



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.

PAZ rádio

Motivácia: Istá podnikavá korytnačka sa rozhodla založiť si rádio. Keďže nemá šajnu, ako sa také rádio vedie, rozhodla sa celý proces čo najviac okopírovať zo slovenských rádii. Vytvorila si program, ktorý jej automatizovane zaznamenáva, čo ktoré rádio práve vysiela. S analýzou tohto veľkého prúdu dát jej už však pomôžete vy.



Pohľad analytika: Pri implementácii budeme potrebovať:

- triedu `ZvukovyUsek`, ktorá reprezentuje jeden súvislý časový úsek, v ktorom dané rádio vysiela tematicky samostatný zvukový prúd (napr. správy, dopravné informácie, hudba, vstup moderátora, reklama)
- triedu `ZoznamUsekov`, ktorá bude uchovávať nejaký zoznam zvukových úsekov (napr. všetky zaznamenané).

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

- casOd** (poradie sekundy začiatku zvukového úseku od polnoci **prvého dňa**, v ktorom sa začali zvukové úseky zaznamenávať; napr 925 znamená čas 0h 15min 25s prvého dňa)
- casDo** (poradie sekundy konca zvukového úseku od polnoci prvého dňa, v ktorom sa začali zvukové úseky zaznamenávať)
- radio** (názov rádia napr. Fun rádio, Rádio Expres, Rádio Košice,...)
- typZvuku** (práve jedna z 5 hodnôt: správy, doprava, reklama, slovo, hudba)
- interpret** (meno interpreta – iba ak ide o hudbu; napr. Adele, Katy Perry, One Direction,..)
- nazovSkladby** (názov skladby – iba ak ide o hudbu, napr. Hello, Dark Horse, ...)
- jazykSkladby** (jazyk skladby, v ktorej je skladba interpretovaná – iba ak ide o hudbu, napr. SK, EN, CZ, FR,...)

Upozornenie: Zadanie triedy `ZvukovyUsek` 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.ZoznamUsekov`, ktorá bude uchovávať nejaký zoznam zvukových úsekov (napr. zoznam všetkých pesničiek v Rádiu Slovensko za daný deň).

Konštruktory a pridávanie zvukových úsekov (3 body dokopy – povinné):

- **public** ZvukovyUsek(long casOd, long casDo, String radio, String typZvuku) – použije sa na vytvorenie zvukového úseku iného než je *hudba*.
- **public** ZvukovyUsek(long casOd, long casDo, String radio, String interpret, String nazovSkladby, String jazykSkladby) – použije sa na vytvorenie zvukového úseku typu *hudba*
- **public void** pridaj(ZvukovyUsek usek) – inštančná metóda v triede ZoznamUsekov, ktorá pridá záznam o zvukovom úseku do zoznamu úsekov.

Práca so súbormi (povinné):

V triede ZvukovyUsek:

- **public static** ZvukovyUsek zoStringu(String popis) – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy ZvukovyUsek. Parameter je **String** v tvare "casOd \t casDo \t radio \t typZvuku", resp. " casOd \t casDo \t radio \t typZvuku \t interpret \t nazovSkladby \t jazykSkladby", ak typZvuku je *hudba* (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 o zvukovom úseku (1 bod).

V triede ZoznamUsekov:

- **public static** ZoznamUsekov zoSuboru(String nazovSuboru) – statická metóda, ktorá z uvedeného súboru prečíta zoznam zvukových úsekov, pričom v každom riadku bude popis jedného zvukového úseku (4 body).
- **public void** uloz(String nazovSuboru) – uloží všetky záznamy o všetkých zvukových úsekoch v zozname do súboru v tvare, ktorý vie spracovať metóda zoSuboru(String nazovSuboru) (3 body).
- **public** String toString() – vráti reťazec vhodne reprezentujúci všetky zvukové úseky v zozname (1 bod).

Inštančné metódy triedy ZoznamUsekov:

- **public long** trvanieUsekovTypu(String typ) – vráti celkové trvanie zvukových úsekov v zozname zadaného typu (1 bod)
- **public double** podielSlovenskejHudby() – vráti koľko percent z úsekov typu *hudba* má jazyk skladby "SK" (2 body).
- **public** List<String> zoznamRadii() – vráti zoznam všetkých rádii bez opakovania, ktoré sú v zozname zachytené (3 body).
- **public** ZvukovyUsek najdlhsiUsek(String typ) – vráti zvukový úsek daného typu, ktorý má najdlhšie trvanie zo zoznamu (4 body).
- **public** ZoznamUsekov fulltext(String hladane) – vráti zoznam úsekov, ktorý obsahuje iba tie úseky, ktorých interpret alebo nazovSkladby obsahuje v sebe ako podreťazec reťazec určený parametrom hladane. (5 bodov). Hint: metóda String.indexOf
- **public** ZoznamUsekov usekyRadiavIntervale(String radio, Integer denOd, Integer denDo) – vráti zoznam zvukových úsekov daného rádia v danom intervale dní. Napr. usekyRadiavIntervale("Fun radio", 1, 3) vráti zoznam zvukových úsekov Fun rádia z prvého, druhého a tretieho dňa od začiatku zaznamenávania (dni číslujeme od 1). Všetky vstupné hodnoty môžu byť aj **null**, vtedy sa dané kritérium neuplatňuje, napr. usekyRadiavIntervale(null, 2, null) vráti zoznam zvukových úsekov všetkých rádii od druhého dňa až po koniec nášho zoznamu záznamov. Ak zvuk presahuje hranicu polnoci, treba ho niekde zarátať (buď do oboch dní alebo do dňa, v ktorom zvuk začal). Poznámka: jeden deň má 86400 sekúnd. (5 bodov).
- **public** Set<String> oblubeniInterpreti() – vráti interpretov, ktorí boli „hraní“ v každom rádiu vyskytujúcim sa v zozname (6 bodov).
- **public** String najproduktivnejsiInterpret() – vráti meno interpreta, ktorý mal odvysielaných najviac rôznych skladieb (6 bodov). Hint: časová efektívnosť nie je dôležitá, zväzťe vytvorenie

pomocnej metódy, ktorá pre každého interpreta vráti počet jeho rôznych skladieb. Bonus: +3 body, ak si vystačíte s jedným prechodom cez zoznam úsekov.

- **public** Map<String, Double> venovanyCas(String typ) – vráti pre každé rádio percento času, ktorý venuje dané rádio vysielaniu daného typu záznamu (6 bodov).
- **public** Set<String> dopravneRadio() – vráti rádiá, ktoré vysielajú dopravné informácie aj inokedy ako po správach. Môžete predpokladať chronologické usporiadanie záznamov podľa času začiatku záznamu (6 bodov) Bonus: +3 body za metódu na usporiadanie List-u alebo poľa zvukových záznamov podľa času začiatku (zide sa, keď predpoklad usporiadania neplatí).
- **public void** najoblubenejsieSkladby(int pocet) – vypíše príslušný počet najobľúbenejších skladieb v tvare "interpret: názov skladby (x krát)", ktoré sa hrali najviac krát spolu za všetky rádiá v zozname (7 bodov). Pozor na to, aby metóda fungovala aj v prípade, keď ešte nemáme zaznamenaných veľa zvukových úsekov.
- **public boolean** neopakujeSkladbyPocasDna(String radio) – vráti, či rádio počas dňa nehrá nejakú skladbu (interpret+názov) viac ako raz. Predpokladajte chronologické usporiadanie záznamov (5-10 bodov podľa efektívnosti implementácie).
- **public int** ibaHudba(long start) - vráti maximálny počet sekúnd od zadaného času start, dokedy bolo možné v ľubovoľnej chvíli preladíť aspoň na jedno rádio, na ktorom hrala hudba. (4 body: stačí časovo extra neefektívne riešenie analyzujúce každú sekundu po čase start).

Triedenie a komparátor (dokopy 5 bodov):

Automatický zaznamenávač hudobných úsekov zapisuje úseky zo všetkých rádií tak, ako sa vyskytnú, do toho istého zoznamu, čo je neprehľadné. Vytvorte preto metódu, ktorá zotriedi zoznam najprv podľa názvu rádia a až potom podľa času.

Vytvorte triedu RadiovyKomparator implementujúcu `java.util.Comparator<ZvukovyUsek>` s metódou (3 body):

- **public int** compare(ZvukovyUsek zu1, ZvukovyUsek zu2) – porovná zvukové úseky podľa názvu rádia lexikograficky a v rámci každého rádia podľa času vzniku. (+1 bod, ak lexikografické = abecedné usporiadanie bude rešpektovať pravidlá slovenského jazyka)

V triede ZoznamUsekov implementujte inštančnú metódu (2 body):

- **public void** zoradPodlaRadio() – usporiada zvukové úseky podľa názvu rádia lexikograficky a v rámci každého rádia podľa času vzniku.

Výnimky (3 body)

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