



# Záverečný test

## Zadanie



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 (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.

## PAZTurtle útulok

**Motivácia:** PAZ korytnačky majú veľmi radi zvieratá a veľmi dobre vedia, že zviera nie je vec, ale aj tam sa stane, že sa niektoré ocitne na ulici a skončí v útulku. Avšak, v útulkoch pracuje dostatok zamestnancov a tí sa spolu s dobrovoľníkmi starajú o zvieratá na špičkovej úrovni. O každom zhodnotia, čo potrebuje vzhľadom na svoj vek, zdravotný stav a predispozíciu rasy, aby bolo spokojné. Potom mu to po celú dobu pobytu v útulku aj zabezpečia. Či už sú to dlhé vychádzky do prírody, špeciálna diéta, alebo výcvik. Zároveň sa snažia, aby sa každé zviera dostalo do novej rodiny, ktorá mu bude rozumieť a správne sa o neho postará. Aby to všetko bolo možné, útulky potrebujú dobrú evidenciu zvierat, kde o každom budú uvedené základné údaje a forma starostlivosti.



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

- triedu **Zviera**, ktorá reprezentuje jedno zviera
- triedu **Útulok**, ktorá bude uchovávať zoznam všetkých zvierat, ktoré prešli útulkom

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

- **menoZvierata** (meno zvierat'a, ktoré dostalo pri prijatí do útulku, napr. "Freda")
- **datumPrijatia** (dátum, kedy bolo zviera prijaté do útulku, vo formáte `DD.MM.RRRR`, napr. "24.01.2021" – v dátume sú vždy úvodné nuly doplnené tak, aby deň aj mesiac boli dvojciferné),
- **datumOdovzдания** (dátum, kedy bolo zviera odovzdané na adopciu, alebo do dočasnej opatery, v rovnakom formáte ako `datumPrijatia`; ak je zviera ešte v útulku, táto položka je `null`),
- **druhARasa** (názov druhu a rasy zvierat'a, ak je známa, vo formáte `druh:rasa`, napr. "pes:dlhosrstá kólia", "mačka:britská modrá", ak nie je rasa známa tak napríklad iba "pes:križenec"),
- **vek** (približne odhadnutý vek zvierat'a pri prijatí do útulku, v celých rokoch),
- **zodpovednyZamestnanec** (meno zamestnanca, ktorý má pridelenú starostlivosť o zviera, napr. "Jožko Mrkvička"),
- **dobrovolnici** (meno dobrovoľníka/dobrovoľníkov, ktorí pomáhajú zodpovednému zamestnancovi so starostlivosťou o zviera, napr. "Katka Milá").),
- **casZaciatkuVencenia** (čas, kedy každý deň pravidelne začína venčenie zvierat'a, vo formáte `HH:MM`, napr. "16:05", v čase sú vždy úvodné nuly doplnené tak, aby sa každá časť časového reťazca skladala z 2 cifier, resp. bola dvojciferná),

- **casKoncaVencenia** (čas, kedy každý deň pravidelne končí venčenie, v rovnakom formáte ako casZaciatku, navyše pre jednoduchosť predpokladáme, že casKonca je vždy v rovnaký deň ako casZaciatku, lebo zvieratá sa cez noc nevenčia),

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

Nie všetky zvieratá potrebujú venčenie mimo areálu útulku (napr. zajace), a preto nemajú vyplnené položky `casZaciatkuVencenia` a `casKoncaVencenia`.

Ďalej vytvorte aj triedu `sk.upjs.finalTerm.Utulok`, ktorá bude uchovávať zoznam zvierat, ktoré prešli útulkom.

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

- **public** `Zviera(String menoZvierata, String datumPrijatia, String datumOdovzdania, String druhARasa, int vek, String zodpovednyZamestnanec, String dobrovolnici, String casZaciatkuVencenia, String casKoncaVencenia)` – použije sa na vytvorenie záznamu o zvierati, ak ide o zvieratá, ktoré je potrebné venčiť mimo areálu útulku.
- **public** `Zviera(String menoZvierata, String datumPrijatia, String datumOdovzdania, String druhARasa, int vek, String zodpovednyZamestnanec, String dobrovolnici)` – použije sa na vytvorenie záznamu o zvierati, ktoré nepotrebuje venčenie mimo areálu útulku.
- **public void** `pridaj(Zviera zviera)` – inštančná metóda v triede `Utulok`, ktorá pridá záznam o jednom zvierati do zoznamu zvierat v útulku.

### Práca so súborami (povinné):

V triede `Zviera`:

- **public static** `Zviera zoStringu(String popis)` – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy `Zviera`. Parameter je `String` v tvare `"menoZvierata\t datumPrijatia \t datumOdovzdania \t druhARasa \t vek \t zodpovednyZamestnanec \t dobrovolnici \t casZaciatkuVencenia \t casKoncaVencenia"`, resp. `"menoZvierata\t datumPrijatia \t datumOdovzdania \t druhARasa \t vek \t zodpovednyZamestnanec \t dobrovolnici "`, ak ide o zvieratá, ktoré nie je potrebné venčiť mimo areálu útulku (3 body).

*Poznámka:* Znak `\t` je neviditeľný znak tabulátora. Scanner-u môžete povedať, že oddeľovač má byť tabulátor zavolaním jeho metódy `useDelimiter("\t")`. Medzera pred a za `\t` sú len kvôli zlepšeniu čitateľnosti zadania, v reťazci reálne nie sú.

*Poznámka:* Forma oddeľovania dobrovoľníkov v string-och je na vašej voľbe.

- **public** `String toString()` – vráti reťazec vhodne reprezentujúci údaje o zvierati (1 bod).

V triede `Utulok`:

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

Nasledujúce úlohy môžete riešiť v ľubovoľnom poradí. Ale riešenie jednej môže zjednodušiť nasledujúce.

### Inštančné metódy triedy `Zviera`:

- **public int** `vratCasVUtulku()` – vráti počet dní, ktoré strávilo zvieratá v útulku; ak je zvieratá ešte stále v útulku, vráti `null`. Započítava sa aj deň prijatia a deň odovzdania zvieratá, zároveň kvôli zjednodušeniu môžeme predpokladať, že každý mesiac má presne 31 dní (2 body).

- **public** List<String> vratDobrovolnikov() – vráti list dobrovoľníkov, ktorí pomáhajú so starostlivosťou o zvierá (2 body).
- **public int** aktualnyVek(String aktualnyDatum) – vráti odhadovaný aktuálny vek zvieratá v celých rokoch, na základe aktuálneho dátumu zadaného parametrom, dátumu prijatia a odhadovaného veku pri prijatí do útulku (3 body).

### Inštančné metódy triedy Utulok:

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

- **public int** vratDlzkuvencenia() – vráti koľko minút dokopy strávia zamestnanci a dobrovoľníci pri venčení zvierat v útulku v priebehu jedného dňa (1 bod). +1bod, ak vezmeme do úvahy iba zvieratá, ktoré sú naozaj v útulku, teda neboli odovzdané na adopciu, alebo do dočasnej starostlivosti.
- **public int** najdlhsievencenie() – vráti koľko minút trvalo najdlhšie venčenie (1 bod).
- **public** List<String> vratSkusenychDobrovolnikov(String rasa) – vráti všetkých dobrovoľníkov, ktorí majú skúsenosť so starostlivosťou o zadanú rasu zvieratá. Každý dobrovoľník sa môže v zozname vyskytovať najviac raz (2 body).
- **public** String zvieraSNajdlhsimPobytom() – metóda vráti meno zvieratá, ktoré strávilo v útulku najdlhší čas, od jeho prijatia, až po odovzdanie na adopciu, alebo do dočasnej starostlivosti (3 body).
- **public** Set<String> vratZvierataPodlaVekuADruhu(String vek, String druh) – vráti množinu mien zvierat, ktoré zodpovedajú veku určenému parametrom vek a druhu určenému parametrom druh (2 body). +2 body ak je vek **null** alebo prázdny reťazec, tak metóda vráti mená zvierat len podľa druhu, podobne ak je druh **null** alebo prázdny reťazec, tak berieme do úvahy iba vek.
- **public double** percentoPodlaVeku(String vek) – vráti koľko percent všetkých zvierat má aktuálny odhadovaný vek zodpovedajúci veku zadanému parametrom (3 body).
- **public** Utulok vratZoznamZvieratZaObdobie(String odDatumu, String poDatum) – metóda vráti zoznam zvierat, ktoré boli prijaté do útulku počas obdobia určeného parametrami (vrátane dní určených parametrami). (3 body).
- **public double** dotaciaNaUtulok(Map<String, Integer> dotaciaNaZviera, String odDatumu, String poDatum) – dotaciaNaZviera je mapa, kde pre každý druh zvieratá je určená dotácia od zriaďovateľa útulku na jeden deň. Môžeme predpokladať, že sa tam nachádzajú všetky rasy zvierat, o ktoré sa útulok stará. Metóda vypočíta a vráti celkovú výšku dotácie na útulok od zriaďovateľa v období odDatumu po poDatum (4 bodov). +2 body, ak pri výpočte dotácie zoberieme do úvahy aj vek zvieratá nasledovne: ak je zviera staršie ako 5 rokov, tak sa dotácia na zviera pre násobí koeficientom  $1 + 0.1 * \text{počet zvierat daného druhu v útulku vo veku viac ako 5 rokov}$ .
- **public** Utulok vratZoznamZvieratBezVencenia() – metóda vráti referenciu na novovytvorený objekt triedy Útulok, ktorý bude obsahovať iba zvieratá, ktoré nechodia na pravidelné venčenie (2 body).
- **public** Map<String, Integer> najaktivnejisiDobrovolnici() – vráti mapu, kde je dobrovoľníkovi priradené aké celkové množstvo času v minútach (po zaokrúhlení) boli zvieratá, o ktoré sa pomáha starať, na venčení počas jedného dňa. Neberieme do úvahy zvieratá, ktoré sa práve nenachádzajú v útulku (4 body).
- **public** List<String> top5najaktivnejisichDobrovolnikov() – na základe predchádzajúcej metódy vytvorte novú metódu tak, aby vrátila 5 najaktívnejších dobrovoľníkov (3 body). Pozn. Bez korektného riešenia predchádzajúcej úlohy nemôžete riešiť túto.
- **public** Map<String, String> topDvojice() – vytvorte mapu, kde je zodpovednému zamestnancovi priradený dobrovoľník, s ktorým sa spoločne starajú o najväčší počet zvierat. Ak je pre nejakého zamestnanca viac takých dobrovoľníkov, prideli sa mu niektorý z nich (4 body).
- **public** List<Zviera> mimoOtvoracejDoby(String otvorenie, String zatvorenie) – vráti zoznam zvierat, ktoré chodia na venčenie mimo otváracích hodín pre verejnosť, ktoré sú uvedené vo formáte HH:MM, napr. "08:15"(3 body). +3 body, ak ignorujeme 30 minút pred otvorením a 30 minút po zatvorení.

- **public boolean** spoluNaPrechadzke(String datum) – metóda vráti **true**, ak dve zvieratá počas daného dňa boli na venčení v tom istom čase (2 body). + 5 bodov za riešenie, ktoré vykoná iba jeden prechod zoznamom zvierat v útulku.

### **Triedenie a komparátor (dokopy 6 body):**

Vytvorte triedu PorovnavacZvieratPodlaPrijatia implementujúcu rozhranie `java.util.Comparator<Zviera>` s metódou (4 body):

- **public int** compare(Zviera zviera1, Zviera zviera2) – porovná zvieratá chronologicky podľa dátumu ich prijatia do útulku, ak boli prijaté v rovnakom dátume porovná ich lexikograficky podľa druhu a rasy.

V triede `Utulok` implementujte inštančnú metódu (2 body):

- **public void** usporiadaj() – usporiada zvieratá v útulku pomocou predchádzajúcej metódy.

### **Výnimka (4 body)**

Vytvorte nekontrolovanú výnimku `NekorektnyDatum` a vhodne ju použite aspoň v jednej metóde. Táto výnimka má byť vyhodенá vtedy, keď nejaký reťazec nie je korektným dátumovým reťazcom. Korektný dátumový reťazec je tvaru `DD.MM.RRRR`, pričom  $1 \leq DD \leq 31$ ,  $1 \leq MM \leq 12$ ,  $1000 \leq RRRR \leq 9999$ . Zvážte vytvorenie statickej metódy v triede `Zviera`, 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("\\.")`