

3 Štruktúra programu v Jave

3.1 Popis zdrojového kódu HelloWorld.java aplikácie HelloWorld

Vývojové prostredie NetBeans nám pri vytvorení prvej aplikácie HelloWorld, automaticky vygenerovalo nasledovný zdrojový kód, ktorý sa nachádza v súbore HelloWorld.java:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package helloworld;

/**
 *
 * @author Student
 */
public class HelloWorld {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

Toto bude zatiaľ základná štruktúra každého zdrojového kódu, ktorý budeme vytvárať, iné budú len názvy balíkov (packages), prípadne hlavnej triedy (class). Zdrojový kód budeme dopĺňať o vlastné príkazy, taktiež vlastné triedy, ich metódy a pod.

Popíšeme si teraz jednotlivé časti zdrojového kódu HelloWorld.java aplikácie HelloWorld:

Komentár

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
```

Toto je **komentár**, ktorý my nahradíme vlastným komentárom, charakterizujúcim úlohu, ktorú budeme riešiť - ostatné komentáre z vygenerovaného zdrojového kódu vymažeme.

V jazyku Java môžeme používať tri typy komentárov:

```
/* Toto
 * je
 * viacriadkový
 * komentár*/
```

// Toto je jednoriadkový komentár, platný až do konca riadku

/** Toto je tzv. dokumentačný komentár. Pomocou aplikácie **javadoc** je možné ho vygenerovať
* priamo zo zdrojového kódu a použiť ho ako dokumentáciu k aplikácii. Tento typ komentára zatiaľ
* používať nebudeme, a preto ho z každej aplikácie vymažeme */

Komentáre používame na popis jednotlivých častí programu, aby sme po čase rýchlejšie pochopili, čo sa vlastne v komentovanej časti programu vykonáva. Komentáre sú užitočné nielen pre nás, ale aj pre iných programátorov, ktorí prípadne môžu náš kód upravovať.

Balík (package)

```
package helloworld;
```

Na tomto riadku je definovaný menný priestor (podobne ako v C++ using namespace), ktorý sa v jazyku Java realizuje rezervovaným slovom **package**, za ktorým nasleduje názov balíka. V našom prípade ide o balík s názvom *helloworld*, ktorý bol vygenerovaný priamo prostredím NetBeans.

Balíky sú dôležitou súčasťou jazyka Java, lebo zabráňujú konfliktom v pomenovaní tried, obmedzujú prístupové práva k triedam, metódam a premenným. Balíkom sa budeme bližšie venovať neskôr, zatiaľ si zapamätajme, že každý náš program bude „zabalený“ v samostatnom balíku, čo je v skutočnosti priečinok, ktorý obsahuje preložené zdrojové súbory (.class).

V našich programoch budeme ale často importovať – pomocou rezervovaného slova **import** – existujúce balíky jazyka Java, aby sme mohli využívať stovky hotových tried s množstvom funkcií. Importovanie balíkov je niečo podobné ako používanie hlavičkových súborov v jazyku C++.

Trieda HelloWorld

```
public class HelloWorld {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
    }  
  
}
```

Najdôležitejšou časťou zdrojového kódu je metóda, ktorá má nasledovnú signatúru:

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

Podobá sa funkcii `main()` v jazyku C++ a ide o vstupný bod aplikácie. V jazyku Java **musí byť táto metóda zapuzdrená** v triede (**class**), v našom prípade sa trieda volá **HelloWorld**. Je dôležité aby názov triedy, v ktorej je metóda *main* zapuzdrená, bol rovnaký ako meno súboru, v ktorom je metóda *main* uložená!

3 Štruktúra programu v Java

Neskôr budeme naše aplikácie dopĺňať aj o ďalšie triedy a plne tak využívať vlastnosti objektovo – orientovaného programovania.

Metóda `main(String[] args)`

```
public static void main(String[] args) {  
    // TODO code application logic here  
}
```

Metódu *main* musí obsahovať každá Java aplikácia. Po spustení aplikácie je volaná metóda *main*, jej sa vlastne predá riadenie celej aplikácie. V ďalšej fáze sú potom z tejto metódy volané všetky ostatné metódy, ktoré program vyžaduje.

Štandardná deklarácia a definícia metódy *main* vyzerá nasledovne:

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

`public static void main(String[] args)` – toto je hlavička metódy

`public` a `static` sú modifikátory – ich význam bude vysvetlený neskôr

`void` – znamená, že metóda nič nevracia

`main` – meno metódy; musí byť vždy `main`

`String[] args` – metóda prijíma jediný argument: pole prvkov typu `String`.

3.2 Cvičenie

1. Vytvorte aplikáciu s názvom **VypisTextu**. Postupujte rovnako ako pri vytváraní prvej aplikácie. V kroku 4., v poli *Create Main Class*, zmeňte názov triedy *VypisTextu* na *HlavnyProgram*.

Vo vygenerovanom kóde vymažte všetky komentáre, okrem prvého. V ňom zmeňte text nasledovne:

```
//Vypis textu.
```

```
//Program vypise niekoľko riadkov textu.
```

Ďalej do tela metódy *main()* zapíšte nasledovné tri riadky kódu:

```
System.out.println("Programujeme v jazyku JAVA");
```

```
System.out.println("Jazyk JAVA je objektový jazyk");
```

```
System.out.println("Jazyk JAVA je podobný jazyku C++");
```

Program skompilujte a spustite.

3 Štruktúra programu v Java

2. Vytvorte aplikáciu s názvom **InfoZiak**. Pri vytváraní aplikácie zmeňte triedu *InfoZiak* na *HlavnyProgram*. Vo vygenerovanom kóde zmažte všetky komentáre, okrem prvého – do neho zapíšete text charakterizujúci danú aplikáciu.

Úlohou vášho programu bude vypísať na prvom riadku Vaše meno a priezvisko a na druhom riadku mesto, v ktorom bývate.

3.3 Otázky

1. Na čo slúžia komentáre v zdrojovom kóde?
2. Aké typy komentárov používa Java?
3. Na čo slúžia balíky?
4. Uveďte hlavičku metódy `main()` a popíšte jednotlivé jej časti.
5. Na čo slúži metóda `main()` ?