



## Závèrečný test Zadanie



Ústav informatiky  
Prírodovedecká fakulta  
UPJS v Košiciach

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



### Nemocnica

**Motivácia:** Zdravotníctvo je dôležitou oblasťou pri správe štátu. Mnohé vlády ju deklarujú ako jednu z priorit. Na vykonávanie uvážených rozhodnutí na úrovni štátu je potrebné mať prehľad o jednotlivých oblastiach. V tomto zadani sa budeme zaoberať nemocnicami, ich štruktúrou a personálnym obsadením. Úlohou je vytvoriť informačný systém na správu základných informácií o jednotlivých nemocničných oddeleniach.

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

- triedu `Oddelenie`, ktorá uchováva informácie o jednom nemocničnom oddelení.
- triedu `Nemocnica` ktorá bude reprezentovať nemocnicu, ako skupinu oddelení.

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

- **nazov** – pomenovanie oddelenia (napr. `Oddelenie geriatrickej chirurgie`, `Oddelenie otorinolaryngológie a chirurgie hlavy a krku`, ...)
- **lekari** – súhrnná hodnota úväzkov lekárov na danom oddelení, (napr. 2.5 - dvaja lekári na 100% úväzkov a 1 lekár na polovičný, alebo 2.0 - traja lekári na 3 dni v týždni a jeden lekár na 1 deň v týždni,  $3 \times 0.6 + 1 \times 0.2 = 2.0$ )
- **sestry** – celočíselný počet zdravotných sestier priradených k danému oddeleniu,
- **ambulancie** – počet ambulancií, kde sa poskytuje zdravotná starostlivosť, môže byť aj nula (napr. pre rádiologické oddelenie, kde sú RTG pracoviská)
- **kapacita** – počet dostupných lôžok v lôžkovej časti oddelenia, môže byť aj nula (napr. pre zubné oddelenie)
- **miesto** – napr. A10, M2, P0, G-2 kde prvý znak je označením pavilónu (resp. budovy) a nasledovné číslice označením poschodia. Pavilón je vždy označovaný jedným veľkým písmenom. Prízemie má hodnotu 0. Pre účely tohto zadania predpokladáme, že ambulancie a lôžková časť sú na rovnakom poschodí a v rovnakom pavilóne.
- **rokRekonstrukcie** – rok ukončenia (a skolaudovania) rekonštrukcie príslušného oddelenia,

*Upozornenie:* Zadané pre triedu `Oddelenie` predpisuje dátové položky prístupné cez `gettre`. Aké privátne inštančné premenné použijete na uloženie týchto dátových položiek je na vašom rozhodnutí. Odporúčame sa zamyslieť nad reprezentáciou pavilónu.

Ďalej vytvorte triedu `sk.upjs.zdravotnictvo.Nemocnica`, ktorá bude uchovávať zoznam oddelení.

### Konštruktory a evidovanie oddelení (povinné):

- **public** `Oddelenie(String nazov, double lekari, int sestry, int ambulancie, int kapacita, String miesto)` – použije sa na evidovanie oddelenia, ktoré neprešlo rekonštrukciou,
- **public** `Oddelenie(String nazov, double lekari, int sestry, int ambulancie, int kapacita, String miesto, int rokRekonstrukcie)` – použije sa na evidovanie rekonštruovaného oddelenia,
- **public void** `pridajOddelenie(Oddelenie oddelenie)` – metóda v triede `Nemocnica`, ktorá zaeviduje údaje o oddelení.

### Práca so súbormi (povinné):

V triede `Oddelenie`:

- **public static** `Oddelenie zoStringu(String popisOddelenia)` – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy `Oddelenie`. Parameter je reťazec v tvare `"nazov\tlekari\tsestry\tambulancie\tkapacita\tmiesto\trokRekonstrukcie"`, resp. `"nazov\tlekari\tsestry\tambulancie\tkapacita\tmiesto"` ak neprebehla rekonštrukcia; *Poznámka:* Znak `\t` je neviditeľný znak tabulátora. Scanner-u môžete povedať, že oddeľovač má byť tabulátor zavolaním jeho metódy `useDelimiter("\t")`.
- **public** `String toString()` – vráti reťazec vhodne reprezentujúci údaje o oddelení.

V triede `Nemocnica`:

- **public static** `Nemocnica nacistajPopisNemocnice(String nazovSuboru)` – statická metóda, ktorá z uvedeného súboru prečíta popis nemocnice (zoznam oddelení), pričom v každom riadku bude popis jedného oddelenia.
- **public void** `ulozPopisNemocnice(String nazovSuboru)` – uloží všetky zaevidované oddelenia do súboru v tvare, ktorý vie spracovať metóda `nacistajPopisNemocnice`.
- **public** `String toString()` – vráti reťazec vhodne reprezentujúci kompletný popis nemocnice.

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

### Ďalšie úlohy (úlohy sú zoradené podľa počtu bodov):

- [3b] Vytvorte nekontrolovanú výnimku `NekorektnyNazovException`, ktorá bude vyhodená v prípade ak názov oddelenia neobsahuje žiaden z kľúčových slov: {`oddelenie`, `pracovisko`, `centrum`}. Do úvahy berieme aj verzie s veľkým začiatočným písmenom. Výnimka musí byť vyhodená na všetkých miestach, kde sa zadáva nový názov oddelenia.
- [4b] Vytvorte statickú metódu (v ľubovoľnej triede), ktorá bude mať na vstupe referencie na dva objekty triedy `Oddelenie` a názov nového miesta. Táto metóda vráti referenciu na nový objekt tejto triedy, ktorý bude reprezentovať spojenie dvoch oddelení. Spojené oddelenia majú spojené úväzky lekárov, sestry, lôžka aj ambulancie. Nové spojené oddelenie je v evidencii uvedené ako zatiaľ bez rekonštrukcie a názov vznikne spojením dvoch existujúcich názvov. Ak oba názvy začínajú slovom `Oddelenie` (napr. `Oddelenie A`, `Oddelenie B`), tak nový názov bude `Spojené oddelenie A a B`. V opačnom prípade názov vznikne cez lomítko (`Pracovisko A / Oddelenie B`).

## Inštančné metódy triedy **Nemocnica** (úlohy sú zoradené podľa počtu bodov):

- [1b] **public double** celkovyUvazok() - vráti súčet úväzkov všetkých lekárov.
- [1b] **public boolean** rekonstruovanaNemocnica() - metóda vráti **true**, ak bola rekonštruovaná väčšina oddelení.
- [2b] **public int** rokySRekonstrukciou() - metóda vráti počet rokov, v ktorých prebehla aspoň jedna rekonštrukcia.
- [2b] **public List<String>** nazvyOddeleni() - metóda vráti zoznam názvov oddelení utriedený podľa abecedy.
- [3b] **public Oddelenie** najmensieOddelenie() - metóda vráti referenciu na lôžkové oddelenie s najmenším počtom lôžok. Ak je takýchto oddelení viac, vráti to, ktoré má menej sestier. Oddelenie s nulovou kapacitou nie je považované za lôžkové oddelenie.
- [3b] **public int** pocetRekonstruovanychAmbulancií(**int** rok, **char** pavilon) - vráti počet ambulancií, ktoré boli rekonštruované v zadanom roku a vo vybranom pavilóne.
- [4b] **public Map<String, String>** najvacsieOddelenia() - metóda vráti referenciu na mapu, kde kľúčom sú miesta (pavilón a poschodie) a hodnotou je názov najväčšieho oddelenia vzhľadom na kapacitu lôžkovej časti s prihliadnutím na počet ambulancií. Pre tento výpočet sa jedna ambulancia ráta ako 10 lôžok.
- [4b] **public List<Oddelenie>** plneVyuziteAmbulancie() - metóda vráti referenciu na zoznam oddelení, ktoré majú naplno využité ambulancie. V tomto výpočte je pre malé oddelenia (s kapacitou lôžok do 15) polovica úväzku lekára priradená lôžkovej časti a polovica ambulantnej starostlivosti a pre veľké oddelenia (nad 15 lôžok) sú tri štvrtiny úväzku lekára priradené lôžkovej časti a štvrtina ambulantnej starostlivosti. Ak ide o bezlôžkové oddelenie, všetky úväzky sú priradené ambulantnej starostlivosti. Plne využitá ambulancia potrebuje jeden lekársky úväzok na 100%.
- [5b] **public List<String>** zoradeneOddelenia() - metóda vráti zoznam názvov oddelení, zoradených podľa súhrnnej hodnoty úväzkov lekárov. V prípade rovnosti úväzkov je kritériom počet ambulancií. V zozname sú najprv uvedené rekonštruované oddelenia podľa týchto kritérií, nasledované oddeleniami, ktoré neprešli rekonštrukciou.
- [5b] **public int[]** poschodia() - metóda vráti pole čísel, kde hodnota zodpovedá počtu poschodí v danej budove. Pole je indexované nasledovne, index 0 zodpovedá označeniu 'A', index 1 označeniu 'B' a index 25 označeniu 'Z' (zvyšné indexy 2 až 24 podľa abecedy). Budova môže mať aj záporné hodnoty poschodia. Nie všetky poschodia sú obsadené oddeleniami. Môžete predpokladať, že najnižšie aj najvyššie poschodia budú obsahovať aspoň jedno oddelenie.
- [6b] **public int** personalneZmeny(**int** min, **int** max, **int** maxInt) - zadané parametre min a max hovoria o minimálnom a maximálnom počte lôžok na jednu sestru. Výnimkou je oddelenie intenzívnej starostlivosti, kde je maximálna hodnota iná (určená parametrom maxInt). Ak je na oddelení veľa sestier, je potrebné ich presunúť na iné oddelenie alebo prepustiť. Ak je na oddelení málo sestier, je potrebné ich presunúť z iného oddelenia alebo prijať nové. Metóda vráti koľko sestier je potrebné prijať, resp. záporné číslo označí koľko sestier je potrebné prepustiť celkovo za celú nemocnicu.
- [6b] **public List<String>** spolocneOddelenia(List<Nemocnica> nemocnice) - metóda vráti zoznam názvov oddelení, ktoré sa okrem tejto nemocnice nachádzajú vo všetkých ďalších nemocniciach (zadaných parametrom).

- [6b] **public boolean** premenovanie() - pavilóny môžu mať rôzne označenie (napr. geriatrický pavilón môže dostať označenie G) a nemocnica môže mať pavilóny napr. D, G, U. Táto metóda premenuje existujúce pavilóny abecedne (začínajúc pavilónom A, B, C ...). Pavilón s najväčším počtom ambulancií bude mať označenie A, všetky ďalšie budú označené tak, že čím viac ambulancií, tým skôr je označenie v abecede. Ak viacero pavilónov nemá ambulancie alebo majú rovnaký počet ambulancií, tak sa zachová abecedné poradie (pavilón, ktorý bol skôr v abecede pred premenovaním, bude skôr v abecede aj po premenovaní). Metóda vráti true ak sa premenoval aspoň jeden pavilón. Pozn. metóda modifikuje existujúce objekty

*Upozornenie:* Pri úlohach za 5 a viac bodov môžete požiadať o prezradenie myšlienky riešenia. Maximálny bodový zisk za danú úlohu je o 2 body menej ako je napísané pri zadaní.