



# Záverečný test

## Zadanie



Dvakrát meraj (rozmýšľaj), 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.

## Renovácia



**Motivácia:** Julio sa pustil do rekonštrukcie. Ale počas pandémie išli ceny materiálov aj cena práce hore. Keďže má kopol kamarátov ktorí sú ochotní a niektorí majú blízko aj k „stavbarine“ tak sa skúsi pustiť do rekonštrukcie s pomocou rodiny, kamarátov ale aj platených profesionálov. Evidovať cenu materiálov je vďaka účtenkám jednoduché. Aby vedel koľko práce sa vykonalo a kto každý mu pomohol, tak sa rozhodol to evidovať samostatne. Pomôžte Julovi s implementáciou aplikácie na evidenciu všetkých pracovníkov a prác, ktoré vykonali.

zdroj: <https://www.topbyvanie.sk>

**Pohľad analytika:** Pri implementácii aplikácie budeme potrebovať:

- triedu `StavebnyUkon`, ktorá reprezentuje jeden úkon ktorý sa vykonal,
- triedu `Renovacia`, ktorá bude uchovávať zoznam vykonaných stavebných úkonov ktoré sa vykonali počas renovácie.

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

- **ukon** (názov stavebného úkonu ktorý bol vykonávaný napr. maľovanie stien, strhávanie kachličiek, vynášanie sutiny, ukladanie podlahy, kontrola elektroinštalácie alebo iné)
- **datum** (dátum úkonu, vo formáte DD.MM.RRRR, napr. "25.01.2022" – v dátume sú vždy úvodné nuly doplnené tak, aby deň aj mesiac boli dvojčiferné),
- **dlzka** (počet začatých hodín strávených daným úkonom),
- **meno** (meno človeka ktorý vykonával úkon, napr. "Jožko Mrkvička")
- **vztah** ("kamarát", "rodina", "profesionál" alebo iné podľa toho aký je vzťah osoby ku Julovi)
- **cenaJednotka** (napr. euro ak sa platilo v peniazoch alebo pizza, pivo, guľáš, obed, nič alebo iné)
- **cenaPocet** (suma v eurách ktoré boli zaplatené alebo počet litrov vypitého piva alebo počet zjednených guľášov)

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

Niektoré úkony boli vykonané úplne zadarmo a preto nemajú vyplnené položky `cenaJednotka` a `cenaPocet`.

Ďalej vytvorte aj triedu `sk.upjs.finalTerm.Renovacia`, ktorá bude uchovávať zoznam úkonov.

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

- **public** `StavebnyUkon(String ukon, String datum, int dlzka, String meno, String vztah, String cenaJednotka, double cenaPocet)` – použije sa na vytvorenie záznamu o úkone s doplňujúcim popisom.
- **public** `StavebnyUkon(String ukon, String datum, int dlzka, String meno, String vztah)` – použije sa na vytvorenie záznamu o úkone vykonanom zadarmo.
- **public void** `pridaj(StavebnyUkon stavebnyUkon)` – inštančná metóda v triede `Renovacia`, ktorá pridá vykonaný úkon do zoznamu úkonov.

### Práca so súborami (povinné):

V triede `StavebnyUkon`:

- **public static** `StavebnyUkon zoStringu(String popis)` – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy `StavebnyUkon`. Parameter je `String` v tvare `"ukon \t datum \t dlzka \t meno \t vztah \t cenaJednotka \t cenaPocet"`, resp. `ukon \t datum \t dlzka \t meno \t vztah`, ak ide o úkon vykonaný zadarmo (3 body);  
*Poznámka:* Znak `\t` je neviditeľný znak tabulátora. Scanner-u môžete povedať, že oddeľovač má byť tabulátor zavolaním jeho metódy `useDelimiter("\t")`. Farebnosť, 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 úkone (1 bod).

V triede `Renovacia`:

- **public static** `Renovacia zoSuboru(String nazovSuboru)` – statická metóda, ktorá z uvedeného súboru prečíta zoznam úkonov, pričom v každom riadku bude popis jedného úkonu (4 body).
- **public void** `uloz(String nazovSuboru)` – uloží všetky úkony z renovácie do súboru v tvare, ktorý vie spracovať metóda `zoSuboru(String nazovSuboru)` (3 body).
- **public** `String toString()` – vráti reťazec vhodne reprezentujúci všetky úkony uchované v renovácii (1 bod).

Nasledujúce úlohy môžete riešiť v ľubovoľnom poradí. Ale riešenie jednej môže zjednodušiť nasledujúce.

### Inštančné metódy triedy `Renovacia`:

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

- **public int** `vratCelkovuDlzku()` – vráti koľko hodín sa strávilo úkonmi (1 bod).
- **public int** `najdlhsiUkon()` – vráti koľko hodín trval najdlhší úkon (1 bod).
- **public** `List<String> vratNazvyUkonov()` – vráti názvy všetkých úkonov, vo vrátenom zozname sa môže každý názov nachádzať najviac raz (2 body).
- **public** `List<String> vratMenaPodlaVztahu(String vztah)` – vráti mená všetkých ľudí, ktorí sa venovali nejakému úkonu a majú vzťah zadaný parametrom, vo vrátenom zozname sa môže každé meno vyskytovať najviac raz (1 bod).
- **public int** `casUkonovPocasDna(String datum)` – vráti počet hodín úkonov ktoré sa vykonali počas dňa ktorý je zadaný parametrom (2 body).
- **public boolean** `odbornyUkon(String nazovUkonu)` – metóda vráti `true` vtedy keď úkon s názvom zadaným parametrom vykonával **vždy** iba profesionál (2 body).
- **public** `Map<String, Integer> stravenyCas()` – vráti mapu, kde je každému menu priradený celkový čas ktorý strávil úkonmi (2 body).
- **public** `Map<String, Double> celkovaCena()` – vráti mapu, kde je každej jednotke ceny priradené koľko sa jej spotrebovalo napr. pivo 10,5; euro 752,23; gul'áš 17 (3 body).

- **public** List<String> rychleUkony(int limit) – vráti zoznam názvov úkonov ktorých sumárna dĺžka nepresiahla čas zadaný parametrom limit. Pozn. ak sa maľovalo raz 3 hodiny a potom 5 hodín tak sumárny čas maľovania je 8 hodín. (3 body).
- **public** Map<String, String> ukonySJedinyMvykonavatelom() – vráti mapu názvov úkonov, kde je názvu úkonu priradený jeho vykonávateľ. Ak sa úkonu s daným názvom venovali aspoň dvaja rôzni ľudia, tak sa v mape nenachádza (4 body). +2 bod za efektivitu podľa počtu prechodov zoznamom stavebných úkonov.
- **public** List<String> skymSaStretol(String meno) – vráti zoznam mien ľudí s ktorými sa mohol stretnúť človek ktorého meno je zadané parametrom. To znamená, že obaja vykonávali nejaký úkon v rovnaký dátum. (4 body). +2 body za efektivitu.
- **public** Map<String, Double> percentualnyPodielUkonovZadarmoPodlaVztahu() – vráti mapu, kde je ku každému druhu vzťahu "rodina", "kamarát" alebo ďalšie druhy priradené, koľko percent celkového času aktivít sa vykonalo zadarmo (5 body). +1 bod za zaokrúhlenie percent na 2 desatinné miesta.
- **public** Renovacia vratRenovacijuZaObdobie(String odDatumu, String poDatum) – metóda vráti referenciu na nový objekt triedy Renovácia, ktorá obsahuje iba úkony za obdobie určené parametrami (vrátane dní určených parametrami) (3 body).
- **public** Map<String, Integer> pocetZrucnostiLudi() – vytvoríte mapu, kde je každému menu človeka priradené koľkým rôznym úkonom sa venoval (napr. ak Jožko Mrkvička dvakrát maľoval, čo je v dvoch stavebných úkonoch, a Mária Nováková, postavila stenu, strhávala kachličky a trikrát vynášala sutinu, čo je v piatich stavebných úkonoch tak mapa bude nasledovná Jožko Mrkvička 1, Mária Nováková 3). (6 bodov).
- **public** Map<String, Integer> casovaNarocnostUkonov() – vráti mapu, kde je každému názvu úkonu priradené aké celkové množstvo času v hodinách sa mu venovalo (3 body).
- **public** List<String> top5Ukonov() – vráti zoznam piatich úkonov ktoré zabrali najviac času. (4 body). Pozn. Bez korektného riešenia predchádzajúcej úlohy nemôžete riešiť túto.
- **public int** najviacDniZaSebou(String ukon) – metóda vráti najviac koľko dní za sebou (bez prerušenia) sa vykonával úkon ktorého názov je zadaný parametrom. Pozn. pre jednoduchosť predpokladajte, že každý mesiac má 31 dní. (6 bodov). Za verziu ktorá funguje len v rámci jedného mesiaca je polovica bodov.

### **Triedenie (4 body)**

Vytvoríte triedu PorovnavacUkonov, ktorá implementuje rozhranie java.util.Comparator<StavebnyUkon> s príslušnou metódou compare, aby po aplikovaní triediaceho algoritmu boli zotriedené aktivity podľa dátumu a času začiatku (4 body).

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