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.

Jedálne UPJŠ

Motivácia: Jedálne UPJŠ sú neustále pod paľbou (viac, či menej) oprávnenej kritiky. Základom pre každé zlepšovanie sú rôzne podnety a analýza súčasného stavu. A práve oblasť analýzy je miesto, kde informačné technológie môžu pomôcť. Obzvlášť ak uvážime, že súčasný systém CARDPAY by mohol poskytovať množstvo zaujímavých údajov, ktorých spracovaním by išlo vydolovať zaujímavé informácie (dokonca aj o sociálnych vzťahoch medzi stravníkmi, ako naznačuje jedna z úloh).

Pripomeňme si, ako fungujú jedálne UPJŠ. UPJŠ spravuje niekoľko jedálni (Jesenná, Medická, Šrobárová, ...). Stravníci si môžu obedy (ďalej len jedlá) objednávať vopred cez webovú aplikáciu alebo priamo cez terminály. Jednotlivé uskutočnené objednávky uchováva používaný informačný systém CARDPAY. Pri výdaji stravy („pri okienku“) sa stravník identifikuje priložením svojej čipovej (ISIC alebo zamestnaneckej) karty k čítačke. Následne terminál vo výdajni pani kuchárke zobrazí objednané jedlo a poznačí sa, že táto objednávka bola vybavená (=jedlo bolo vydané).

Vašou úlohou je teraz naprogramovať systém na jednoduchú analýzu objednávok jedál.

Pohľad analytika: Pri implementácii budeme potrebovať:

- triedu *ObjednavkaJedla*, ktorá bude uchovávať údaje o jednej objednávke jedla (kto, kedy, čo a kam objednal) a stave jej „vybavenia“ (či už jedlo bolo vydané),
- triedu *Objednavky*, ktorá bude uchovávať zoznam objednávok jedál.

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

- **datumObjednavky** (dátum, kedy bolo jedlo objednané "7.1.2013"),
- **jedalen** (názov jedálne, do ktorej bolo jedlo objednané, napr. "Jesenná"),
- **jedlo** (názov objednaného jedla, napr. "Tvarohový nákyp"),
- **idKarty** (ID karty objednávateľa, napr. "PIK1234567890" alebo "ZAM1234567890"),
- **datumVydaja** (dátum, na ktorý bolo jedlo objednané, napr. "10.1.2013").
- **casVydaja** (čas výdaja jedla, ak už jedlo bolo vydané, napr. "12:05:15")

Upozornenie: Zadanie triedy *ObjednavkaJedla* 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.Objednavky*, ktorá bude uchovávať zoznam objednávok jedál.

Konštruktory a pridávanie objednávok jedál (3 body dokopy – povinné):

- `public ObjednavkaJedla(String datumObjednavky, String jedalen, String jedlo, String idKarty, String datumVydaja)` – použije sa na vytvorenie objednávky jedla, ktoré ešte nebolo vydané v jedálni,
- `public ObjednavkaJedla(String datumObjednavky, String jedalen, String jedlo, String idKarty, String datumVydaja, String casVydaja)` – použije sa na vytvorenie objednávky jedla, ktoré už bolo vydané v jedálni (konštruktor je doplnený o čas, kedy bolo jedlo vydané),
- `public void pridaj(ObjednavkaJedla objednavka)` – inštančná metóda v triede *Objednavky*, ktorá pridá objednávku jedla do zoznamu objednávok.

Práca so súbormi (povinné):

V triede *ObjednavkaJedla*:

- `public static ObjednavkaJedla zoStringu(String popis)` – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy *ObjednavkaJedla*. Parameter je *String* v tvare "dátum objednávky `\t` jedáleň `\t` jedlo `\t` id karty `\t` dátum výdaja `\t` čas výdaja" ak jedlo už bolo vydané, resp. "dátum objednávky `\t` jedáleň `\t` jedlo `\t` id karty `\t` dátum výdaja" ak jedlo ešte nebolo vydané (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 objednávku jedla (1 bod).

V triede *Objednavky*:

- `public static Objednavky zoSuboru(File f)` – statická metóda, ktorá z uvedeného súboru prečíta zoznam objednávok jedál, pričom v každom riadku bude popis jednej objednávky (4 body).
- `public void uloz(File subor)` – uloží všetky záznamy o všetkých objednávkach jedál do súboru (3 body).
- `public String toString()` – vráti reťazec vhodne reprezentujúci všetky objednávky jedál (1 bod).

Inštančné metódy triedy *Objednavky*:

- `public int pocetNevydanychJedal(String jedalen, String datum)` – vráti počet jedál, ktoré ešte neboli vydané v zadanej jedálni v daný deň výdaja (2 body).
- `public Objednavky zoznamObjednavokPodlaKarty(String idKarty)` – vráti referenciu na novovytvorený zoznam objednávok, ktorý obsahuje len objednávky používateľa so zadaným číslom karty (3 body).
- `public double podielJedalPreStudentov()` – vráti koľko percent všetkých jedál bolo objednaných študentmi. To, či jedlo objednal študent alebo zamestnanec, je možné rozpoznať na základe id-čka karty. Id-čka študentských kariet začínajú písmenami PIK (3 body).
- `public List<String> dniSJedlom(String idKarty)` – vráti zoznam dátumov, v ktorých bol stravník (určený id-čkom karty) v jedálni. Pozor, stravník môže mať na jeden deň objednaných aj viacero jedál (3 body).
- `public Map<String, Integer> nevydaneJedla(String jedalen, String datum)` – vráti map-u, ktorá pre každé jedlo obsahuje počet nevydaných porcií tohto jedla v zadanej jedálni v daný deň výdaja (4 body).
- `public boolean objednavaVopred(String idKarty)` – vráti, či si používateľ so zadaným číslom karty objednáva jedlá na niekoľko dní vopred (napr. v piatok si objedná jedlá na celý budúci týždeň). Takého používateľa rozpoznáme tak, že množina dátumov (A), kedy si jedlo objednal, je oveľa menšia ako množina dátumov (B), na kedy bolo jedlo objednané. Konkrétne vyžadujeme, aby prvá z množín (A) bola nanajvýš 30% z veľkosti druhej množiny (B). (5 bodov).

- `public String najoblubenejsieJedlo()` - vráti názov najobľúbenejšieho jedla (ak je takých viac, vráti ktorékoľvek z nich). Najobľúbenejšie jedlo je také jedlo, ktoré bolo najviac krát objednané (5 bodov).
- `public String najoblubenejsieZJedal(Set<String> jedla)` - vráti názov najobľúbenejšieho jedla (ak je takých viac, vráti ktorékoľvek z nich) spomedzi jedál, ktoré sú v zadanej množine názvov jedál. Tak ako v predošlej úlohe, obľúbenosť jedla je „meraná“ počtom objednávok tohto jedla (3 body).
- `public String predpokladanyVyber(Set<String> jedla, String idKarty)` - vráti názov jedla z množiny jedál, o ktorom je najpravdepodobnejšie, že si ho používateľ so zadaným číslom karty vyberie. Ako najpravdepodobnejšie jedlo vyberáme to jedlo, ktoré je najobľúbenejším jedlom tohto používateľa spomedzi zadaných jedál na základe doterajšej histórie objednávok jedál tohto používateľa (3 body).
- `public Objednavky zoznamObjednavokVMesiaci(int mesiac, int rok)` – vráti referenciu na novovytvorený zoznam objednávok obsahujúci len objednávky jedál s dátumom výdaja v zadanom kalendárnom mesiaci (5 bodov).
  Rada: Ak chcete nastaviť pre *Scanner* ako oddeľovač tokenov znak `.` (bodka), môžete tak spraviť zavolaním metódy *useDelimiter("\\.")*.
- `public int[] pocetJedalVMesiaci(String idKarty, int rok)` – pre zadaného stravníka (*idKarty*) a zadaný rok vráti pre každý mesiac daného roka, koľko jedál mal tento stravník objednaných na daný mesiac. Vrátené pole má dĺžku 12 a na indexe 0 je počet jedál v januári, na indexe 1 vo februári, atď. (6 bodov).
- `public boolean chodiaSpoluNaObed(String idKarty1, String idKarty2)` – vráti, či zadaní dvaja stravníci (určení id-čkami kariet) sa chodia stravovať spolu. Povieme, že dvaja stravníci sa chodia stravovať spolu práve vtedy, keď počet dní, v ktorých časový rozdiel medzi časmi výdaja ich jedál v tej istej jedálni je menší ako 3 minúty, tvorí aspoň 80% z počtu všetkých dní, v ktorých boli obaja na jedlo v ktorejkoľvek z jedální. Kvôli jednoduchosti môžete predpokladať, že každý stravník má v daný deň objednané nanajvýš jedno jedlo (15 bodov).
  Rada: Táto úloha vyžaduje určiť časový rozdiel medzi dvoma časmi. Tieto časy ale máme na vstupe vo forme reťazca vo formáte "H:M:S", čo nie je veľmi praktické na zistenie časového rozdielu. Jedna z možných fínt je previesť si tento časový reťazec na číslo vyjadrujúce počet sekúnd od polnoci daného dňa pomocou jednoduchého vzorca: 3600*H+60*M+S.

Výnimky:

- Vytvorte nekontrolovanú výnimku `NeznamaJedalenException` a použite ju aspoň na dvoch vhodných miestach (3 body)

Triedenie a komparátor (dokopy 6 bodov):

Vytvorte triedu *KomparatorObjednavok* implementujúcu *java.util.Comparator<ObjednavkaJedal>* s metódou (4 body):

- `public int compare(ObjednavkaJedal o1, ObjednavkaJedal o2)` - porovná objednávky jedál podľa dátumu objednania; ak majú objednávky rovnaký dátum, pri porovnaní rozhoduje názov jedálne (abecedne). Pri rovnakom dátume objednávky a jedálni, porovnajte objednávky abecedne podľa názvu jedla.

V triede *Objednavky* implementujte inštančnú metódu (2 body):

- `public void zoradPodlaDatumuObjednania()`-usporiada objednávky jedál podľa dátumu objednania.