



# Záverečný test

## Zadanie



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.

## Morské korytnačky

**Motivácia:** Morské korytnačky majú problém. Zistili, že sa im prudko znižuje svetová populácia a preto sa rozhodli osloviť korytnačky rodu `sk.upjs.jpaz2.Turtle`, známe skvelými znalosťami programovania, aby im navrhli aplikáciu, ktorá by im pomohla zistiť príčiny znižovania populácie a nájsť riešenia na záchranu. Korytnačky rodu `sk.upjs.jpaz2.Turtle` navrhli program, ktorý spracuje aktuálne známe údaje o morských korytnačkách a poskytne informácie, ktoré by im mohli pomôcť navrhnúť riešenie, ako sa dostať z tejto nelichotivej situácie. S implementáciou však oslovili vás, svojich najvernejších fanúšikov, študentov PF UPJŠ.



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

- triedu `Korytnacka`, ktorá bude uchovávať údaje o jednej korytnačke,
- triedu `ZoznamKorytnaciek`, ktorá bude uchovávať zoznam všetkých korytnaciek referenčnej vzorky použitej v prieskume.

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

- **meno** (meno korytnačky)
- **pohlavie** ('M' alebo 'F')
- **rodnaKrajina** (názov krajiny, v ktorej sa korytnačka narodila)
- **rodnaPlaz** (názov pláže, kde sa korytnačka narodila)
- **datumNarodenia** (dátum narodenia v obvyklom formáte, napr. „17.1.1905“)
- **rokPrvejZnasky** (napr. 1940) Poznámky: Korytnačky znesú každoročne stovky vajec, pričom kvôli zjednodušeniu predpokladajme, že vždy na rodnú pláž. Korytnačky si nepamätajú, koľko ich v ktorom roku zniesli. Hodnota nie je vyplnená, ak korytnačka vajcia ešte nikdy nezniesla.
- **rokPoslednejZnasky** (napr. 1988) Veľmi staré korytnačky už vajcia neznášajú. Korytnačky, ktoré znášajú aj v súčasnosti, majú vyplnený aktuálny rok. Hodnota nie je vyplnená, ak korytnačka vajcia ešte nikdy nezniesla.

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

Ďalej vytvorte aj triedu `sk.upjs.finalTerm.ZoznamKorytnaciek`, ktorá bude uchovávať nejakým spôsobom zoznam korytnaciek.

### Konštruktory a pridávanie korytnáčiek (3 body dokopy – povinné):

- **public** Korytnacka(String meno, **char** pohlavie, String rodnaKrajina, String rodnaPlaz, String datumNarodenia) – použije sa na vytvorenie záznamu o korytnačke, ktorá nikdy nezniesla vajcia.
- **public** Korytnacka(String meno, **char** pohlavie, String rodnaKrajina, String rodnaPlaz, String datumNarodenia, **int** rokPrvejZnasky, **int** rokPoslednejZnasky) – použije sa na vytvorenie záznamu o korytnačke, ktorá znáša/znášala vajcia.
- **public void** pridaj(Korytnacka korytnacka) – inštančná metóda v triede ZoznamKorytnaciek, ktorá pridá záznam o korytnačke do zoznamu korytnáčiek.

### Práca so súbormi (povinné):

V triede Korytnacka:

- **public static** Korytnacka zoStringu(String popis) – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy Korytnacka. Parameter je **String** v tvare "meno \t pohlavie \t rodnaKrajina \t rodnaPlaz \t datumNarodenia", resp. " meno \t pohlavie \t rodnaKrajina \t rodnaPlaz \t datumNarodenia \t rokPrvejZnasky \t rokPoslednejZnasky", ak korytnačka znáša/znášala vajcia (3 body);  
*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 korytnačke (1 bod).

V triede ZoznamKorytnaciek:

- **public static** ZoznamKorytnaciek zoSuboru(File f) – statická metóda, ktorá z uvedeného súboru prečíta zoznam korytnáčiek, pričom v každom riadku bude popis jednej korytnačky (4 body).
- **public void** uloz(File subor) – uloží všetky záznamy o všetkých korytnačkách v zozname do súboru v tvare, ktorý vie spracovať metóda zoSuboru(File f) (3 body).
- **public** String toString() – vráti reťazec vhodne reprezentujúci všetky korytnačky v zozname (1 bod).

### Inštančné metódy triedy ZoznamKorytnaciek:

- **public int** pocetAdolescentiek() – vráti počet samičiek, ktoré ešte neznášajú vajcia (1 bod).
- **public int** pocetZnasajucichKorytnaciek(**int** rok, String plaz) – vráti počet korytnáčiek, ktoré v danom roku na zadanej pláži zniesli vajcia, za predpokladu, že každá znáša vajcia na svoju rodnú pláž (2 body).
- **public** Map<String, Integer> poctyPlazi() – vráti počet pláží pre každú krajinu (4 body).
- **public** String najvacsiaTlacenica(**int** rok) – vráti názov pláže, kde v danom roku znieslo vajčička najviac korytnáčiek, za predpokladu, že každá znáša vajcia na svoju rodnú pláž (6 bodov).
- **public** Set<Korytnacka> mozneMamy(Korytnacka dieta) – vráti množinu korytnáčiek, ktoré zniesli vajčička na svojej rodnej pláži v rovnaký rok, ako sa na nej narodilo dané dieťa (4 body).
- **public** SortedMap<Integer,Integer> poctyVajec(**int** priemernaZnaska) – vráti pre každý rok počet vajec znesených na celom svete, ak na vstupe je priemerný počet vajec znesených jednou samičkou za rok (5 bodov).
- **public** Set<String> nezvycajneMena() – Korytnačky obvykle nie sú veľmi kreatívne pri voľbe svojho mena a často si zvolia bežné mená ako Jožko, Starec a podobne. Niektoré sú však kreatívne a vymyslia si také meno, aké nemá žiadna iná korytnačka. Táto metóda vráti práve takéto unikátne mená (4 body).
- **public** String najchladnejsiaPlaz() – vráti názov pláže, kde sa narodil najväčší podiel samčekov (pri vysokých teplotách sa rodí viac samičiek) (8 bodov).
- **public** String najbezpecnejsiaKrajina(**int** rok) – vráti tú krajinu, kde v danom roku prežilo najväčšie percento korytnáčiek voči počtu nakladených vajec, za predpokladu, že každá korytnačka zniesla v danom roku rovnaký počet vajec (8 bodov).

- **public** Set<String> ohrozenePlaze(int odRoku, int poRok) – vráti množinu pláží, kde sa od roku odRoku po rok poRok počet narodených a dodnes prežitých medziročne nikdy nezvýšil, t.j. každý rok sa oproti minulému roku počet narodených a dodnes prežitých korytnačiek znížil alebo ostal rovnaký (12 bodov). **Rada:** *Odporúčame vytvoriť si pomocnú metódu, ktorá vráti zoznam pláží, na ktorých sa v daných rokoch znášalo, a inú metódu, ktorá pre zadanú pláž overí, či je ohrozená.*
- **public** Set<String> radaStarsich() - radu starších tvoria dva najstaršie samčeky a dve samičky, ktoré najdlhšie znášali vajcia. V prípade, že je takých samičiek viac, vyberá sa podľa veku od najstaršej. Môžete predpokladať, že medzi staršími korytnačkami majú všetky iný dátum narodenia (15 bodov).

### Inštančné metódy triedy Korytnacka:

- **public int** vekKorytnacky(String datum) – vráti koľko rokov má daná korytnačka v daný dátum (napr. pre dátum “26.1.2015“ má korytnačka narodená 30.1.2000 14 rokov a korytnačka narodená 1.1.2000 15 rokov) (8 bodov).

### Triedenie a komparátor (dokopy 7 bodov):

Vytvorte metódu, ktorá zoradí korytnačky podľa veku.

Vytvorte triedu VekovyKomparator implementujúcu `java.util.Comparator<Korytnacka>` s metódou (6 bodov):

- **public int** compare(Korytnacka o1, Korytnacka o2) – porovná korytnačky podľa veku.

V triede ZoznamKorytnaciek implementujte inštančnú metódu (1 body):

- **public void** zoradPodlaVeku()– usporiada korytnačky v zozname podľa veku počnúc najmladšou korytnačkou.

### Výnimky (3 body)

Vytvorte kontrolovanú výnimku `RokZBuducnostiException`, informujúcu o tom, že daný rok ešte nenastal a vhodne ju použite aspoň v jednej metóde.