

# PB162 Programování v jazyce Java

## 4. cvičenie

# kontrola

- opravených písomiek
  - na konci cvičenia prosím vrátiť
- funkčnosti príkladu z minula
- spúšťania z príkazového riadku, `CLASSPATH`
  - 1 bod

# rozhrania (interfaces)

- prvok dedičnosti (*inheritance*)
  - o dedičnosti nabudúce
- vyjadrujú „sľubovanú“ funkcionálnosť
  - = povinné metódy
- implementujúca trieda sa k nej „hlási“

# rozhrania (interfaces)

- Stack

- rozhranie:

```
public interface Stack {  
    boolean isEmpty();  
    void push(Object o);  
    Object pop();  
}
```

- iba **hlavičky** metód

# rozhrania – syntax

- `LinkedNodesStack`
  - implementuje rozhranie `Stack`:

```
public class LinkedNodesStack  
    implements Stack {
```

```
    private Node top = null;
```

```
    public void push(Object o) {  
        if (top != null) {  
            ...  
        }  
    }
```

# rozhrania – syntax

- deklarujeme **typom rozhrania**
- inicializujeme **konštruktorom impl. triedy**

```
public static void main(String[] args) {  
    Stack stack = new LinkedNodesStack();  
    stack.push("!");  
    ...  
}
```

- mohli by sme vytvoriť impl. `ArrayStack`  
(zásobník pomocou **poľa**) a nahradiť ním `LNS`  
zmenou **jediného riadku**

# rozhranie ≠ rozhranie

- *rozhranie*
  - vopred daný/dohodnutý spôsob komunikácie
  - blízke významu slova **protokol**
  - umožňuje delenie zložitosti na jednoduchšie podcelky
- rozhranie ako prvok jazyka Java
  - budeme nazývať slovom **interface**
  - interface slúži na definíciu *rozhrania*

# opäť zásobník

- `ArrayStack` – zásobník pomocou poľa
- rovnaká funkcionálnosť ako `LinkedListStack`
- funkcionálnosť vyjadríme **rozhraním** `Stack`



# javadoc

- trieda
  - popis, čo trieda reprezentuje + metadáta
    - `@author`, `@version`
    - `@since` – verzia, od ktorej je súčasťou API
- metódy
  - každej `public` metóde patrí práve jeden komentár
  - `@param` – parametre metódy (jedna na parameter)
  - `@return` – ak metóda niečo vracia
- javadoc k príkladu so zásobníkom

# syntaktické konštrukcie

- aritmetika

  - +, -, \*, /, %

- stručná aritmetika

  - c++, c\*=...

- podmienky

  - if-else, ?-:

- preťažovanie metód a konštruktorov

- statické atribúty a metódy

# aritmetika

- / – delenie

- `int / int = int`: celočíselné delenie

- `float / int = float, int / float = float`: delenie v obore reálnych čísel

ako dosiahnuť

- `3 / 4 = 0.75?`

- `int i = 3; int j = 4; i / j = 0.75?`

- % – delenie modulo (zvyšok po delení)

- `10 % 3 = 1`

# stručná aritmetika

- +=
  - `a += 3;`  $\Leftrightarrow$  `a = a + 3;`
  - `*=`, `-=`, `/=`, ...
  - `c _ = 5;`  $\Leftrightarrow$  `c = c _ 5;`
- ++c, c++ – inkrementácia
  - `c++;`  $\Leftrightarrow$  `c = c + 1;`
  - ale `c++` je výraz, `c = c + 1;` je príkaz
- vyskúšajte si (v metóde `main()`)

# podmienky

- `if-else`

```
if (a == 3) {  
    b = 0;  
} else {  
    b++;  
}
```

- ternárny operátor `?-:`

– výsledkom je výraz – **má hodnotu**

```
b = (a == 3) ? 0 : b + 1;
```

# preťažovanie metód

- `System.out.println()`
  - 1 metóda?
  - viac metód rovnakého mena!
- Ak existuje viac metód rovnakého mena, ktoré sa líšia počtom a typmi parametrov, hovoríme o **preťaženej metóde**.
- **preťažený konštruktor**
  - konštruktor triedy `Node`
- príklad

# statické atribúty a metódy

- atribút inštancie

```
private Node top
```

- môže k nemu pristupovať metóda inštancie

```
top = topNode.getPrevious();
```

- v každej inštancii **iná hodnota**

- atribút triedy – statický atribút

- kľúčové slovo **static**

```
private static int instancesCount;
```

- patrí **triede** – inštancie **zdieľajú** jeho **hodnotu**

- v rámci VM má len **jednu hodnotu**

# statické atribúty a metódy

- **statické metódy**

- vlastnené triedou
- volané nad triedou

```
ArrayStack.getInstancesCount ();
```

- **priamy** prístup **len** k **statickým** atribútom
- volanie metód inštancií, prístup k atribútom – musia **poznať referenciu**

- **nestatické metódy**

- môžu volať statické metódy (rešpektujúc viditeľnosť)

- príklad



# úlohy

odovzdávanie úloh – 2 archívy

odkaz na Bárinu stránku

voľné zadanie

chýba inštrukcia na vytvorenie objektu

