



Záverečný test

Zadanie



Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Dvakrát meraj (rozmýšľaj), raz rež (programuj)

Pravidlá a informácie:

- čas na riešenie úloh je **240 minút**,
- nie je dovolená žiadna (elektronická aj neelektronická) komunikácia s kýmkoľvek okrem dozoru
- v prípade akýchkoľvek problémov alebo z dôvodu ohodnotenia riešenia kontaktujte dozor,
- riešenia je možné nechať si ohodnotiť aj priebežne,
- funkčnosť každej metódy musí byť preukázaná spustením na vami vytvorenom testovacom vstupe, nespustiteľné metódy neumožňujú zisk príslušných bodov,
- všetky inštančné premenné a metódy s výnimkou metód uvedených v zadaní musia byť neverejné.

Plaváreň mesta PAZice

Motivácia: Mesto PAZice zmodernizovalo svoju krytú plaváreň. Jednou z novinek je vstup/výstup cez turniket na kartu. Vďaka tomu získalo mesto aj presnú evidenciu využitia plavárne jednotlivými návštevníkmi. Vytvorte systém, ktorý mestu pomôže „vydolovať“ z údajov, ktoré majú, nejaké zaujímavé informácie. Výstupy z turniketov sú vo forme mesačných správ, t.j. všetky údaje sú viazané na konkrétny kalendárny mesiac.



Zadanie: V balíku *sk.upjs.finalTerm* vytvorte triedu *Navsteva*, ktorá bude definovať inštančné premenné na uloženie údajov o jednej návšteve plavárne:

- deň** v mesiaci, kedy návštevník do plavárne vstúpil (deň návštevy),
- hodina a minúta** (čas) dňa, kedy návštevník vstúpil na plaváreň (prešiel turniketom),
- trvanie** vstupu v minútach, t.j. čas aký návštevník počas tejto návštevy strávil na plavárni,
- meno a priezvisko** návštevníka (kvôli jednoduchosti predpokladáme, že ľudia majú rôzne mená – pri skutočnej aplikácii by sme namiesto mena použili napríklad číslo karty).

Ďalej vytvorte triedu *ZoznamNavstev*, ktorá reprezentuje zoznam všetkých návštev plavárne v danom kalendárnom mesiaci. Všetky inštančné premenné a metódy vytváraných tried s výnimkou metód uvedených v zadaní musia byť neverejné.

Konštruktory a pridávanie návštev (4 body dokopy – povinné):

- public** *Navsteva*(**int** den, **int** hodina, **int** minuta, **int** trvanie, String meno)
- public void** *pridaj*(*Navsteva* navsteva) – inštančná metóda v triede *ZoznamNavstev*, ktorá pridá záznam o návšteve plavárne do zoznamu

Práca so súbormi :

V triede *Navsteva* (zoStringu 5 bodov, toString 1 bod):

- public static** *Navsteva* zoStringu(String popis) – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy *Navsteva*. Parametrom je *String* v tvare "den v mesiaci \t hodina:minúta \t trvanie \t meno a priezvisko". Príklady:

```
3      8:30      120   Peter Plavec
4      15:59     50    Martina Moravcová
```

Pozn.: Znak \t je neviditeľný znak tabulátora. Scanner-u môžete povedať, že oddeľovačom je tabulátor zavolaním jeho metódy *useDelimiter("\t")*.

- public** String *toString*() – vráti reťazec vhodne reprezentujúci daný vstup do plavárne.

V triede `ZoznamNavstev` (zoSuboru 4 body, uloz 3 body, toString 1 bod):

- **public static** `ZoznamNavstev zoSuboru(String nazovSuboru)` – statická metóda, ktorá z uvedeného súboru prečíta zoznam návštev (v každom riadku bude informácia o jednej návšteve plavárne; odporúčame formát, aký používa metóda `Navsteva.toStringu`).
- **public void** `uloz(String nazovSuboru)` – uloží zoznam návštev do súboru v takom tvare, že ho metóda `zoSuboru` vie naspäť prečítať.
- **public** `String toString()` – vráti reťazec vhodne reprezentujúci všetky návštevy v zozname.

Inštančné metódy triedy `ZoznamNavstev`:

- **public double** `priemernaDlzkNavstevy()` – vráti priemernú dĺžku jednej návštevy plavárne (2 body)
- **public** `List<Navsteva> navstevyOsoby(String meno)` – vráti zoznam všetkých návštev daného návštevníka (3 body)
- **public** `Navsteva najdlhsiaNavsteva()` – vráti referenciu na najdlhšie trvajúcu návštevu plavárne v mesiaci (3 body)
- **public** `Map<String, Integer> casNaPlavarniPodlaOsob()` – pre každú osobu (návštevníka) vypočíta, aký celkový čas strávil na plavárni, výsledok vráti vo forme mapy (4 bodov)
- **public boolean** `bolViacKratZaDen(String meno)` – vráti, či zadaná osoba navštívila plaváreň v niektorý deň mesiaca aspoň dvakrát, t.j. v niektorý deň mesiaca máme pri tejto osobe poznačené aspoň 2 návštevy (5 bodov)
- **public** `String odisielPosledny(int den)` – vráti meno návštevníka, ktorý spomedzi návštevníkov, ktorí plaváreň navštívili v daný deň mesiaca, z nej odišiel ako posledný (6 bodov)
- **public** `String najDenTyzdna(int prvvyPondelok)` – vráti deň v týždni (pondelok, ..., nedeľa), počas ktorého sme zaznamenali najväčší celkový počet návštev, ak vieme, že prvý pondelok v mesiaci bol v zadaný deň mesiaca (*prvyPondelok*). Celkový počet návštev v dni týždňa znamená, že sčítavame **počty** návštev v daný deň týždňa v rámci celého mesiaca (7 bodov).
- **public** `ZoznamNavstev zoznamNavstevVObdobi(int denOd, int denDo)` – vráti referenciu na nový zoznam návštev, v dňoch *denOd* (vrátane) až *denDo* (vrátane) - (5 bodov)
- **public double** `prijmy(double cenaZaHodinu, int hodinyNaZdarmaHodinu)` – vráti celkový mesačný príjem plavárne, ak cena za každú začatú hodinu je *cenaZaHodinu* a po každých *hodinyNaZdarmaHodinu* platených hodinách je jedna hodina zdarma (6 bodov)
- **public** `List<String> naPlavarni(int den, int hodina, int minuta)` – vráti mená osôb, ktoré sa v danom čase nachádzali na plavárni (6 bodov)
- **public** `String najviacLudi(int den)` – vráti čas, kedy bolo v daný deň na plavárni najviac osôb – poznamenajme, že stačí uvažovať iba časy, kedy niekto do plavárne vstúpil (3 bodov)

Výnimky (každá použitá po 4 body):

- Vytvorte kontrolovanú výnimku `NacitanieZlyhaloException` a použite ju v metóde `ZoznamNavstev.zoSuboru`.
- Vytvorte nekontrolovanú výnimku `MenoNeznameException` a použite ju vo vhodných metódach.

Triedenie a komparátor (dokopy 5 bodov):

Implementovaním rozhrania `java.util.Comparator<Navsteva>` vytvorte triedu `ChronologickyPorovnavac` s metódou:

- **public int** `compare(Navsteva n1, Navsteva n2)` – porovná návštevy podľa dátumu a času vstupu (chronologické porovnanie).

V triede `ZoznamNavstev` implementujte inštančnú metódu:

- **public void** `zoradPodlaCasuVstupu()` – chronologicky usporiada návštevy v zozname podľa času vstupu na plaváreň.