



Záverečný test

Zadanie

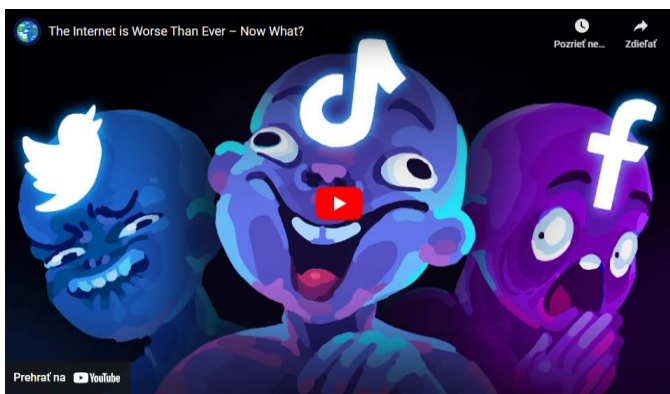


Dvakrát meraj (rozmyšľaj), raz rež (programuj)

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.

(Anti) Socialne siete alebo E-LIFE



zdroj: youtube.com/@kurzgesagt

Motivácia: Hedviga prišla počas Vianoc na to, že trávi priveľa času na sociálnych sieťach. Vôbec ju to nenaplnia, žerie to čas a vyrušuje ju to od práce aj zábavy. Bolí ju krk ako stále pozerá do mobilu. Často s nikým nekomunikuje ale len bezcieľne scrolluje. Teda aj sociálne siete sú bez sociálneho kontaktu. Preto sa rozhodla, že si začne monitorovať čas na všetkých sociálnych sieťach aj na všetkých zariadeniach. Okrem toho chce zistiť či sú pre ňu sociálne siete viac o komunikácii alebo len o bezcieľnom scrollovaní. Tým chce zlepšiť svoj prístup k životu a lepšie sa sústrediť na všetko podstatné.

Pohľad analytika: Pri implementácii aplikácie budeme potrebovať:

- triedu **Aktivita**, ktorá reprezentuje jeden prístup na sociálnu sieť,
- triedu **Elife**, ktorá bude uchovávať zoznam všetkých aktivít, ktoré prebehli.

Zadanie: V balíku `sk.upjs.finalTerm` vytvorte triedu **Aktivita** obsahujúcu dátové položky prístupné cez `getter` (a podľa uváženia aj modifikovateľné cez `setter`):

- **datum** (dátum v tvare DD.MM.RRRR)
- **casZaciatku** (čas začiatku napr. 9:00:00 resp. môžete uvažovať 9:0:0 pre jednoduchosť)
- **casKonca** (čas konca v rovnakom tvare ako `casZaciatku`, pozn. pre jednoduchosť predpokladáme, že každý prístup je len v rámci jedného dňa)
- **oznacenieSieť** (označenie siete napr. FB, IG, YT, email, TikTok alebo iné)
- **spokojnosť** (číslo z rozsahu 0.0 až 10.0 ktoré vyjadruje nakoľko je spokojná Hedviga s takto stráveným časom, napr. 10.0 vyjadruje, že to bolo rozumne využitie času alebo zábava)
- **zariadenie** (názov zariadenia na ktorom bol prístup napríklad „DESKTOP-gH1bt9“, „VYROBCA MODEL Android W654ta21G“)
- **partner** (meno alebo mená s kým sa komunikovalo, ak ide o využitie na komunikáciu, napr. „Jožko Mrkvička“ alebo „Mária Nováková, Janko Hraško“ ak ide o komunikáciu s viacerými osobami naraz)

Upozornenie: Zadanie pre triedu **Aktivita** predpisuje dátové položky prístupné cez `getter`. Aké privátne inštančné premenné použijete na uloženie týchto dátových položiek je na vašom rozhodnutí.

Ďalej vytvorte aj triedu `sk.upjs.finalTerm.Elife`, ktorá bude uchovávať zoznam Aktivít, pričom **poradie nemusí byť garantované** podľa času kvôli synchronizácii medzi zariadeniami.

Konstruktory a pridávanie Aktivít do Elife (povinné):

- **public** Aktivita(String datum, String casZaciatku, String casKonca, String oznacenieSiete, **double** spokojnost, String zariadenie, String partner)– použije sa na vytvorenie Aktivít kedy sa s niekým komunikovalo.
- **public** Aktivita(String datum, String casZaciatku, String casKonca, String oznacenieSiete, **double** spokojnost, String zariadenie) – použije sa na vytvorenie aktivít bez komunikácie.
- **public void** pridaj(Aktivita aktivita) – inštančná metóda v triede Elife, ktorá pridá prístup do zoznamu prístupov.

Práca s reťazcami a súbormi (povinné):

V triede Aktivita:

- **public static** Aktivita zoStringu(String popis) – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy Aktivita. Parameter je String v tvare "datum \t casZaciatku \t casKonca \t oznacenieSiete \t spokojnost \t zariadenie \t partner", resp. "datum \t casZaciatku \t casKonca \t oznacenieSiete \t spokojnost \t zariadenie ", ak ide o aktivitu bez partnera.
Poznámka: Znak \t je neviditeľný znak tabulátora. Scanner-u môžete povedať, že oddeľovač má byť tabulátor zavolaním jeho metódy useDelimiter("\t"). Medzera pred a za \t sú len kvôli zlepšeniu čitateľnosti zadania, v reťazci reálne nie sú.
- **public** String toString() – vráti reťazec vhodne reprezentujúci údaje o jednej aktivite.

V triede Elife:

- **public static** Elife zoSuboru(String nazovSuboru) – statická metóda, ktorá z uvedeného súboru prečíta zoznam aktivít, pričom v každom riadku bude popis jednej aktivít.
- **public void** uloz(String nazovSuboru) – uloží všetky aktivít zoznamu aktivít do súboru v tvare, ktorý vie spracovať metóda zoSuboru(String nazovSuboru).
- **public** String toString() – vráti reťazec vhodne reprezentujúci všetky aktivít uchované v zozname prístupov.

Za povinnú časť bude udelených 15 bodov. Nasledujúce (nepovinné) úlohy môžete riešiť v ľubovoľnom poradí. Ale riešenie jednej môže zjednodušiť nasledujúce.

Inštančné metódy triedy Aktivita:

- [1b] **public int** dlzka() – vráti koľko sekúnd trvala aktivita.
- [1b] **public** List<String> partneri() – vráti list partnerov, pozn. ak je to aktivita bez partnera, tak list má veľkosť nula, ak je tam iba jeden partner tak list ma veľkosť jedna.

Inštančné metódy triedy Elife:

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

- [1b] **public int** pocetSpokojnejšíchAktivít(**double** spokojnost) – vráti koľko aktivít malo spokojnosť väčšiu rovnú ako spokojnosť zadaná parametrom.
- [1b] **public** List<Aktivita> aktivítZoZariadenia(String zariadenie) – metóda vráti zoznam všetkých aktivít zo zariadenia ktoré je zadané parametrom.
- [1b] **public int** pocetSocialnýchSietí() – metóda vráti počet rôznych použitých označení sietí.
- [1b] **public** Aktivita najlepšíAktivita() – metóda vráti referenciu na aktivitu z Elife-u, ktorá má najväčšiu spokojnosť.
- [1b] **public** Aktivita najdlhšiaAktivita() – metóda vráti referenciu na aktivitu z Elife-u, ktorá mala najdlhšie trvanie v sekundách.

- [2b] **public** List<Aktivita> nocneAktivity(String zariadenie) – metóda vráti zoznam všetkých aktivít ktoré majú čas začiatku po 21:59:59 alebo čas konca pred 6:00:00.
- [2b] **public** List<String> zoznamPartnerov() – vráti zoznam partnerov s ktorými sa komunikovalo, pričom sa v zozname žiaden nenachádza duplicitne.
- [2b] **public boolean** scrollovacieZariadenie(String zariadenie) – metóda vráti true práve vtedy ak počet aktivít, ktoré boli z daného zariadenia bez partnera je väčší ako počet aktivít z daného zariadenia s partnerom.
- [3b] **public** Map<String, Integer> pocetAktivitZoZariadenia(String osoba) – metóda vráti mapu kde je pri každom zariadení, počet aktivít ktoré z neho boli realizované.
- [3b] **public** Map<String, Integer> casNaJednotlivychSocialnychSietach(String osoba) – metóda vráti mapu kde je pri každej sociálnej sieti počet sekúnd strávený na nej.
- [3b] **public double** vazenyPriemerHodnoteni() – metóda vážený priemer hodnotení aktivít. Napr. Ak máme dve aktivity, prvá od 19:00:00 do 19:00:05 (dĺžka 5 sekúnd) s hodnotením 8.7 a druhá od 10:00:00 do 10:01:00 (dĺžka 60 sekúnd) s hodnotením 7.2, tak ich vážený priemer hodnotení vypočítame ako $(5 * 8.7 + 60 * 7.2) / (5 + 60)$.
- [4b] **public** Elife vRozmedziDni(String datumOd, String datumPo) – metóda vráti referenciu na novovytvorenú inštanciu triedy Elife, ktorá obsahuje iba aktivity, ktoré majú dátum z intervalu <datumOd, datumPo>.
- [3+2+1+2b] **public int[]** histogram() – metóda vráti pole veľkosti 24, kde index i zodpovedá hodine i počas dňa. Na indexe i sa nachádza počet sekúnd koľko trvali tie aktivity, ktoré majú čas začiatku v danú hodinu. +2b Ak ide o aktivitu, ktorá je na prelome hodín, tak sa jej trvanie rozdelí na jednotlivé hodiny. +1b Za rozdelenie aktivít, ktoré sa realizujú cez polnoc. +2b Ak aktivita zasahuje do troch alebo viac hodín.
- [5b] **public** String topPartner() – metóda vráti meno partnera s ktorým sa komunikovalo najviac času v sekundách. Pričom ak sa komunikovalo naraz s viacerými partnermi, tak tento čas sa ráta ako podiel počtu partnerov.
- [4+2b] **public** Map<String, Integer> pocetSietiPartnera() – metóda vráti mapu, kde je kľúčom meno partnera a hodnotou je počet sociálnych sieti, na ktorých spolu komunikovali. +2b za efektivitu.
- [6b] **public** Map<Aktivita, Aktivita> hookUp() – metóda vráti mapu, kde je kľúčom aktivita na jednom zariadení a hodnotou je druhá aktivita, ktorá je na inom zariadení na tej istej sociálnej sieti a navyše druhá aktivita začína v ten istý deň najviac 30 sekúnd po konci prvej aktivity. Pozn. Ak je viacero takých druhých aktivít, tak hodnotou je ľubovoľná z nich.
- [7b] **public void** casVTop5Dnoch() – Metóda vypíše 5 dátumov, kedy sa strávilo najviac sekúnd aktivitami. Tieto aktivity sú usporiadané podľa počtu sekúnd strávených aktivitami. Za každým dátumom je aj výpis koľko sekúnd sa strávilo s aktivitami v daný deň. Pre jednoduchosť predpokladáme, že každá aktivita začína a končí v ten istý deň.

Nezaradené úlohy

- [3b] Usporiadajte všetky aktivity chronologicky (podľa dátumu a času začiatku).
- [2b] Pridajte nekontrolovanú výnimku Spokojnost'MimoRozsahuException, ktorá sa bude vhodne týkať prípadu, ak je hodnotenie mimo rozsahu <0,10>. Výnimku použite na vhodnom mieste.