



Záverečný test

Zadanie



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 musia byť neverejné.

Podrobné výpisy hovorov

Motivácia: Už dlhšie každý slovenský telefónny operátor umožňuje ako súčasť faktúry získať podrobný výpis uskutočnených telefónnych hovorov (ale aj odoslaných SMS a uskutočnených dátových prenosov, no kvôli zjednodušeniu zadania tieto nebudeme uvažovať). Keďže žijeme v období dôsledkov ekonomickej krízy a všade sa hovorí o šetrení a optimalizácii výdavkov, pre jednotlivcov a obzvlášť firmy má zmysel detailne analyzovať a následne optimalizovať výdavky za telekomunikačné služby. Naším cieľom bude vytvoriť základ pre jednoduchú aplikáciu umožňujúcu jednoduchú analýzu uskutočnených hovorov z firemných telefónov. Hlavným „vstupom“ pre našu aplikáciu by mal byť textový súbor obsahujúci podrobný výpis uskutočnených telefónnych hovorov z (jedného alebo viacerých) firemných telefónov.



Pohľad analytika: Pri implementácii budeme potrebovať:

- triedu `Hovor`, ktorá bude uchovávať údaje o jednom odchádzajúcom telefonickom hovore (kedy a z akého čísla na aké číslo sa volalo, koľko trval a stál hovor),
- triedu `ZoznamHovorov`, ktorá bude uchovávať zoznam telefonických hovorov (typicky v rámci jedného alebo viacerých zúčtovacích období).

Zadanie: V balíku `sk.upjs.finalTerm` vytvorte triedu `Hovor` obsahujúcu dátové položky prístupné cez `gettre` (a podľa uváženia aj modifikovateľné cez `settre`):

- **datum** (dátum, kedy sa hovor uskutočnil, resp. začal, napr. "7.1.2013"),
- **cas** (čas začiatku telefonického hovoru v rámci dňa, napr. "12:05:15"),
- **cisloVolajuceho** (telefónne číslo, z ktorého sa volalo, napr. "+421-903123456"),
- **smer** (operátor, do ktorého siete sa volalo, napr. "O2 SK", "ORANGE SK", "O2 AT"),
- **volaneCislo** (telefónne číslo, na ktoré sa volalo, napr. "+421-948123456"),
- **trvanie** (dĺžka trvania hovoru v sekundách, napr. 120, 45, ...),
- **cena** (cena za uskutočnený hovor v eurách, napr. 0.52, 1.45, ...).

Upozornenie: Zadanie triedy `Hovor` predpisuje dátové položky prístupné cez `gettre`. Aké privátne inštančné premenné použijete na uloženie týchto dátových položiek je na vašom rozhodnutí.

Ďalej vytvorte aj triedu `sk.upjs.finalTerm.ZoznamHovorov`, ktorá bude uchovávať zoznam telefonických hovorov.

Poznámka: Môžete predpokladať, že všetky telefónne čísla s výnimkou špeciálnych skrátených čísel (napr. 158, audiotext, prémiové SMS, ...) sú zadané vo formáte s medzinárodnou telefónnou predvoľbou. Tá začína znakom + a od ďalšej časti telefónneho čísla je oddelená znakom -.

Konštruktory a pridávanie hovorov (2 body dokopy – povinné):

- **public** Hovor(String datum, String cas, String cisloVolajuceho, String smer, String volaneCislo, int trvanie, double cena) – použije sa na vytvorenie záznamu o uskutočnenom telefonickom hovore,
- **public void** pridaj(Hovor hovor) – inštančná metóda v triede ZoznamHovorov, ktorá pridá záznam o hovore do zoznamu hovorov.

Práca so súbormi (povinné):

V triede Hovor:

- **public static** Hovor zoStringu(String popis) – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy Hovor. Parameter je String v tvare "dátum \t čas \t číslo volajúceho \t smer \t volané číslo \t trvanie \t cena" (3 body);
Poznámka: Znak \t je neviditeľný znak tabulátora. Scanner-u môžete povedať, že oddeľovač má byť tabulátor zavolaním jeho metódy useDelimiter("\t").
- **public** String toString() – vráti reťazec vhodne reprezentujúci záznam o hovore (1 bod).

V triede ZoznamHovorov:

- **public static** ZoznamHovorov zoSuboru(File f) – statická metóda, ktorá z uvedeného súboru prečíta zoznam hovorov, pričom v každom riadku bude popis jedného hovoru (4 body).
- **public void** uloz(File subor) – uloží všetky záznamy o všetkých hovoroch do súboru vo formáte, ktorý vie prečítať metóda zoSuboru (3 body).
- **public** String toString() – vráti reťazec vhodne reprezentujúci všetky záznamy hovorov v zozname (1 bod).

Inštančné metódy triedy ZoznamHovorov:

- **public double** vydavky(String telefonneCislo) – vráti, koľko sa zaplatilo celkom za hovory uskutočnené zo zadaného telefónneho čísla (1 bod).
- **public** Map<String, Double> vydavky() – vráti map-u, ktorá pre každé telefónne číslo obsahuje celkové výdavky za hovory z tohto telefónneho čísla (4 body).
- **public** List<Hovor> hovoryNaCislo(String volaneCislo) – vráti referenciu na novovytvorený zoznam záznamov hovorov, ktorý obsahuje všetky hovory smerujúce na zadané číslo (3 body).
- **public** Map<String, Double> priemernaDlzkaHovoru() – vráti map-u, ktorá pre každé telefónne číslo obsahuje priemernú dĺžku jedného hovoru z tohto telefónneho čísla (7 bodov).
- **public** List<String> cislaUOperatora(String operator) – vráti zoznam všetkých telefónnych čísel, na ktoré sa volalo a ktoré sú (alebo boli) u zadaného operátora (napr. "O2 SK"). Každé telefónne číslo nech je v zozname nanajvyš raz (4 body).
- **public** String presunuteCislo() – vráti akékoľvek telefónne číslo, u ktorého možno zo záznamov hovorov „vyčítať“, že toto číslo bolo presunuté medzi operátormi (napr. z "O2 SK" do "ORANGE SK"). Ak také číslo neexistuje, vráti sa null (6 bodov).
- **public** List<Hovor> mimoPracovnejDoby(String cislo) – vráti zoznam všetkých hovorov, ktoré boli realizované zo zadaného čísla mimo pracovnej doby. Hovor považujeme za hovor mimo pracovnej doby, ak jeho začiatok je mimo čas 8:00-16:00 (4 body).
- **public** ZoznamHovorov zahranicneHovory() – vráti referenciu na novovytvorený zoznam hovorov obsahujúci všetky hovory, ktoré smerovali do zahraničia. Hovor do zahraničia možno rozpoznať tak, že medzinárodná predvoľba volaného čísla je iná ako +421 (5 bodov).
- **public** Hovor najdlhsiHovor() – vráti, referenciu na najdlhšie trvajúci hovor (3 body).

- **public double** podielVnutrofiremnychHovorov() – vráti, koľko percent z prevolaného času tvorili vnútrofiremné hovory (volaný aj volajúci sú firemné čísla). Za firemné čísla považujeme tie čísla, ktoré sú v zozname hovorov uvedené ako volajúce, t.j. predpokladáme, že z každého firemného čísla sa volalo aspoň raz (7 bodov).
- **public int[]** casHovorovPoHodinach() – vráti pole dĺžky 24, ktoré pre každú hodinu bude obsahovať, koľko sekúnd hovorov bolo realizovaných v danej hodine dňa. Napríklad na indexe 3 bude celkové trvanie hovorov v sekundách, ktoré prebiehali v akýkoľvek deň v čase od 03:00:00 do 03:59:59. Ak hovor začal o 03:59:00 a trval 80 sekúnd, tak 60 sekúnd trvania hovoru „prirátavame“ k hodine 3 a 20 sekúnd hovoru k hodine 4. Hovor môže trvať aj niekoľko hodín (14 bodov).

Triedenie a komparátor (dokopy 8 bodov):

Vytvorte triedu KomparatorHovorov implementujúcu `java.util.Comparator<Hovor>` s metódou:

- **public int** compare(Hovor h1, Hovor h2) – porovná hovory podľa dátumu a času začiatku hovoru.

V triede ZoznamHovorov implementujte inštančnú metódu:

- **public void** zorad()– usporiada hovory podľa dátumu a času začiatku hovoru počnúc najstarším hovorom.

Ďalšie metódy:

V triede *Hovor* vytvorte pomocnú statickú metódu:

- **public static** String naCasovyRetazec(int sekundyOdPolnoci) – vráti reťazec vo formáte HH:MM:SS, ktorý zodpovedá zadanému času v sekundách od polnoci. Pri jednotlivých číslach v prípade potreby použite úvodnú nulu, t.j. očakáva sa výstup v tvare "05:10:09" a nie "5:10:9" (5 bodov).

Rady a nápoveda:

Čas: Pri riešení viacerých úloh môže byť vhodné čas vo forme reťazca vo formáte "H:M:S" previesť na číslo vyjadrujúce počet sekúnd od polnoci daného dňa pomocou jednoduchého vzorca: $3600 * H + 60 * M + S$.

```
public static int casovyRetazecNaSekundy(String casovyRetazec) {
    Scanner citac = new Scanner(casovyRetazec);
    citac.useDelimiter(":");
    int hh = citac.nextInt();
    int mm = citac.nextInt();
    int ss = citac.nextInt();
    citac.close();
    return hh * 3600 + mm * 60 + ss;
}
```

Dátum: V prípade porovnávania dátumov možno použiť podobnú fintu ako s prevodom časového reťazca.

Scanner 1: Ak chcete nastaviť pre *Scanner* ako oddeľovač tokenov znak . (bodka), môžete tak spraviť zavolaním metódy `useDelimiter("\\. ")`.

Scanner 2: Ak chcete nastaviť, aby *Scanner* korektne čítal čísla s desatinnou bodkou, spravte tak zavolaním metódy `useLocale(Locale.US)`.