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 musia byť neverejné.

MAIN - Klobásky

Motivácia: Väčšina prvákov prijala pozvanie na úvodné sústredenie študentov prvého ročníka. Jedna z aktivít, ktorá sa realizuje na sústredení matematikov a informatikov, je kombinatoricko-logická súťaž jednotlivcov *Klobásky*. Súťaž má stanovené časové trvanie. Na začiatku súťaže majú všetci rovnaký rating. Počas súťaže súťažiaci vyzývajú svojich protihráčov na duely (jeden duel je jedna partia hry *Klobásky*). Na základe výsledku duelu a aktuálnych ratingov duelantov sa potom vypočíta ich nový rating. Cieľom je mať na konci súťaže maximálny rating. Výsledky jednotlivých duelov chronologicky zapisuje zapisovateľ a na základe nich aktualizuje aktuálne ratingy súťažiacich. Keďže matematici, informatici a určite aj fyzici radi analyzujú údaje, vytvoríme program, ktorý nám pomôže zodpovedať niektoré otázky o priebehu celej súťaže.

Pohľad analytika: Pri implementácii budeme potrebovať:

- triedu `Duel`, ktorá bude uchovávať údaje o výsledku jedného duelu, t.j. kto bol účastníkom duelu, kto duel vyhral, kto prehral a kedy duel skončil,
- triedu `Sutaz`, ktorá bude uchovávať zoznam duelov, ktoré sa uskutočnili v rámci danej súťaže.

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

- **cas** (čas v sekundách od začiatku súťaže, kedy duel skončil, resp. bol zapísaný výsledok duelu),
- **vitaz** (meno osoby, ktorá vyhrala duel),
- **porazeny** (meno osoby, ktorá prehrala duel).

Upozornenie: Zadanie triedy `Duel` predpisuje dátové položky prístupné cez gettre. Aké privátne inštančné premenné použijete na uloženie týchto dátových položiek je na vašom rozhodnutí. Predpokladáme tiež, že žiadni dvaja súťažiaci nemajú rovnaké mená.

Ďalej vytvorte aj triedu `sk.upjs.finalTerm.Sutaz`, ktorá bude uchovávať zoznam duelov uskutočnených v rámci súťaže.

Konštruktory a pridávanie duelov (3 body dokopy – povinné):

- `public Duel(int cas, String vitaz, String porazeny)` – použije sa na vytvorenie záznamu o výsledku uskutočneného duelu,
- `public void pridaj(Duel duel)` – inštančná metóda v triede `Sutaz`, ktorá pridá záznam o dueli do zoznamu duelov súťaže.

Práca so súbormi (povinné):

V triede `Duel`:

- `public static Duel zoStringu(String popis)` – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy `Duel`. Parameter je `String` v tvare "čas `\t` meno víťaza `\t` meno porazeného" (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 údaje duelu (1 bod).

V triede `Sutaz`:

- `public static Sutaz zoSuboru(File f)` – statická metóda, ktorá z uvedeného súboru prečíta zoznam duelov, pričom v každom riadku bude popis jedného duelu (4 body).
- `public void uloz(File subor)` – uloží všetky záznamy o všetkých dueloch súťaže do súboru v tvare, ktorý vie spracovať metóda `zoSuboru(File f)` (3 body).
- `public String toString()` – vráti reťazec vhodne reprezentujúci všetky duely v súťaži (1 bod).

Inštančné metódy triedy `Sutaz` (za predpokladu, že výsledky duelov sa pridávajú chronologicky)**:**

- `public int pocetDuelov(String meno)` – vráti počet duelov, ktorých sa zúčastnil súťažiaci so zadaným menom (1 bod).
- `public List<String> sutaziaci()` – vráti zoznam mien (bez opakovania) tých súťažiacich, ktorí sa zúčastnili aspoň jedného z duelov súťaže (4 body).
- `public double priemernyPocetDuelovNaSutaziaceho()` – vráti priemerný počet duelov, ktorých sa zúčastnil jeden súťažiaci (2 body).
- `public int pocetOdvetnychDuelov()` – vráti počet odvetných duelov v súťaži. Povieme, že duel je odvetný, ak sa jeho duelanti už skôr stretli v nejakom dueli (4 body).
- `public List<String> stastniSutaziaci()` – vráti zoznam mien (bez opakovania) tých súťažiacich, ktorí svoj posledný duel v súťaži skončili výhrou (6 bodov).
- `public int najdlhsiaVyhernaSeria(String meno)` – vráti maximálny počet výhier v rade za sebou zadaného súťažiaceho (6 bodov).
- `public List<String> najdeptajucejsiSutaziaci()` – vráti zoznam (bez opakovania) mien tých súťažiacich, ktorí dosiahli najväčší počet výhier v dueloch – vracia sa zoznam, keďže najväčší počet výhier v súťaži môže mať viacero súťažiacich (7 bodov).
- `public Map<String, Integer> vysledkovka(int pociatocnyRating, int a, int b, int c)` – pre každého súťažiaceho vráti jeho rating na konci súťaže. Na začiatku má každý súťažiaci rating s hodnotou `pociatocnyRating` (napr. 1000). Po každom dueli sa rating duelantov zmení o hodnotu *a – b*(rating vyhrávajúceho – rating prehrávajúceho)/c*. Rating vyhrávajúceho sa zvýši o túto hodnotu a rating prehrávajúceho sa zníži o túto hodnotu (7 bodov).
- `public int casovyUsekSNajviacDuelmi(int trvanieUseku)` – vráti čas od začiatku súťaže v sekundách, kedy začal časový úsek s trvaním `trvanieUseku`, počas ktorého sa zapísal (skončil) najväčší počet výsledkov duelov. Táto úloha má veľa prípustných výsledkov. Dá sa však ukázať, že vždy existuje taký prípustný výsledok, ktorý je totožný s časom zapísania nejakého duelu (7 bodov).
- `public int najkratsiDuel()` – vráti trvanie najkratšieho duelu v sekundách. Predpokladáme, že prvé duely začali v čase 0 a každý súťažiaci je hneď po zapísaní výsledku duelu pripravený na ďalší duel, t.j. čas na výber protihráča zanedbávame. Duel začína hneď ako sú obaja duelanti pripravení (7 bodov).

Triedenie a komparátor (dokopy 5 bodov):

Niekedy sa stane, že výsledky duelov sa nepodarí zapísať chronologicky. Vytvorte riešenie, ktoré umožní usporiadať duely chronologicky.

Vytvorte triedu `KomparatorDuelov` implementujúcu `java.util.Comparator<Duel>` s metódou:

- `public int compare(Duel d1, Duel d2)` – porovná duely podľa času ukončenia (zapísania výsledku) duelu.

V triede `Sutaz` implementujte inštančnú metódu:

- `public void zoradChronologicky()`– usporiada duely podľa času ukončenia duelu počnúc prvým zapísaným duelom súťaže.

Výnimky (3 body)

Vytvorte nekontrolovanú výnimku `NeznamySutaziaciException` a vhodne ju použite aspoň v jednej metóde.

Nový formát (dokopy 6 bodov):

Objavila sa verzia programu, ktorá zapisuje výsledky duelov do súboru v inom formáte. Namiesto času v sekundách je čas od začiatku súťaže do ukončenia duelu zapísaný vo forme reťazca vo formáte "H:M:S", kde H je počet hodín, M počet minút a S počet sekúnd. Upravte príslušné metódy (`zoStringu`, `zoSuboru`) tak, aby načítavanie podporovalo aj takýto formát (t.j. má byť podporovaný pôvodný aj nový formát).