PB162 Programování v jazyce Java

9. cvičenie

úvodom...

- 2. písomka – zhrnutie
- úlohy

systém MHD

- zastávky (trieda `Stop`)
- linky (trieda `Line`)
  - priame
    - symetrické (12), nesymetrické (4, 39)
  - polokružné (6), okružné (?)
- sieť je zložitá a rozmanitá
  - uplatnenie dedičnosti
  - zjednodušenie
    - na nesymetrické a polokružné sa vyabstrahujme

systém MHD

- zásadnejšia otázka: ako modelovať sieť liniek
  - vymenovať zastávky v atribútoch
    - môžu pribudnúť
  - vytvoriť, povedzme, n atribútov a obsadiť len prvých k
    - zákon schválnosti:
    - \exist k. k = n + 1

systém MHD

- ...ako modelovať sieť liniek
  - použiť pole
    - pre programátora zložité na obsluhu, náchylné na chyby
      - pridávanie prvku
        - na koniec – zväčšovanie poľa
        - doprostred – detto + presúvanie prvkov
      - odoberanie prvku
        - …
        - + hrozí memory leak
- dôvod problémov – linky sú dynamické

systém MHD

- potrebujeme dynamickú dátovú štrukúru
  - vybudovanú nad poľom, binárnym stromom, pomocou hashovania
  - vybudovanú dobre
    - bez chýb, odskúšanú
    - efektívnu / so známym výkonom
- toto existuje v podobe rámca kolekcií – Java Collections Framework
  - hlavná stránka , triedy, tutoriál

Java Collections Framework

- súčasť Java Core API od v. 1.2
- poskytuje pohodlné rozhranie na prácu s dynamickými dátovými štruktúrami
- vyššia úroveň abstrakcie
  - množina, zoznam, fronta...
- usporiadanosť kolekcií
  - explicitne, implicitne usporiadané
  - neusporiadané

rozhranie `java.util.List` = zoznam

- lineárna dátová štruktúra
- explicitne (zvonka) usporiadané dáta
- metódy
  - `add`, `insert`
  - `contains`, `indexOf`, `get`
  - `remove`
- implementované triedami
  - `ArrayList` – pomocou poľa
  - `LinkedList` – pomocou previazaných objektov

`java.util.List` – použitie

- import balíčkov

```
import java.util.List;
import java.util.ArrayList;
```

- vytvorenie inštancie

```
private final List<Stop> stops = new ArrayList<Stop>();
```

- pridanie zastávky
  - `stops.add(stop);` (na koniec)
  - `stops.add(4, stop);` (medzi 4. a 5.)
- získanie 5. zastávky
  - `stops.get(4);` (zoznam číslovaný od 0)

`java.util.List` – použitie

- odobratie zastávky 5. zastávky
  - `stops.remove(4);`
- odobratie 1. výskytu zastávky
  - `stops.remove(stop);`
- vyprázdnenie zoznamu
  - `stops.clear()`
- iterácia – for-each

```
for (Stop stop: stops) {
    System.out.println(stop);
}
```