# Matematické příkazy

`round(x) ;`

- zaokrouhlí zadanou hodnotu na celé číslo
- `x` musí být konstanta nebo proměnná typu reálné číslo
- výsledkem je celé číslo
- příkaz má výsledek, který je nutno vhodným způsobem zpracovat
- příklady použití

`y := round(x) ;`

uložení výsledku do proměnné `y`

je možné s ním dále pracovat

`writeln(round(x)) ;`

přímý výpis výsledku na obrazovku

nelze s ním dále pracovat

# Matematické příkazy

`trunc(x) ;`

- odstraní z reálného čísla desetinnou část
- `x` musí být konstanta nebo proměnná typu reálné číslo
- výsledkem je celé číslo
- příkaz má výsledek, který je nutno vhodným způsobem zpracovat
- příklady použití

`y := trunc(x) ;`

uložení výsledku do proměnné `y`

je možné s ním dále pracovat

`writeln(trunc(x)) ;`

přímý výpis výsledku na obrazovku

nelze s ním dále pracovat

# Matematické příkazy

`sqr(x);`

- vypočte druhou mocninu zadaného čísla
- `x` musí být konstanta nebo proměnná typu číslo (celé i reálné)
- výsledkem je číslo typu odpovídajícího vstupní informaci
- příkaz má výsledek, který je nutno vhodným způsobem zpracovat
- příklady použití

`y := sqr(x);`

uložení výsledku do proměnné `y`

je možné s ním dále pracovat

`writeln(sqr(x));`

přímý výpis výsledku na obrazovku

nelze s ním dále pracovat

# Matematické příkazy

`sqrt(x);`

- vypočte druhou odmocninu zadaného čísla
- `x` musí být konstanta nebo proměnná typu číslo (celé i reálné)
- výsledkem je vždy reálné číslo
- příkaz má výsledek, který je nutno vhodným způsobem zpracovat
- příklady použití

`y := sqrt(x);`

uložení výsledku do proměnné `y`

je možné s ním dále pracovat

`writeln(sqrt(x));`

přímý výpis výsledku na obrazovku

nelze s ním dále pracovat

# Matematické příkazy

`sin(x);`

- vypočte sinus zadaného úhlu
- `x` musí být konstanta nebo proměnná typu číslo (celé i reálné) představující úhel v úhlové míře (radiánech)
- výsledkem je vždy reálné číslo
- příkaz má výsledek, který je nutno vhodným způsobem zpracovat
- příklady použití

`y := sin(x);`

uložení výsledku do proměnné `y`

je možné s ním dále pracovat

`writeln(sin(x));`

přímý výpis výsledku na obrazovku

nelze s ním dále pracovat

# Matematické příkazy

`cos (x) ;`

- vypočte kosinus zadaného úhlu
- `x` musí být konstanta nebo proměnná typu číslo (celé i reálné) představující úhel v úhlové míře (radiánech)
- výsledkem je vždy reálné číslo
- příkaz má výsledek, který je nutno vhodným způsobem zpracovat
- příklady použití

`y := cos (x) ;`

uložení výsledku do proměnné `y`

je možné s ním dále pracovat

`writeln (cos (x) ) ;`

přímý výpis výsledku na obrazovku

nelze s ním dále pracovat

# Další jednoduché příkazy

`chr(x) ;`

- vrací znak, jehož pořadové číslo odpovídá číslu zadanému
- `x` musí být konstanta nebo proměnná typu celé číslo
- výsledkem je znak
- příkaz má výsledek, který je nutno vhodným způsobem zpracovat
- příklady použití

`y := chr(x) ;`

uložení výsledku do proměnné `y`

je možné s ním dále pracovat

`writeln(chr(x)) ;`

přímý výpis výsledku na obrazovku

nelze s ním dále pracovat

# Další jednoduché příkazy

`ord(x) ;`

- vrací pořadové číslo zadané hodnoty
- `x` musí být konstanta nebo proměnná libovolného ordinálního datového typu (např. znak, logický, ...)
- výsledkem je celé číslo
- příkaz má výsledek, který je nutno vhodným způsobem zpracovat
- příklady použití

`y := ord(x) ;`

uložení výsledku do proměnné `y`

je možné s ním dále pracovat

`writeln(ord(x)) ;`

přímý výpis výsledku na obrazovku

nelze s ním dále pracovat

# Další jednoduché příkazy

`pred(x) ;`

- vrací hodnotu pořadově předcházející hodnotě zadané
- `x` musí být konstanta nebo proměnná libovolného ordinálního datového typu (např. znak, logický, ...)
- výsledkem je hodnota stejného typu jako zadaná
- příkaz má výsledek, který je nutno vhodným způsobem zpracovat
- příklady použití

`y := pred(x) ;`

uložení výsledku do proměnné `y`

je možné s ním dále pracovat

`writeln(pred(x)) ;`

přímý výpis výsledku na obrazovku

nelze s ním dále pracovat

# Další jednoduché příkazy

`succ(x) ;`

- vrací hodnotu pořadově následující po hodnotě zadané
- `x` musí být konstanta nebo proměnná libovolného ordinálního datového typu (např. znak, logický, ...)
- výsledkem je hodnota stejného typu jako zadaná
- příkaz má výsledek, který je nutno vhodným způsobem zpracovat
- příklady použití

`y := succ(x) ;`

uložení výsledku do proměnné `y`

je možné s ním dále pracovat

`writeln(succ(x)) ;`

přímý výpis výsledku na obrazovku

nelze s ním dále pracovat

# Další jednoduché příkazy

`length(x) ;`

- vrací délku textového řetězce (počet jeho znaků)
- `x` musí být konstanta nebo proměnná typu textový řetězec
- výsledkem je celé číslo
- příkaz má výsledek, který je nutno vhodným způsobem zpracovat
- příklady použití

```
y := length(x) ;
```

uložení výsledku do proměnné `y`

je možné s ním dále pracovat

```
writeln(length(x)) ;
```

přímý výpis výsledku na obrazovku

nelze s ním dále pracovat

# Další jednoduché příkazy

`odd (x) ;`

- vrací informaci o lichosti či sudosti zadané hodnoty
- `x` musí být konstanta nebo proměnná typu celé číslo
- výsledkem je logická hodnota
- příkaz má výsledek, který je nutno vhodným způsobem zpracovat
- příklady použití

`y := odd (x) ;`

uložení výsledku do proměnné `y`

je možné s ním dále pracovat

`writeln (odd (x) ) ;`

přímý výpis výsledku na obrazovku

nelze s ním dále pracovat