



# Záverečný 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 (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.

## Cloud-žnica by PAZ

**Motivácia:** S príchodom počítačov v 90.-tych rokoch sme mohli počuť tvrdenie, že knihy sú mŕtve. Dnes vieme, že opak je pravdou. Čítanie vymenilo formu klasickej knihy za e-book, PDF-ka a ďalšie formáty. Váš starší spolužiak Elemír, chcel na Vianoce darovať celej rodine knihy. Kvôli pobytu na internáte sleduje menej, čo čítajú jeho starí rodičia, rodičia a súrodenci. Tento rok im daroval elektronické čítačky a potom si uvedomil, že vďaka nim bude vedieť lepšie vybrať knihy na budúce Vianoce. Rodina teraz číta na mobiloch, čítačkách, počítačoch a notebookoch, preto sa rozhodol implementovať cloud-ové úložisko kníh. Vďaka moderným softvérom na zobrazovanie dokumentov (reader-om) a cloud-ovým riešeniam bude úložisko navyše uchovávať detailné informácie o každom jednom prístupe k literatúre.



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

- triedu `PosedenieSKnihou`, ktorá reprezentuje jeden prístup ku knihe v úložisku,
- triedu `ZoznamPosedeni`, ktorá bude uchovávať zoznam prístupov do úložiska.

Pozn.: V zadaní sú pojmy "prístup" a "posedenie s knihou" voľne zameniteľné podobne sú zameniteľné aj "zoznam posedení" a "zoznam prístupov".

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

- **datum** (dátum, kedy bol prístup do úložiska uskutočnený vo formáte DD.MM.RRRR, napr. "08.01.2020" – v dátume sú vždy úvodné nuly doplnené tak, aby deň aj mesiac boli dvojčiferné),
- **casZaciatku** (čas, kedy bol prístup uskutočnený vo formáte HH:MM:SS, napr. "13:05:45", v čase sú vždy úvodné nuly doplnené tak, aby sa každá časť časového reťazca skladala z 2 cifier, resp. bola dvojčiferná),
- **casKonca** (čas, kedy bol prístup ukončený v rovnakom formáte ako `casZaciatku`, navyše pre jednoduchosť predpokladáme, že `casKonca` je vždy v rovnaký deň ako `casZaciatku`, teda prístup nie je počas polnoci),
- **zariadenie** (názov zariadenia z ktorého bol prístup realizovaný, napr. "PocketBook 616"),
- **pouzivatel** (meno používateľa, napr. "mama", "Jožko")
- **nazovDokumentu** (názov súboru v úložisku napr. "hp3.pdf")
- **stranaZaciatku** (číslo strany na ktorej sa dokument otvoril na čítanie, pozn. čítačky si väčšinou pamätajú na ktorej strane používateľ skončil, preto vedia otvoriť dokument rýchlejšie lebo ho neotvárajú celý)
- **stranaKonca** (číslo strany na ktorej bol dokument otvorený naposledy)
- **nazov** (názov diela napr. "Harry Potter a rezeň z Azkabanu")
- **autor** (autor diela napr. "J. K. Rowling", ak je autorov viacej sú oddelený čiarkami)

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

Nie všetky dokumenty majú vyplnené dielo a autora vo vlastnostiach dokumentu, preto môžu chýbať. Podobne ak ide o napr. naskenované PDF-ko skener tieto vlastnosti nevyplní.

Ďalej vytvorte aj triedu `sk.upjs.finalTerm.ZoznamPosedeni`, ktorá bude uchovávať zoznam prístupov.

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

- **public** `PosedenieSKnihou(String datum, String casZaciatku, String casKonca, String zariadenie, String pouzivatel, String nazovDokumentu, int stranaZaciatku, int stranaKonca, String nazov, String autor)` – použije sa na vytvorenie záznamu o prístupe, ktorý nemá vyplnené vlastnosti `nazov` a `autor`.
- **public** `PosedenieSKnihou(String datum, String casZaciatku, String casKonca, String zariadenie, String pouzivatel, String nazovDokumentu, int stranaZaciatku, int stranaKonca)` – použije sa na vytvorenie záznamu o prístupe ku dokumentu ktorý má vyplnené vlastnosti `nazov` a `autor`.
- **public void** `pridaj(PosedenieSKnihou posedenie)` – inštančná metóda v triede `ZoznamPosedeni`, ktorá pridá záznam o jednom posedení s knihou do zoznamu posedení.

### Práca so súbormi (povinné):

V triede `PosedenieSKnihou`:

- **public static** `PosedenieSKnihou zoStringu(String popis)` – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy `PosedenieSKnihou`. Parameter je `String` v tvare `"datum \t casZaciatku \t casKonca \t zariadenie \t pouzivatel \t nazovDokumentu \t stranaZaciatku \t stranaKonca"`, resp. `" datum \t casZaciatku \t casKonca \t zariadenie \t pouzivatel \t nazovDokumentu \t stranaZaciatku \t stranaKonca \t nazov \t autor"`, ak `popis` obsahuje aj názov a autora literatúry (3 body);  
*Poznámka:* Znak `\t` je neviditeľný znak tabulátora. Scanner-u môžete povedať, že oddeľovač má byť tabulátor zavolaním jeho metódy `useDelimiter("\t")`. Medzera pre a za `\t` sú len kvôli zlepšeniu čitateľnosti zadania, v reťazci reálne nie sú.
- **public** `String toString()` – vráti reťazec vhodne reprezentujúci údaje o prístupe (1 bod).

V triede `ZoznamPosedeni`:

- **public static** `ZoznamPosedeni zoSuboru(String nazovSuboru)` – statická metóda, ktorá z uvedeného súboru prečíta zoznam prístupov, pričom v každom riadku bude popis jedného prístupu (4 body).
- **public void** `uloz(String nazovSuboru)` – uloží všetky prístupy v zozname do súboru v tvare, ktorý vie spracovať metóda `zoSuboru(String nazovSuboru)` (3 body).
- **public** `String toString()` – vráti reťazec vhodne reprezentujúci všetky prístupy (`PosedenieSKnihou`) v zozname posedení (1 bod).

### Inštančné metódy triedy `PosedenieSKnihou`:

- **public int** `vratCas()` – vráti, koľko sekúnd trvalo dané posedenie s knihou (1 bod). Hint.: čas začiatku a konca sa prevedie na sekundy od začiatku dňa a vráti sa ich rozdiel.
- **public int** `vratPocetStran()` – vráti koľko strán si používateľ prečítal, predpokladáme, že poslednú stranu neprečítal (1 bod).
- **public** `List<String> vratAutorov()` – vráti list autorov ak má súbor vyplnenú túto vlastnosť ináč vráti `null` (2 body).

## Inštančné metódy triedy ZoznamPosedeni:

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

- **public int** vratCas() – vráti koľko celých hodín si používatelia čítali (1 bod).
- **public int** najviacStran() – vráti koľko najviac strán bolo prečítaných naraz (1 bod).
- **public List<String>** vratOblubenychAutorov(String pouzivatel) – vráti všetkých autorov, od ktorých používateľ niečo čítal; každý autor sa v zozname môže vyskytovať najviac raz (2 body).
- **public double** priemernaRychlostPouzivatela(String pouzivatel) – vráti priemerný počet strán koľko používateľ prečíta za hodinu (2 body).
- **public Set<String>** naEx(Map<String, Integer> dokumentyAStrany) – vráti používateľov, ktorí prečítali aspoň 95% nejakého dokumentu naraz, parameter vyjadruje dokument a jeho počet strán (3 body).
- **public Set<String>** vratZariadeniaPouzivatela(String pouzivatel) – vráti množinu názvov zariadení, využívaných používateľom (2 body).
- **public double** percentoCasuZKratkychPosedeni(int minuty) – vráti koľko percent celkového času čítania patrí do posedení ktorých trvanie je menšie alebo rovne ako parameter minuty (3 body).
- **public Map<String, Integer>** vratCasKuKazdejKnihe(String pouzivatel) – vráti mapovanie, ktoré používateľovi ku názvu súboru priradí počet minút, ktoré strávil jeho čítaním (3 body + 1 bod ak súbor obsahuje názov diela tak namiesto názvu dokumentu sa zobrazí názov diela)
- **public int[]** vytvorCasovyHistogram(Set<String> pouzivatelialia) – vráti 24 prvkové pole, ktoré na indexe h bude obsahovať počet minút, ktoré si používatelia čítali v h-tej hodine. Zároveň uvažujeme iba používateľov, ktorí sú v množine zadanej parametrom. Ak je parameter pouzivatelialia **null**, uvažujeme všetkých používateľov (7 bodov).
- **public List<Integer>** vytvorDlzkovyHistogramPoPolhodinach() – vráti list, ktorí na indexe i bude obsahovať počet prístupov, ktorých dĺžka v minútach je z intervalu  $\langle i*30, (i+1)*30 \rangle$ . Príklad: Ak boli dĺžky prístupov v minútach 7, 29, 30, a 127 metóda vráti list 2, 1, 0, 0, 1 (7 bodov).
- **public ZoznamPosedeni** vytvorZoznamVObdobi(String odDatumu, String poDatum) - vráti referenciu na novovytvorený zoznam prístupov, ktorý bude obsahovať len prístupy realizované medzi dátumami odDatumu a poDatum (vrátane). Vzájomné poradie záznamov musí byť zachované (6 bodov).
- **public boolean** trendovaKniha(String nazovDokumentu, int pocet, int minuty) – vráti hodnotu **true** práve vtedy, ak knihu čítalo aspoň pocet používateľov a každý ju čítal aspoň minuty minút (6 bodov).
- **public Set<String>** jedinaKnihaZaRok(int rok) – vráti používateľov, ktorí v danom roku prečítali iba jedinú knihu (6 bodov + 5 bodov za implementáciu, ktorá zoznamom posedení preiteruje len raz).

## Výnimky (5 bodov)

Vytvorte nekontrolovanú výnimku `NekorektnyDatum` a vhodne ju použite aspoň v jednej metóde. Táto výnimka má byť vyhodnená vtedy, keď nejaký reťazec nie je korektným dátumovým reťazcom. Korektný dátumový reťazec je tvaru `DD.MM.RRRR`, pričom  $1 \leq DD \leq 31$ ,  $1 \leq MM \leq 12$ ,  $2000 \leq RRRR \leq 9999$ . Zvážte vytvorenie statickej metódy v triede `PosedenieSKnihou`, ktorá vyhodí výnimku, ak parametrom zadaný reťazec nie je korektný dátumový reťazec.

- `public static void overRetazec(String datum) throws NekorektnyDatum`

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

## Triedenie a komparator (dokopy 5 bodov)

Vytvorte triedu `ChronoKomparator` implementujúcu `java.util.Comparator<PosedenieSKnihou>` s metódou, ktorá porovná posedenia chronologicky podľa ich dátumu a času začiatku (2 body):

- `public int compare(PosedenieSKnihou posedenie1, PosedenieSKnihou posedenie2)`

V triede `ZoznamPosedeni` implementujte inštančnú metódu (3 body):

- `public List<PosedenieSKnihou> vratPosedeniaPodlaCasu()` – vráti posedenia usporiadané chronologicky podľa ich dátumu a času začiatku.