



Závèrečný test

Zadanie



Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Dvakrát meraj (rozmýšľaj), raz rež (programuj)

Pravidlá a informácie:

- 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,
- **funkčnosť každej metódy musí byť preukázaná spustením na vami vytvorenom testovacom vstupe, nespustiteľné metódy neumožňujú zisk príslušných bodov,**
- všetky inštančné premenné musia byť neverejné.

Knižnica

Motivácia: V knižnici JPAZ majú problém. Práve im vypršala licencia na drahý knižničný softvér a kvôli kríze nemajú na zaplatenie predĺženia licencie. Bez výpožičného programu nemôžu ďalej pracovať. Pomôžte im vytvorením jednoduchého programu.



Pohľad analytika: Pri implementácii budeme potrebovať:

- triedu **Kniha**, ktorá bude uchovávať údaje o jednej knihe,
- triedu **Kniznica**, ktorá bude uchovávať zoznam všetkých kníh v knižnici.

Zadanie: V balíku `sk.upjs.finalTerm` vytvorte triedu **Kniha**, ktorá obsahuje nasledovné dátové položky prístupné cez `gettre` (a podľa uváženia aj modifikovateľné cez `settre`):

- **nazovKnihy** (napríklad "Kronika rodu Spiderwickovcov")
- **menaAutorov** (napríklad "Holly Blacková, Tony DiTerlizzi" – knihy, najmä odborné, ale niekedy aj beletria, mávajú aj viac autorov; mená autorov sú oddelené čiarkami).
- **jePozicana** (napríklad `true`)
- **citatel** (napr. "Ján Javák")
- **dobaPozicania**, t.j. počet dní od výpožičky (napr. 22)

Upozornenie: Zadanie triedy **Kniha** predpisuje dátové položky prístupné cez `gettre`. Aké privátne inštančné premenné použijete na uloženie týchto dátových položiek je na vašom rozhodnutí. Špeciálne autori sa dajú uložiť „spolu“ v jednom reťazci, ale možno je výhodnejšie ich uložiť (aj) v štruktúrovanejšej podobe.

Ďalej vytvorte triedu `sk.upjs.finalTerm.Kniznica`, ktorá obsahuje zoznam všetkých kníh – požičaných aj nepožičaných.

Konštruktory a pridávanie kníh (3 bodov dokopy – povinné):

- **public** `Kniha(String nazov, String menaAutorov, Sting citatel, int dobaPozicania)` – použije sa, ak je kniha požičaná,
- **public** `Kniha(String nazov, String menaAutorov)` – použije sa, ak kniha nie je požičaná,
- **public void** `pridaj(Kniha kniha)` – inštančná metóda v triede `Kniznica`, ktorá pridá záznam o knihe do zoznamu kníh.

Práca so súbormi (povinné):

V triede `Kniha`:

- **public static** `Kniha zoStringu(String popis)` – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy `Kniha`. Parameter je `String` v tvare "nazov knihy \t meno autora 1, meno autora 2,... meno autora n \t meno citateľa \t doba pozicania", ak je kniha požičaná, alebo "nazov knihy \t meno autora 1, meno autora 2,... meno autora n", ak kniha požičaná nie je. (3 body).
Poznámka: Znak `\t` je neviditeľný znak tabulátora. Scanneru môžete povedať že oddeľovač má byť tabulátor zavolaním jeho metódy `useDelimiter("\t")`. Ak je viacero autorov, sú od seba oddelení čiarkou.
- **public** `String toString()` – vráti reťazec vhodne reprezentujúci danú knihu (1 bod).

V triede `Kniznica`:

- **public static** `Kniznica zoSuboru(File f)` – statická metóda, ktorá z uvedeného súboru prečíta zoznam kníh, pričom v každom riadku bude popis jednej knihy (4 body).
- **public void** `uloz (File subor)` – uloží všetky položky v knižnici do súboru v tvare, ktorý vie spracovať metóda `zoSuboru(File f)` (3 body).
- **public** `String toString()` – vráti reťazec vhodne reprezentujúci všetky knihy zaevidované v knižnici (1 bod).

Inštančné metódy triedy `Kniznica`:

Pozor! V nasledujúcich metódach predpokladajte, že názov knihy aj autor sa môžu opakovať. Typicky v knižnici je viac kópií tej istej knihy a autor typicky napísal viac titulov, potenciálne s rôznymi spoluautormi. Naopak budeme predpokladať, že knižnica nemá rôzne knihy s rovnakým názvom od rôznych autorov (aj keď v reálnom svete je to bežné).

- **public void** `novyDen(int n)` – zvýši všetkým požičaným knihám dobu požičania o n dní. Typicky v pondelok je $n=3$ a v ostatné pracovné dni je $n=1$. (2 body)
- **public int** `pocetNepozicanych(String nazov)` – vráti počet kníh s daným názvom, ktoré nie sú požičané žiadnym čitateľom. (2 body)
- **public void** `pozicka(String nazov, String citatel)` – nájde jednu knihu s daným názvom, ktorá nie je požičaná a zaeviduje ju ako požičanú pre daného čitateľa. Počet dní od výpožičky sa nastaví na nulu. (2 body)
- **public int** `vratenie(String nazov, String citatel)` – nájde jednu knihu s daným názvom, ktorú mal tento čitateľ požičanú. Nastaví ju ako vrátenú a vráti počet dní trvania výpožičky. (3 body)
- **public int** `najdlhsiaVypozička()` – vráti koľko dní je vypožičaná najdlhšie vypožičaná kniha v knižnici; ak takej niet, vráťte -1 . (3 body)
- **public int** `pocetTitulov()` – vráti počet kníh s rôznym názvom evidovaných v knižnici bez ohľadu na to, či sú požičané. (5 bodov)

- **public** Map<String, Integer> poctyVypozicanychKnih() – vráti mapovanie, ktoré každému čitateľovi s nenulovým počtom vypožičaných kníh priradí celkový počet ním vypožičaných kníh (5 bodov)
- **public** List<String> zoznamNazvovKnihAutora(String menoAutora) – vráti referenciu na zoznam názvov kníh, kde uvedený človek je aspoň jeden z autorov. Zabezpečte, aby sa názvy kníh neopakovali. (7 bodov)
- **public** List<String> slavniCitatelja() – vráti referenciu na zoznam mien takých čitateľov, ktorí sú zároveň autormi aspoň jednej knihy, alebo aspoň menovci autora. (8 bodov)
- **public** String najOblubenejsiaKniha() – vráti názov knihy, ktorú má požičaná najviac ľudí. Ak je takých kníh viac, vyberte takú, kde súčet dní výpožičky je najväčší. Ak je aj takých viac, vyberte náhodnú z nich. (10 bodov)
- **public void** vypisUpomienky() – vypíše mená všetkých čitateľov, ktorí majú aspoň jednu knihu požičaná dlhšie ako 30 dní. Vedľa každého čitateľa je vypísaný aj počet kníh, ktoré má vypožičané dlhšie ako 30 dní. (8 bodov)
 - bonus 10 bodov: podľa cenníka je pokuta za nevrátenie od 30 do 60 dní 2 eurá, od 60 dní do 90 dní 5 eur a nad 90 dní 10 eur. Vedľa každého čitateľa vypíšte akú veľkú pokutu má zaplatiť za všetky knihy, ktoré má požičané.

Triedenie (dokopy 9+1 bodov):

V triede `Kniha` implementujte rozhranie `java.util.Comparable<Kniha>`, ktorá porovná knihy abecedne podľa priezviska prvého autora, v prípade rovnakých priezvisk podľa mena prvého autora a v prípade rovnakých mien aj priezvisk prvých autorov podľa názvu knihy:

- **public int** compareTo(Kniha druha).

V triede `Kniznica` implementujte inštančnú metódu:

- **public void** zoradPodlaAutorov() – usporiada knihy podľa mien autorov.

Výnimky (3 body):

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