



Záverečný test Zadanie



Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Dvokrát meraj (rozmyšľaj), 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, povinná časť musí byť ohodnotená do 150 minút od začiatku
- funkčnosť každej metódy musí byť preukázaná spustením na vami vytvorenom testovacom vstupe, nespustiteľné metódy neumožňujú získ prísušných bodov,**
- všetky inštančné premenné musia byť neverejné.



Moreplavci

Motivácia: Moreplavectvo má bohatú históriu od čias objaviteľských ciest Krištofa Kolumba až po moderné vedecké expedície. Každý prístav si vedie záznamy o svojich lodiach - o tých, ktoré už dorazili, aj o tých, ktoré sú ešte na mori.

Vašou úlohou je pripraviť podklad pre informačný systém prístavu, ktorý bude evidovať všetky lode patriace do flotily. Či už ide o tie, ktoré sa aktuálne plavia, alebo tie, ktoré sa už úspešne vrátili.

Pohľad analytika: Pri implementácii budeme potrebovať:

- triedu **Ship**, ktorá uchováva informácie o jednej lodi, resp. plavbe,
- triedu **Port**, ktorá bude uchovávať zoznam všetkých lodí evidovaných v prístave.

Zadanie: V (ľubovoľnom) balíčku vytvorte triedu **Ship** obsahujúcu dátové položky prístupné cez gettre (a podľa uváženia aj modifikovateľné cez settre):

- name** – meno lode,
- departure** – mesiac a rok odchodu lode z prístavu vo formáte MM/RRRR (napr. 09/1519),
- lastMessage** – mesiac a rok poslednej informácie o lodi (rovnaký formát ako departure). V prípade ak o lodi nie sú žiadne správy, bude tam uvedené N/A,
- mission** – označuje názov misie, ku ktorej loď patrí,
- armed** – označuje, či je loď a jej posádka ozbrojená,
- arrival** – presný dátum príchodu do prístavu vo formáte DD.MM.RRRR (napr. 08.09.1522),
- sailorCount** – počet námorníkov na lodi (v čase odchodu).

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

Jedna loď sa môže v zozname opakovať aj viackrát. Každý záznam reprezentuje jednu plavbu. Každá loď bude v zozname bez času príchodu maximálne jedenkrát. Môžete predpokladať korektný vstup. Odchod a príchod sa nepovažuje za získanie správy o lodi. Tento dátum bude iný (avšak môže mať rovnaký mesiac ako odchod alebo príchod).

Ďalej vytvorte triedu **Port**, ktorá bude uchovávať zoznam lodí.

Konštruktory a evidovanie lodí (povinné):

- **public** Ship(String name, String departure, String lastMessage, String mission, **boolean** armed, String arrival, **int** sailorCount) – použije sa na evidovanie lode po príchode
- **public** Ship(String name, String departure, String lastMessage, String mission, **boolean** armed, **int** sailorCount) – použije sa na evidovanie lode, ktorá je stále na mori
- **public void** addShip(Ship ship) – metóda v triede **Port**, ktorá zaeviduje údaje o lodi.

Práca so súborami (povinné):

V triede **Ship**:

- **public static** Ship fromString(String input) – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy **Ship**. Parameter je reťazec v tvare "name;departure;lastMessage;mission;armed;arrival;sailorCount", resp. "name;departure;lastMessage;mission;armed;sailorCount" ak loď je ešte na mori.
Poznámka: Scanner-u môžete povedať, že oddelovač má byť bodkočiarka zavolaním jeho metódy `useDelimiter(";")`.
- **public** String toString() – vráti reťazec vhodne reprezentujúci údaje o lodi.

V triede **Port**:

- **public static** Port loadShips(String fileName) – statická metóda, ktorá z uvedeného súboru prečíta informácie o prístave (zoznam lodí), pričom v každom riadku bude popis jednej lode (plavby).
- **public void** saveShips(String fileName) – uloží všetky zaevidované plavby (lode) do súboru v tvare, ktorý vie spracovať metóda loadShips.
- **public** String toString() – vráti reťazec vhodne reprezentujúci kompletný popis plavieb (lodí).

Za povinnú časť bude udelených **15 bodov**. Nasledovné úlohy môžete riešiť v ľubovoľnom poradí:

Metódy triedy **Ship** (úlohy sú zoradené podľa počtu bodov):

- [1b] 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.
- [3b] Trieda **Ship** nech implementuje rozhranie **Comparable<Ship>**. Implementujte príslušnú metódu takým spôsobom, aby po zotriedení boli najprv uvedené lode, ktoré sa už vrátili do prístavu zoradené podľa dátumu príchodu (tie najnovšie ako prvé) a následne lode, ktoré sa ešte nevrátili zoradené podľa počtu námorníkov (loď s najväčším počtom ako prvá).
- [3b] **public int** monthsAway() - vráti koľko mesiacov bola loď mimo prístavu - čas medzi odchodom a príchodom.

Metódy triedy **Port** (úlohy sú zoradené podľa počtu bodov):

- [1b] **public int** arrivedShips() - vráti počet realizovaných plavieb - koľko lodí sa už vrátilo do prístavu.
- [1b] **public int** sailorsAway() - vráti počet námorníkov, ktorí sú aktuálne na mori.
- [2b] **public** Map<String, Integer> sailorsPerMission() - vráti mapu, kde bude každej misii pridelený celkový počet námorníkov.
- [2b] **public** List<String> missionNames() - vráti zoznam názvov misií zoradený podľa abecedy, kde sa každý názov nachádza najviac jedenkrát.
- [2b] **public** Ship firstArrival() - vráti loď, ktorá prišla do prístavu ako prvá.
- [3b] **public double** armedSailorsMedian() - vráti medián počtu námorníkov pre všetky ozbrojené lode. 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 párny, medián sa počíta ako priemer dvoch hodnôt "v strede".
- [3b] **public int** multipleDepartures() - vráti počet lodí, ktoré vyrazili na more viac ako raz.
- [4b] **public boolean** safePort() - vráti true ak je prístav bezpečný. Prístav je bezpečný ak o každej lodi existuje nejaká správa. V žiadnej lodi sa nenachádza viac ako 50 námorníkov (v ozbrojenej lodi maximálne 30). Každá misia obsahuje minimálne 3 lode a žiadna ozbrojená loď nevyplávala v zimných mesiacoch (mesiace november-marec).
- [4b] **public** String topMission() - vráti názov misie, v ktorej je zaradených najviac lodí. Pri rovnosti počtu lodí rozhoduje celkový počet (súčet) námorníkov v danej misii. Riešenie, ktoré bude prechádzať zoznamom lodí viackrát získa iba 2 body.
- [4b] **public** List<String> wreckedShips(String currentDate) - vráti zoznam lodí, ktoré sa rozbili (stroskotali). Loď je stroskotaná ak ešte neprišla do prístavu a za posledný rok neprišla žiadna správa. Aktuálny dátum je v parametri currentDate v rovnakom formáte ako parameter lastMessage (MM.RRRR).
- [4b] **public double** ratioOfArmedSailors(String mission) - pre zadanú misiu vráti koľko percent námorníkov je na ozbrojených lodiach. Percento je zaokrúhlené na dve desatinné miesta. Riešenie bez zaokrúhlenia je za 2 body.
- [6b] **public int** maxShipsAtSea() - vráti maximálny počet lodí, ktoré boli naraz na mori. Ide o počet lodí na mori v rámci jedného mesiaca. Mesiac odchodu a príchodu je časom na mori (a všetky mesiace medzi nimi). Hľadáme maximálnu hodnotu počtu lodí.
- [7b] **public void** newMessages(Ship ship, String currentDate, Map<String, String> messages) - metóda eviduje novú správu od lode (parameter ship). Táto správa príde v dátum (currentDate - formát si môžete vybrať DD.MM.RRRR alebo MM.RRRR). Takáto metóda je za 2 body. Navyše je v parametri messages zoznam správ, ktoré zozbierala táto loď. Je to mapa, kde kľúčom je meno inej lode a hodnotou dátum poslednej správy, ktorá sa dostala k tejto lodi (MM.RRRR). Aktualizujte zoznam lodí (referencovaný inštančnou premennou), aby pri každej lodi bol čas poslednej správy (novšia hodnota z poslednej správy do prístavu a z poslednej správy k lodi ship). V mape môžu byť aj lode z iného prístavu.

Ďalšie úlohy:

- [2b] Vytvorte triedu ShipComparator, ktorá implementuje rozhranie java.util.Comparator<Ship> s príslušnou metódou compare, aby po aplikovaní triediaceho algoritmu boli zoradené lode podľa mesiaca odchodu (departure) bez ohľadu na rok a v rámci jedného mesiaca podľa mena lode. Alternatívne môžete použiť anonymnú triedu alebo lambda výraz.
- [4b] Vytvorte nekontrolovanú výnimku InvalidDateFormatException a použite ju na všetkých miestach v triede Ship, kde sa nastavuje hodnota premenných departure, lastMessage a arrival.