

# Algoritmizace a programování

## Řídicí struktury jazyka Java

Příkazy cyklu

Příkaz `for`

Návěští

Příkazy `break`, `continue`

# Struktura programu v jazyce Java

```
package pkgnic;  
  
public class Main {  
  
    public static void main(String[] args) {  
  
        // příkazy  
        // sekvenčně se provedou po spuštění programu  
        // ukončením metody main program končí  
  
        // načtení vstupních hodnot  
        // výpočet výsledků  
        // výpis výsledků  
    }  
}
```

# Příkazy jazyka Java

- Kód metod – (metoda `main`) – příkazy
- Příkazy jsou v kódu ukončeny středníkem
- Deklarace proměnných – příkazy, které proměnné přidělí jméno, paměť, hodnotu
- Výraz ukončený středníkem je příkaz – příkazy přiřazení
- Volání metody ukončené středníkem je příkaz
- Logické příkazy `if` a `if-then`
- Příkazy cyklu `while` a `do-while`
- Další příkazy
  - Příkaz cyklu `for`
  - Návěští a příkazy `break` a `continue`
  - Příkaz `switch` pro mnohonásobné větvení programu
  - Příkaz `return`
- Jazyk Java nemá speciální požadavky na způsob zápisu kódu – jednotlivé syntaktické elementy musí být jednoznačně analyzovatelné, jediný syntaktický element (klíčové slovo, identifikátor, konstantní hodnota, symbol) nesmí být přerušen prázdným znakem (mezera, tabulátor, konec řádku)

# Příkaz cyklu `for`

- Syntaxe příkazu (způsob zápisu)

```
for (<výraz1>; <výraz2>; <výraz3>) <prikaz>;
```

```
for (<výraz1>; <výraz2>; <výraz3>) {  
    <prikaz1>; <prikaz2>; ... ; <prikazN>;  
}
```

- <výraz1> – libovolný výraz jazyka Java
- <výraz2> – logický výraz jazyka Java – podmínka pro pokračování
- <výraz3> – libovolný výraz jazyka
- <prikaz> – libovolný příkaz jazyka, může být nahrazen blokem příkazů

- Obecněji

```
for (<ini. cast>; <podminka>; <iter cast>) <telo cyklu>
```

- V inicializační a iterační části může být více výrazů navzájem oddělených čárkami
- Používáme-li řídicí proměnnou cyklu, potom do inicializační části umístíme její počáteční přiřazení a často i deklaraci (deklaraci s přiřazením hodnoty)
- Libovolná z částí může být prázdná

# Příkaz cyklu `for`

- Sémantika (význam, způsob vykonání)
  1. Vyhodnotí se výraz `<výraz1>`.
  2. Vyhodnotí se výraz `<výraz2>`.
  3. Pokud má výraz `<výraz2>` po vyhodnocení hodnotu `false`, příkaz cyklu se ukončí.
  4. Pokud má výraz `<výraz2>` po vyhodnocení hodnotu `true`,
    - a) provede se příkaz v těle cyklu (popřípadě blok příkazů),
    - b) vyhodnotí se výraz `<výraz3>`,
    - c) přejde se opakovaně k bodu 2

# Příkaz cyklu for

- Příklady použití:

```
int i;
for ( i = 1; i <= 12 ; i++ ){
    // telo cyklu se provede 12x v jednotlivych krocich
    // promenna I nabyva hodnot 1, 2, 3 ... 12
    System.out.println(i);
}
```

```
int n, f;
n = sc.nextInt();
f = 1;
for (int i = 2; i <= n; i++) f = f * i;
```

```
int c, s;
s = 0;
for (c = sc.nextInt(); c > 0; c = sc.nextInt()) {
    s = s + c;
}
```

# Návěští

- Návěští je libovolný identifikátor, který je zakončen dvojtečkou
- Příslušný identifikátor musí být v daném bloku unikátní
- Návěští používáme v souvislosti s příkazy `break` a `continue`
- Za návěštím může následovat pouze příkaz cyklu nebo blok začínající levou složenou závorkou
- Pokud umístíme návěští před jiný příkaz, překladač hlásí chybu pouze v případě, že na takovéto návěští provádíme skok (viz dále)

# Příkaz `break`

## ■ Příkaz `break` ukončí

- vykonávání cyklu
- vykonávání bloku příkazů
- vykonávání příkazu `switch`

## ■ Použití I

`break`;

- Ukončí vykonávání **nejvnitřnějšího** cyklu nebo bloku příkazů nebo příkazu `switch`

## ■ Použití II

`break <navesti>;`

- Ukončí vykonávání cyklu nebo bloku příkazů, jehož začátek je označen příslušným návěštím

# Příkaz `continue`

- Příkaz `continue` ukončí aktuální vykonávání těla cyklu, běh programu pokračuje poté dalším krokem cyklu

- Použití I

`continue;`

- Ukončí vykonávání **nejvnitřnějšího** cyklu nebo bloku příkazů nebo příkazu `switch`

- Použití II

`continue <navesti>;`

- Ukončí vykonávání aktuálního kroku cyklu, jehož začátek je označen příslušným návěštím

# Úlohy

- Výpočet faktoriálu
- Výpočet celočíselné mocniny
- Výpis dělitelů zadaného kladného celého čísla
- Zjištění největšího společného dělitele dvou zadaných celých kladných čísel

# Sekvenční zpracování posloupnosti čísel

- Načtení posloupnosti – 2 způsoby
  - Zadání počtem, následně vlastní hodnoty
  - Zadávání posloupnosti ukončené zadání konkrétní hodnoty, které není součástí zpracovávané posloupnosti
- Součet zadaných hodnot
- Součin
- Počet prvků posloupnosti splňujících zadanou podmínku
- Zjištění, zda posloupnost obsahuje konkrétní hodnotu, popřípadě místo jejího prvního nebo posledního výskytu, vyhledat výskyt hodnoty, vyhledat výskyt další hodnoty
- Zjištění minimální, maximální hodnoty
- Zjištění pořadí minimální, maximální hodnoty
- Zjištění vlastnosti celé posloupnosti