



Závěrečný test Zadanie



Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

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 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.



Múzeum voskových figurín

Motivácia: Najznámejšie múzeum voskových figurín v Londýne *Madame Tussauds* bolo založené v roku 1835. Svoju voskovú figurínu majú mnohé slávne osobnosti z oblasti hudby, športu, filmu a pod. Múzea voskových figurín dnes vieme nájsť vo viacerých mestách. Vašou úlohou je pripraviť podklad pre informačný systém múzea, kde budú evidované všetky figuríny, ktoré sa tam nachádzajú, aj tie, ktoré sú objednané a ešte len budú vyrobené a privezené.

Pohľad analytika: Pri implementácii budeme potrebovať:

- triedu *Figurina*, ktorá uchováva informácie o jednej figuríne,
- triedu *Muzeum*, ktorá bude uchovávať zoznam figurín.

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

- **meno** – meno osobnosti, ktorú vyobrazuje figurína,
- **narodenie** – uvádza mesiac a rok narodenia osobnosti vo formáte MM/RRRR (napr. 06/1942)
- **umrtie** – uvádza mesiac a rok úmrtia osobnosti (rovnaký formát ako pri narodení). V prípade, že osobnosť ešte žije, bude v príslušných vstupoch uvedené N/A,
- **kategoria** – označuje jednu hlavnú kategóriu (napr. šport, hudba a pod.), do ktorej je možné danú osobu zaradiť,
- **busta** – indikuje, či je daná figurína bustou (teda, že to nie je zobrazenie celého tela, ale iba hlavy, prípadne s časťou hrudníka),
- **vaha** – označuje hmotnosť figuríny v kilogramoch (typicky má figurína váhu okolo 25kg),
- **datumDodania** – dátum, kedy bola privezená figurína do múzea vo formáte DD.MM.RRRR (napr. 06.12.2020),
- **cena** – suma v celých eurách, ktorá bola zaplatená za figurínu. V prípade zatiaľ nedodanej figuríny táto cena nereprezentuje celkovú cenu figuríny, ale zálohu, ktorá bola vyplatená.

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

Každá figurína je jednoznačne identifikovaná menom a dátumom doručenia. Jedna osoba sa môže nachádzať v múzeu viackrát (napr. ako busta, ako samostatná figurína, alebo ako súčasť súsošia – spevák

v kapele a pod.). Môžete predpokladať, že figuríny s rovnakým menom neboli dodané v rovnaký deň a tiež, že medzi objednanými figurínami nie sú dve s rovnakým menom. Figuríny s rovnakým menom budú mať vždy priradenú rovnakú kategóriu.

Ďalej vytvorte triedu `sk.upjs.finalterm.Muzeum`, ktorá bude uchovávať zoznam figurín.

Konštruktory a evidovanie figurín (3 body dokopy – povinné):

- **public** `Figurina(String meno, String narodenie, String umrtie, String kategoria, boolean busta, double vaha, String datumDodania, int cena)` – použije sa na evidovanie dodanej figuríny.
- **public** `Figurina(String meno, String narodenie, String umrtie, String kategoria, boolean busta, double vaha, int cena)` – použije sa na evidovanie objednanej figuríny.
- **public void** `eviduj(Figurina figurina)` – inštančná metóda v triede `Muzeum`, ktorá zaeviduje údaje o jednej figuríne.

Práca so súbormi (povinné):

V triede `Figurina`:

- **public static** `Figurina zoStringu(String popis)` – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy `Figurina`. Parameter je reťazec v tvare "meno \t narodenie \t umrtie \t kategoria \t busta \t vaha \t datumDodania \t cena", resp. "meno \t narodenie \t umrtie \t kategoria \t busta \t vaha \t cena" ak ešte nebola figurína dodaná (3 body); *Poznámka:* Znak `\t` je neviditeľný znak tabulátora. Scanner-u môžete povedať, že oddeľovač má byť tabulátor zavolaním jeho metódy `useDelimiter("\t")`. Namiesto hodnôt `true` a `false` môžete využiť nejaké znakové reťazce napr. "busta" a "figurina". 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 figuríne (1 bod).

V triede `Muzeum`:

- **public static** `Muzeum nacistajEvidenciu(String nazovSuboru)` – statická metóda, ktorá z uvedeného súboru prečíta evidenciu múzea (zoznam figurín), pričom v každom riadku bude popis jednej figuríny (4 body).
- **public void** `ulozEvidenciu(String nazovSuboru)` – uloží všetky zaevidované figuríny do súboru v tvare, ktorý vie spracovať metóda `nacistajEvidenciu(String nazovSuboru)` (3 body).
- **public** `String toString()` – vráti reťazec vhodne reprezentujúci kompletnú evidenciu pre dané múzeum (1 bod).

Nasledovné úlohy môžete riešiť v ľubovoľnom poradí. Niekedy je možné využiť volanie existujúcej metódy z inej úlohy. Odporúčame tiež zvážiť vytvorenie pomocnej triedy `Utils`, ktorá bude obsahovať statické metódy na prácu s dátumami.

V triede `Figurina` (úlohy môžete riešiť v ľubovoľnom poradí):

- Doplňte statickú premennú, ktorej nebude možné nastaviť inú hodnotu a bude viditeľná aj v iných triedach. Táto premenná bude uchovávať referenciu na reťazec "N/A". Vhodne použite túto premennú v príslušných metódach (1 bod).
- Vytvorte metódu **public int** `vek(String dnesnyDatum)`, ktorá vypočíta a vráti vek danej osobnosti v celých rokoch. V prípade žijúcej osobnosti berte do úvahy `dnesnyDatum` zadaný parametrom v rovnakom formáte ako dátum narodenia. V prípade nežijúcej osobnosti sa počíta dožitý vek (3 body).

- Trieda `Figurina` nech implementuje rozhranie `Comparable<Figurina>`. Implementujte príslušnú metódu takým spôsobom, aby po zotriedení boli najprv uvedené dodané figuríny podľa času dodania (tie najnovšie ako prvé) a následne objednané figuríny zoradené podľa ceny zálohy (najväčšia záloha ako prvá) (3 body).

Inštančné metódy triedy `Muzeum` (úlohy môžete riešiť v ľubovoľnom poradí):

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

- **`public int`** `pocetZijucichOsobnosti()` – vráti počet žijúcich osobností, ktoré majú v múzeu svoju figurínu (1 bod).
- **`public Map<String, Integer>`** `cenaZaKategorie()` – vráti mapu, kde bude každej kategórii pridelená celková zaplatená suma za už dodané figuríny (2 body).
- **`public void`** `dodanieFiguriny(Figurina objednávka, int doplatok, String datum)` – zaeviduje dodávku objednanej figuríny. Parameter `objednavka` referencuje objekt triedy `Figurina`, ktorý zodpovedá samotnej objednávke a má rovnaké parametre ako sa nachádzajú v príslušnom objekte v zozname figurín (nemusí ísť nutne o rovnaký objekt). Parameter `doplatok` určuje, koľko sa má doplatiť k zaplatenej zálohe a `datum` označuje deň dodania figuríny. Zabezpečte, aby sa po vykonaní tejto metódy v zozname figurín nachádzal objekt prislúchajúci dodanej figuríne buď úpravou hodnôt premenných pre príslušný objekt alebo pridaním nového objektu a odstránením existujúceho (môžete zväziť prekrytie metódy `equals` a `hashCode` v triede `Figurina`) (2 body).
- **`public int`** `umrtieOsobnosti(String meno, String datum)` – zaeviduje úmrtie osobnosti tak, že nastaví dátum úmrtia podľa hodnoty premennej `umrtie` príslušným figurínam (metóda modifikuje zoznam figurín). Dátum zadaný parametrom `datum` je vo formáte `DD.MM.RRRR`. Metóda vráti počet figurín zobrazujúcich danú osobnosť (2 body).
- **`public int`** `zalohyZaBusty()` – spočíta, aká suma bola zaplatená za objednané busty (1 bod).
- **`public double`** `vahovyMedian()` – vráti medián váh všetkých figurín, ktoré nie sú bustami. Medián je hodnota, pre ktorú platí, že polovica hodnôt je väčších alebo rovných ako medián a polovica je menších alebo rovných. Ak je počet hodnôt páry, medián sa počíta ako priemer dvoch hodnôt “v strede” (3 body).
- **`public String`** `topKategoria()` – vráti názov kategórie, v ktorej je zaradených najviac figurín. Pri rovnosti počtu figurín rozhoduje celková hodnota (súčet cien figurín v danej kategórii). Pri rovnosti cien môžete vrátiť ľubovoľnú kategóriu. (2 body za korektné riešenie + 2 body za riešenie s jedným prechodom zoznamom figurín).
- **`public String`** `kedySaMohliStretnut(String kategoria, String dnesnyDatum)` – overí, či sa všetky osobnosti zobrazené na figurínach v zadanej kategórii mohli stretnúť aj v reálnom živote a vráti mesiac, v ktorom bolo takéto stretnutie možné (môžete predpokladať, že ak sa jedna osoba narodila v nejakom mesiaci, kedy iná osoba zomrela, tak sa mohli stretnúť). V prípade, že sa všetky osobnosti stretnúť nemohli, nech metóda vráti **`null`**. Premenná `dnesnyDatum` definuje aktuálny mesiac a rok v tvare `MM.RRRR`. Túto hodnotu môže vrátiť metóda, ak sú všetky osoby z danej kategórie žijúce (5 bodov).
- **`public List<String>`** `dodaneZijuceOsobnosti()` – vráti zoznam mien teraz už zosnulých osobností, ktoré mali v múzeu aspoň jednu svoju figurínu dodanú ešte kým žili (4 body).
- **`public double`** `priemernyRatingMuzea(Map<String, Double> ratingOsobnosti, Map<String, Double> ratingKategorii)` – vypočíta a vráti priemerný rating pre všetky figuríny nachádzajúce sa v múzeu. V mape referencovanej parametrom `ratingOsobnosti` je hodnotenie pre jednotlivé osobnosti získané celosvetovo v stupnici 0-10 (10 je najlepšie hodnotenie). V prípade, že pre nejakú osobnosť nie je evidované hodnotenie, zoberie sa predvolený rating pre jej kategóriu. Dostupné

ratingy pre jednotlivé kategórie sú uvedené v mape referencovanej premennou `ratingKategorii` (2 body).

- **public boolean** `premiumMuzeum(List<String> osobnosti)` – múzeum môžeme nazvať premiovým, ak obsahuje všetky najpopulárnejšie osobnosti (z nejakého rebríčka), ktoré lákajú návštevníkov po celom svete. Metóda vráti **true**, ak múzeum má figurínu zobrazujúcu každú osobnosť zo zadaného zoznamu (3 body).
- **public List<Figurina>** `kontrolaOriginality(boolean podlaCeny)` – múzeum využíva ponuku na overenie originality figurín a odhalenie falzifikátov. Na kontrolu môže poslať maximálne 10 figurín. Metóda poskytne zoznam figurín, pričom polovicu z nich tvorí 5 najľahších figurín, ktoré sú bustami a 5 najľahších figurín, ktoré nie sú bustami. Ak je v múzeu menej figurín daného typu, môžu sa doplniť figuríny druhého typu (napr. ak sú len 3 busty, tak v zozname môže byť 7 ostatných figurín). Ak je parameter `podlaCeny=true`, tak sa namiesto najľahších figurín vyberajú najdrahšie figuríny (8 bodov).
- **public int** `kolkoRokovIbaZDotacii(int[] dotacie, int aktualnyRok)` – vypočíta koľko rokov pokrylo múzeum nákup nových figurín z dotácii na rozvoj a nemuselo to dofinancovať z vlastných zdrojov. V poli `dotacie` sú uvedené dostupné financie za jednotlivé roky, pričom na indexe 0 je hodnota pre aktuálny rok (zadaný parametrom) a nasledovné indexy zodpovedajú predošlým rokom. Hodnota na poslednom indexe prislúcha roku, kedy bola dodaná prvá figurína. Berte do úvahy iba dodané figuríny a ich celkovú cenu, a teda zálohy a objednané figuríny nie sú zahrnuté v tomto výpočte (3 body).

Výnimka a triedenie:

Vytvorte triedu `PorovnavacFigurin`, ktorá implementuje rozhranie `java.util.Comparator<Figurina>` s príslušnou metódou `compare`, aby po aplikovaní triediaceho algoritmu boli zotriedené figuríny podľa mesiaca narodenia (bez ohľadu na rok narodenia) a v rámci jedného mesiaca podľa mena (2 body).

Vytvorte nekontrolovanú výnimku `NekorektnyFormatDatumu` a použite ju na všetkých miestach, kde sa v triede `figurina` nastavujú hodnoty premenných `narodenie` a `umrtie`. (4 body).