

# PB162 Programování v jazyce Java

## 3. cvičenie

## 2. test!

- zadanie rozdám
  - čitateľne učo

# konvencie

- budú vyžadované v úlohách, na písomkách
- pozor na **zbytočne** stratené body!
- viditeľnosť, rozsah platnosti
  - čo najmenší
  - atribúty `private`
  - lokálna premenná, kde ide

# konvencie

- menné konvencie
  - identifikátory anglicky, zmysluplne
  - camelSyntax, triedy veľkým písmenom
- bloky
  - telo `if`, `for`, `while`... v bloku, aj keď iba jeden príkaz
  - správne odsadenie

# spolupráca objektov

- objekty môžu obsahovať **referenciu** na iné objekty
- môžu **volať metódy** iných objektov
- príklad na zásobník

# zásobník

- bežná dátová štruktúra
  - LIFO = *last in, first out*
  - pre jednoduchosť zásobník reťazcov
- definujme pomocou operácií
  - **boolean isEmpty()** ;
    - **true** ak je zásobník prázdny, **false** inak
  - **void push**(String o) ;
    - vloží objekt na vrchol zásobníka
  - String **pop**() ;
    - vyberie objekt z vrcholu zásobníka

# zásobník – implementácia

- „retiazka“ objektov `Node`
  - každý nesie dáta – typ `String`
  - a odkaz na predchádzajúci `Node`
- štruktúra je schovaná = **zapúzdraná** za rozhraním triedy `LinkedListStack`
  - obsahuje iba odkaz na vrchol zásobníka

# trieda Node – zaujímavé detaily

- nemusí byť pre použitie zásobníka známa – **package private**
- používa **preťažený** konštruktor



# rozhrania (interfaces)

- prvok dedičnosti (*inheritance*)
  - podrobnejšie 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*

# nabudúce

- 2. úloha
  - časť na cvičení (3 b) musíte stihnúť tu
  - domáca časť (3 b) môžete dopracovať doma
- **pripravte sa!**
  - prejdite si dnešný príklad
  - skúste ho prerobiť na rozhranie + impl. triedu (ak ste nestihli)
  - minuloročné zadania úlohy na implementáciu rozhraní