



Záverečný test

Zadanie



Ústav informatiky
Prírodovedecká fakulta
UPJS v Košiciach

Dvakrát meraj (rozmyšľaj), raz rež (programuj)

Dôležité pravidlá a informácie (viac na stránke predmetu):

- 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 (nie až v závere testu),
- **funkčnosť každej metódy musí byť preukázaná spustením na vami vytvorenom testovacom vstupe, nespustiteľné metódy neumožňujú zisk príslušných bodov,**
- všetky inštančné premenné musia byť neverejné.

YouTube tracker

Motivácia: Skupina študentov na PAZku sa rozhodla, že vytvorí plugin do prehliadača a zároveň nastavbu mobilnej appky, ktorá bude sledovať a analyzovať pozieranie videí na YouTube. Nie, že by to Google (vlastník YouTube) nerobil, ale môže byť zaujímavé vidieť vlastné štatistiky.

Pohľad analytika: Pri implementácii aplikácie budeme potrebovať:

- triedu **Video**, ktorá reprezentuje jedno pozretie nejakej časti videa,
- triedu **ZoznamVidei**, ktorá bude uchovávať zoznam pozretí videí na YouTube.



Poznámka: Názvy ako `PozretieVidea` a `ZoznamPozretiVidea` by boli lepšie, ale kvôli zjednodušeniu používame kratšie názvy.

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

- **id** (identifikátor videa, napr. „hER0Qp6QJNU“),
- **kanal** (názov kanála, v ktorom sa video nachádza, napr. „UINF“),
- **nazov** (názov videa, napr. „PAZ1a: For-cykly“),
- **dlzka** (trvanie videa v sekundách, napr. 600)
- **datum** (dátum začiatku pozretia videa vo formáte DD.MM.RRRR, napr. „19.12.2018“ alebo „1.1.2019“),
- **cas** (čas začiatku pozretia videa vo formáte HH:MM:SS, napr. „13:05:45“, alebo „4:52:0“),
- **odSekundy** (od ktorej sekundy sa začalo video pozerať, napr. 124),
- **poSekundu** (do ktorej sekundy sa video pozeralo, napr. 520),
- **tagy** (čiarkami oddelený zoznam tagov, ktoré aplikácia pridala ku pozretiu videa, napr. „education, programming, Java“)

Upozornenie: Zadanie pre triedu `Video` 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.ZoznamVidei`, ktorá bude uchovávať zoznam pozretí videí.

Konštruktory a pridávanie videí do zoznamu (3 body dokopy – povinné):

- **public** Video(String id, String kanal, String nazov, **int** dlzka, String datum, String cas, **int** odSekundy, **int** poSekundu) – použije sa na vytvorenie záznamu o pozretí videa bez tagov,
- **public** Video(String id, String kanal, String nazov, **int** dlzka, String datum, String cas, **int** odSekundy, **int** poSekundu, String tagy) – použije sa na vytvorenie záznamu o pozretí videa s tagmi,
- **public void** pridaj(Video video) – inštančná metóda v triede ZoznamVidei, ktorá pridá záznam o pozretí videa do zoznamu videí.

Práca so súbormi (povinné):

V triede Video:

- **public static** Video zoStringu(String popis) – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy Video. Parameter je String v tvare "id \t kanal \t nazov\t dlzka \t datum \t cas \t odSekundy \t poSekundu", resp. "id \t kanal \t nazov\t dlzka \t datum \t cas \t odSekundy \t poSekundu \t tagy", ak video je otagované (3 body);
Poznámka: Znak \t je neviditeľný znak tabulátora. Scanner-u môžete povedať, že oddeľovač má byť tabulátor zavolaním jeho metódy useDelimiter("\t"). Medzera pre a za \t sú len kvôli zlepšeniu čitateľnosti zadania, v reťazci reálne nie sú.
- **public** String toString() – vráti reťazec vhodne reprezentujúci údaje o pozretí videa (1 bod).

V triede ZoznamVidei:

- **public static** ZoznamVidei zoSuboru(String nazovSuboru) – statická metóda, ktorá z uvedeného súboru prečíta zoznam pozretí videí, pričom v každom riadku bude popis jedného pozretia videa (4 body).
- **public void** uloz(String nazovSuboru) – uloží všetky pozretia videí v zozname do súboru v tvare, ktorý vie spracovať metóda zoSuboru(String nazovSuboru) (3 body).
- **public** String toString() – vráti reťazec vhodne reprezentujúci všetky videá v zozname pozretí videí (1 bod).

Inštančné metódy triedy Video:

- **public** String vytvorOdkaz() - vráti URL adresu na zobrazenie videa. Táto adresa vyzerá takto: <https://youtu.be/iyjEVopS16s?t=8143> - prvá vyznačená časť je identifikátor videa a druhá časť je sekunda od začiatku videa, v ktorej sa má začať prehrávanie (t.j. sekunda, v ktorej začína pozretie videa). Ak sa video má prehrať od začiatku (čas je 0), časť ?t=0 sa v URL adrese nenachádza (1 bod).

Inštančné metódy triedy ZoznamVidei:

Ak niektorá z metód nevie vrátiť referenciu na objekt s požadovanými vlastnosťami, metóda nech vráti **null**.

- **public int** vratCasPozerania(String datum) - vráti celkový čas v sekundách, ktoré sa strávili pozeraním videí v daný dátum, resp. začali sa pozerat' v daný dátum (1 bod).
- **public** String najdiNajdlhsieVideo() – nájde najdlhšie video (uvažujeme dĺžku celého videa), aké sa nachádza v zozname (3 body).
- **public** List<String> vratIdVidei() – vráti zoznam identifikátorov videí, pričom každý identifikátor je v zozname uvedený len raz (3 body).

- **public** `ZoznamVidei vratDlhsiePozretia(int percentoPozretia)` – vráti nový zoznam pozretí videí, ktorý budú tvoriť len tie pozretia videa, kedy sa pozrelo aspoň zadané percento z celkovej dĺžky videa. Ak je trebárs percento pozretia 25, chceme len tie pozretia, kedy sa bez prerušenia pozrelo aspoň 25% z celkovej dĺžky videa (3 body).
- **public** `Set<String> vratVideaSTagom(String tag)` – vráti množinu identifikátorov videí, ktoré obsahujú zadaný tag. Pozor, ak hľadáte tag „program“ a video je otagované tagom „programming“, či viacslovné tagy (5 bodov).
- **public** `String najdiNajoblubenejsiKanal()` – vráti názov kanála, na ktorého videách sa strávil najväčší čas pozeraním. Kvôli jednoduchosti predpokladáme, že názvy kanálov sa nemenia (5 bodov).
- **public int** `zistiPocetPremenovanych()` – vráti počet videí, ktoré boli premenované. Premenovanie vieme zistiť tak, že pri jednom pozretí tohto videa (podľa identifikátora) malo jeden názov a pri ďalšom už iný názov. Ak sa video premenovávalo viac ráz, započítavame ho len raz. (6 bodov).
- **public double** `zistiPozretieVidea(String idVidea)` – vráti percento celkového pozretia videa so zadaným identifikátorom. Upozorňujeme, že časové intervaly pozretia sa môžu prekrývať. Teda ak sme mali pozretie (od: 10, po: 20) a pozretie (od:5, po: 15), tak sme celkovo videli nie 20 sekúnd z videa, ale len 15 sekúnd videa, keďže úsek (od: 10, po: 15) sme videli viac ráz. Hint (neefektívne ale implementačne jednoduché riešenie): pre každú sekundu videa si môžete pamätať, či ste ju už videli (9 bodov).
- **public** `String vratIdNajoblubenejsiehoVidea()` – vráti identifikátor najobľúbenejšieho videa, t.j. videa, ktoré sa pozrelo najväčší počet ráz, pričom pozretia videa v jeden deň započítame len raz (9 bodov).
- **public int[]** `vratHistogramPozerania()` – vráti 24 prvkové pole, ktoré na indexe h bude obsahovať celkový čas v sekundách strávený pozeraním videí v danú hodinu h dňa. Ak pozeranie videa začne o 13:55:00 a skončí o 15:35:00 minút, tak 5 minút započítavame do 13. hodiny, 60 minút do 14. tej hodiny a 35 minút do 15. tej hodiny (4 body za zjednodušenú verziu, kde uvážite len čas začiatku pozerania videa; 8 bodov za verziu s hintom z úlohy `zistiPozretieVidea`, 16 bodov za efektívne riešenie).
- (6 bodov) Upravte svoje riešenia tak, aby korektne fungovali aj v prípade, keď reťazce s dátumami nie sú normalizované. Teda po úprave by reťazce “01.01.2019”, “1.01.2019”, “01.1.2019”, či “1.1.2019” mali reprezentovať rovnaký dátum.

Triedenie a komparátor (dokopy 5 bodov):

Vytvorte triedu `CasPozretiaKomparator` implementujúcu rozhranie `java.util.Comparator<Video>` s metódou (3 bod):

- **public int** `compare(Video vide1, Video vide2)` – porovná pozretia chronologicky podľa času začiatku pozerania videa.

V triede `ZoznamVidei` implementujte inštančnú metódu (2 body):

- **public void** `usporiadajChronologicky()` – usporiada pozretia videí v zozname chronologicky.

Výnimky (2 body)

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

Statická metóda triedy Video:

- **public static** String ziskajIdZUrl(String url), extrahuje hodnotu parametra v (identifikátor videa) z URL adresy (5 bodov). Parametre sa nachádzajú za znakom ? a sú oddelené znakmi &. Každý parameter je zapísaný v tvare meno=hodnota. Príklady URL s identifikátorom videa:
 - <https://www.youtube.com/watch?v=iyjEVopSl6s&t=8143>
 - <https://www.youtube.com/watch?v=iyjEVopSl6s>
 - <https://www.youtube.com/watch?v=iyjEVopSl6s&feature=youtu.be&t=8143>

Poznámka: Ak chcete ako oddeľovač (delimiter) tokenov pre Scanner nastaviť bodku, použite `useDelimiter("\\\\.")`