

# Algoritmizace a programování

## Řídicí struktury jazyka Java

Struktura programu

Příkazy jazyka

Blok příkazů

Logické příkazy

Ternární logický operátor

# 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
- Další příkazy
  - Logické příkazy pro větvení programu
  - Příkazy cyklu pro opakované provádění sady operací
  - ...
- 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)

## Blok příkazů

- Při zápisu jednotlivých řídicích konstrukcí, lze na některých místech kódu zapsat pouze jediný příkaz jazyka Java – například v těle cyklu. Pokud na daném místě chceme provádět více akcí – více příkazů jejichž sekvence se má postupně vykonat je nutné na daném místě použít **blok příkazů**
- Základní konstrukce bloku příkazů je tvořena složenými závorkami, uvnitř složených závorek může být libovolné množství příkazů, příkazy ukončené středníkem

```
{ <prikaz1>; <prikaz2>; ... <prikazN>; }
```

- Blok příkazů nesmí být ukončena středníkem
- Tělo metody (používáme zatím pouze metodu `main`) je tvořeno blokem příkazů – viz struktura programu
- Deklarace třídy je tvořena blokem, ve kterém jsou deklarace členských proměnných třídy a metod třídy – viz struktura programu

# Logické příkazy

- Logické příkazy řídí vykonávání kódu na základě hodnoty logického výrazu
- Neúplná podmínka – příkaz `if`
- Úplná podmínka – příkaz `if-else`

# Příkaz `if`

- Slouží k podmíněnému vykonání příkazu nebo bloku příkazů
- Konstrukce (syntaxe) příkazu

```
if ( <podminka> ) <prikazT> ;
```

- `if` – klíčové slovo jazyka Java
  - `<podminka>` – výraz, který po vyhodnocení má hodnotu logického typu (`true`, `false`)
  - `<prikazT>` – libovolný příkaz jazyka Java ukončený středníkem, pokud na místě příkazu stojí blok příkazů, potom za tímto blokem středník být nesmí
- Význam (sémantika) příkazu, způsob vykonávání příkazu
    - Při provádění příkazu se vyhodnotí nejdříve výraz `<podminka>`. Pokud má výraz po vyhodnocení hodnotu `true` potom se provede příkaz `<prikazT>` za podmínkou v opačném případě je příkaz `if` ukončen a pokračuje se prováděním dalšího příkazu za příkazem `if`.

# Příkaz `if-else`

- V závislosti na hodnotě logického výrazu větví kódu programu do dvou větví
- Konstrukce (syntaxe) příkazu

```
if ( <podminka> ) <prikazT>; else <prikazF>;
```

- `if` – klíčové slovo jazyka Java
  - <podminka> – výraz, který po vyhodnocení má hodnotu logického typu (`true`, `false`)
  - <prikazT> , <prikazF> – libovolné příkazy jazyka Java ukončené středníkem pokud na místě příkazu stojí blok příkazů, potom za tímto blokem středník být nesmí
- Význam (sémantika) příkazu, způsob vykonávání příkazu
    - Při provádění příkazu se vyhodnotí nejdříve výraz <podminka>. Pokud má výraz po vyhodnocení hodnotu `true` potom se provede příkaz za podmínkou v opačném případě se provede příkaz <prikazF> za klíčovým slovem `else` . Po provedení příkazu v jedné z větví je příkaz `if` ukončen a pokračuje se prováděním dalšího příkazu za příkazem `if`

# Podmíněný výraz

- Jazyk Java obsahuje logický ternární operátor – `? :`
- Ternární operátor má podobný význam jako úplná podmínka `if-else`
- Ternární operace má tři operandy – vlastní operátor je tvořen dvěma částmi – `? a :`
- Syntaxe – způsob zápisu podmíněného výrazu

`<podminka> ? <vyrazT> : <vyrazF>`

- `? :` – znaky operátoru
  - `<podminka>` – výraz, který má po vyhodnocení hodnotu logického typu
  - `<vyrazT>`, `<vyrazF>` – výrazy jazyka Java
- 
- Sémantika – význam, způsob interpretace
    - Při vyhodnocování je nejdříve vyhodnocen výraz `<podminka>`
    - Pokud tento výraz má hodnoty `true`, vyhodnocuje se dále výraz `<vyrazT>` uvedený za symbolem `?` a jeho hodnota určuje výslednou hodnotu celého podmíněného výrazu
    - Pokud má výraz po vyhodnocení hodnotu `false`, vyhodnocuje se výraz `<vyrazF>` uvedený za symbolem `:` a jeho hodnota určuje výslednou hodnotu celého podmíněného výrazu

# Úplná podmínka a podmíněný výraz

- **Úplná podmínka**
  - Jedná se o **příkaz**, v jednotlivých větvích jsou příkazy, které mohou být zastoupeny blokem příkazů
- **Podmíněný výraz**
  - Jedná se **výraz** – po vyhodnocení má hodnotu konkrétního typu, tuto hodnotu je možné použít ve složitějším výrazu, přiřadit proměnné, vypsát

```
package ja003maximumzedvou;

import java.util.Scanner;

public class Max1 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a, b;

        System.out.println("Zadej prvni cislo ");
        a = sc.nextInt();
        System.out.println("Zadej druhe cislo ");
        b = sc.nextInt();
        System.out.println("Vetsi ze zadanych cisel");
        if (a > b)
            System.out.println(a);
        else
            System.out.println(b);
    }
}
```

```
package ja003maximumzedvou;

import java.util.Scanner;

public class Max2 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a, b;
        int max;

        System.out.println("Zadej prvni cislo ");
        a = sc.nextInt();
        System.out.println("Zadej druhe cislo ");
        b = sc.nextInt();

        max = a;
        if (b > a) max = b;

        System.out.println("Vetsi ze zadanych cisel " + max);
    }
}
```

```
package ja003maximumzedvou;

import java.util.Scanner;

public class Max3 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a, b;
        int max;

        System.out.println("Zadej prvni cislo ");
        a = sc.nextInt();
        System.out.println("Zadej druhe cislo ");
        b = sc.nextInt();

        max = (a > b)? a : b ;
        System.out.println("Vetsi ze zadanych cisel " + max);
    }
}
```

```
package ja003maximumzedvou;
```

```
import java.util.Scanner;
```

```
public class Max4 {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        int a, b;
```

```
        System.out.println("Zadej prvni cislo ");
```

```
        a = sc.nextInt();
```

```
        System.out.println("Zadej druhe cislo ");
```

```
        b = sc.nextInt();
```

```
        System.out.println("Vetsi ze zadanych cisel " + ((a > b)? a : b ));
```

```
    }
```

```
}
```

# Kombinování úplných a neúplných logických příkazů

```
if (<podminka1>
    if (<podminka2>) <prikaz1>;
else <prikaz2>;
```

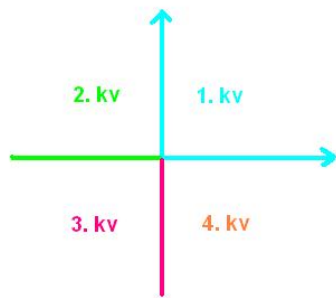
- Příkaz <prikaz2> se provede v případě, že podmínka <podminka1> má hodnotu true a podmínka <podminka2> hodnotu false, větev else se přiřazuje nejbližšímu if.
- Pokud si chceme vynutit jinou interpretaci – provedení příkazu <prikaz2> pro <podminka1>==false, potom je nutné použít blok příkazů následovně:

```
if (<podminka1>){
    if (<podminka2>) <prikaz1>;
} else <prikaz2>;
```

# Příklad

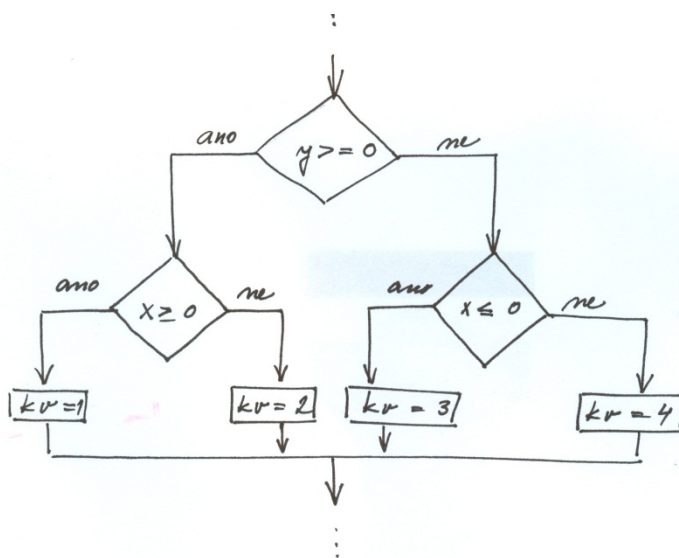
- Zapište program, který pro bod zadaný kartézskými souřadnicemi  $x$  a  $y$  určí číslo kvadrantu, ve kterém bod leží.

- Rozklad 2D na kvadranty



- Rozhodovací strom zapsaný vývojovým diagramem

- $x, y$  – souřadnice bodu
- $kv$  – výsledný kvadrant



```

package ja003kvadrant;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        float x, y;
        int kv;

        System.out.println("Program pro urceni kvadrantu zadaneho bodu");
        System.out.print("Zadej x souradnici bodu");
        x = sc.nextFloat();
        System.out.print("Zadej y souradnici bodu");
        y = sc.nextFloat();

        if (y >= 0)
            if (x >= 0) kv = 1;
            else kv = 2;
        else
            if (x <= 0) kv = 3;
            else kv = 4;

        System.out.format("%nBod (%1.2f; %1.2f) lezi v %d. kvadrantu%n", x, y, kv);
        System.out.format("%nKonec programu%n%n", x, y, kv);
    }
}

```