

## Cvičenie 4

### 4.1 Dedenie

Vytvorte abstraktnú triedu s názvom Živočích, ktorá bude obsahovať nasledujúce atribúty a metódy:

- **pocet\_noh** dátový typ: int
- **zvuk()** návratový typ: void, abstraktná metóda

Vytvorte triedu Pes, ktorá bude dediť z triedy Živočích, a obsahovať nasledujúce metódy:

- **zvuk()** návratový typ: void, preťažuje funkciu z rodičovskej triedy

Vytvorte triedu Človek, ktorá bude dediť z triedy Živočích, a obsahovať nasledujúce metódy:

- **zvuk()** návratový typ: void, preťažuje funkciu z rodičovskej triedy

Pri inicializácii Psa a človeka inicializujte spomenuté atribúty v konštruktoře, pri zavolaní metódy vypíšte zvuk charakteristicky pre daného živočicha.

### 4.2 Collections - List

#### 4.2.1 Inicializácia

Vytvorte zoznam (ArrayList, alebo LinkedList) a naplňte ho prvkami typu:

- String
- int
- vlastný objekt napr. Živočích z predchádzajúcej úlohy

#### 4.2.2 Iterácia

Vypíšte prvky naplneného zoznamu. Vyskúšajte si správanie prvkov pri iterácii keď pracujete so zdedenými triedami - vysúšajte zavolať nejaké metódy triedy Živočích.

#### 4.2.3 Inicializácia z klávesnice

Vytvorte zoznam, ktorý umožní naplniť používateľovi ľubovoľným počtom prvkov z klávesnice.

#### 4.2.4 Modifikácia

Vytvorte zoznam, ktorý bude obsahovať  $N$  prvých prirodzených čísel. Spravte funkciu, ktorá z neho vymaže každé  $K$ -te číslo.  $N$  a  $K$  si zvolte ľubovoľne.  $K > 1$  a  $N > 2K$ .

Umožnite zadať  $N$  a  $K$  používateľovi z klávesnice.

#### 4.2.5 Vnorené kolekcie

Vytvorte ekvivalent dvojrozmerného poľa pomocou zoznamov. Naplňte ho prvkami a vypíšte.

#### 4.2.6 Triedenie

Vytvorte zoznam a naplňte ho vlastnými objektami. Zotriedte zoznam pomocou metódy `List.sort()`. Vyskúšajte si obe možnosti použitia - pomocou interface-u `Comparator` aj pomocou interface-u `Comparable`.

### 4.3 Collections - Map

Vytvorte objekt typu `Map dictionary` ktorý bude obsahovať preklady slov nižšie

- Áno - Yes
- Nie - No
- Objektovo orientované programovanie - Objected oriented programming

#### 4.3.1 Pridávanie

Vytvorte funkciu pre pridanie ďalšieho prekladu na základe interakcie používateľa z klávesnice, používateľ napíše slovo stlačením enter sa dostane k písaniu ďalšieho slova, predtým je však potrebné skontrolovať či už bolo dané slovo zadané v slovníku ak áno je potrebné vypísať hlášku o tom že nie je možné pridať ďalší preklad pre dané slovo. Ďalším stlačením sa daný preklad pridá do slovníku. Používateľovi sa po pridaní slova vypíše hláška či dané slovo bolo pridané do slovníku.

#### 4.3.2 Čítanie

Vytvorte funkciu ktorá vypíše po interakcii s používateľom preklad pre zadané slovo z klávesnice. Používateľ zadá slovo stlačením enteru sa mu pre dané slovo vypíše preklad alebo informácia o tom že dané slovo sa už v slovníku nachádza.

#### 4.3.3 Odstraňovanie

Vytvorte funkciu ktorá vymaže konkrétny prvok z objektu dictionary na základe interakcie používateľa. Používateľ zadá kľúč ktorý chce vymazať, v prípade ak chce používateľ vymazať prvok ktorý sa v objekte nenachádza vypíšte vhodnú hlášku.

#### 4.3.4 Vyprázdnenie

Vytvorte funkciu ktorá vymaže všetky hodnoty z objektu dictionary, pričom objekt zostane inicializovaný(nebude mať hodnotu null)

#### **4.3.5 Iterácia dvojíc**

Vytvorte funkciu ktorá preiteruje celým objektom dictionary a vypíše všetky preklady ktoré sa v ňom nachádzajú.

#### **4.3.6 Iterácia kľúčov**

Vytvorte funkciu ktorá preiteruje cez všetky kľúče objektu dictionary a vypíše jednotlivé kľúče do konzoly.

#### **4.3.7 Iterácia hodnôt s vyhľadávaním substringov**

Vytvorte funkciu ktorá preiteruje cez všetky hodnoty objektu dictionary a vypíše jednotlivé hodnoty do konzoly ktoré začínajú písmenom y.

#### **4.3.8 Kópia objektu a porovnanie**

Vytvorte kópiu objektu dictionary pomenujte ju dictionary2. Následne tieto dva objekty porovnajte s využitím metódy equals(). Výsledok zdôvodnite.