



# 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 a metódy s výnimkou metód uvedených v zadaní musia byť neverejné.

## Mikuláš

**Motivácia:** Firma JPAZ Inc. sa rozhodla zorganizovať pre deti svojich zamestnancov Mikuláša. Keďže je vhodné prispôbiť druh darčiekov veku a pohlaviu detí, musia si urobiť prehľad o deťoch svojich zamestnancov. Pomôžte firme vytvorením jednoduchého programu.

**Zadanie:** V balíku *sk.upjs.finalTerm* vytvorte triedu *Dieta*, ktorá bude definovať inštančné premenné na uloženie tých údajov o dieťati:

- Meno dieťaťa (napríklad "Janko Eclipsa")
- Pohlavie dieťaťa (chlapec alebo dievča)
- Vek dieťaťa (napríklad 10)
- Meno rodiča (napr. "Ján Javák" alebo "Anežka Eclipsová")
- Vzťah rodiča (otec alebo matka)



Ďalej vytvorte triedu *Deti*, ktorá reprezentuje zoznam detí všetkých zamestnancov. Každý zamestnanec môže mať viac detí. Zamestnancov, ktorí nemajú deti, v tomto zozname neevidujeme.

Všetky inštančné premenné a metódy triedy s výnimkou metód uvedených v zadaní musia byť neverejné.

### Konštruktory a pridávanie detí (4 body dokopy – povinné):

- **public** *Dieta*(String menoDietata, **boolean** chlapec, **int** vek, String menoRodica, **boolean** otec)
- **public void** pridaj(*Dieta* dieta) – inštančná metóda v triede *Deti*, ktorá pridá dieťa do zoznamu

### Práca so súbormi :

V triede *Dieta* (zoStringu 5 bodov, toString 1 bod):

- **public static** *Dieta* zoStringu(String popis) – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy *Dieta*. Parameter je *String* v tvare "meno dieťaťa \t pohlavie dieťaťa \t vek dieťaťa \t meno rodiča \t vzťah rodiča" Napr.:

Janko Mrkvička    chlapec    10    Anežka Mrkvičková    matka

Pozn: Znak \t je neviditeľný znak tabulátora. Scanneru môžete povedať že oddeľovač má byť tabulátor zavolaním jeho metódy *useDelimiter("\t")*.

- **public** String toString() – vráti reťazec vhodne reprezentujúci dané dieťa a jeho rodiča.

V triede `Deti` (zoSuboru 4 body, uloz 3 body, toString 1 bod):

- **public static** `Deti zoSuboru(String nazovSuboru)` – statická metóda, ktorá z uvedeného súboru prečíta zoznam detí (v každom riadku bude informácia o jednom dieťati; odporúčame formát, aký používa metóda `Dieta.toStringu`).
- **public void** `uloz (String nazovSuboru)` – uloží zoznam detí do súboru v takom tvare, že ho metóda `zoSuboru` vie naspäť prečítať.
- **public** `String toString()` – vráti reťazec vhodne reprezentujúci všetky deti v zozname.

### Inštančné metódy triedy *Deti* :

V nasledujúcich metódach predpokladajte, že každý zamestnanec a každé dieťa má iné meno.

- **public** `List<Dieta> zoznamDetiBezDuplicity()` – Môže nastať situácia, že niektoré deti sú uvedené v zozname dvakrát, pretože vo firme JPAZ Inc. pracujú obaja ich rodičia (otec aj matka). Metóda nech preto vráti zoznam všetkých detí, v ktorom je každé dieťa uvedené len raz - či už ako dieťa svojho otca alebo matky (6 bodov).
  - Túto metódu vhodne (tam, kde to je treba) použite aspoň v jednej metóde (1 bod).
- **public** `Deti zoznamDetiZamestnanca(String menoZamestnanca)` – vráti referenciu na nový zoznam detí, ktorý obsahuje všetky deti daného zamestnanca (5 bodov).
- **public** `List<String> menaZamestnancov()` – vráti referenciu na zoznam zamestnancov, ktorí majú deti. Každý zamestnanec by mal byť v zozname iba raz. (5 bodov).
  - **Bonus:** vráťte zoznam zotriedený podľa priezviska rodiča (5 bodov).
- **public** `Map<String, Integer> kategorieDeti()` – firma delí deti zamestnancov do štyroch kategórií: "dievča do 5 rokov", "dievča nad 6 rokov", "chlapec do 5 rokov" a "chlapec nad 6 rokov". Metóda nech vráti počet detí v každej kategórii vo forme mapy (5 bodov).
- **public double** `potrebnaSumaPenazi(double zaMale, double zaVelke)` – vráti, koľko peňazí si musí firma celkovo pripraviť, ak rozpočet pre dieťa do 5 rokov je uvedený v prvom parametri a pre dieťa nad 6 rokov v druhom parametri. (4 body).
- **public int** `maxPocetDeti()` – vráti, koľko detí má zamestnanec, ktorý má najviac detí (6 bodov).
- **public int** `priemernyVek()` – vráti priemerný vek detí v zozname (3 body).
- **public boolean** `viacDetiMajuMamicky()` – vráti **true**, ak všetky mamičky majú dohromady (ostro) viac detí, ako všetci oteckovia dohromady (3 body).

### Výnimky (každá použitá po 4 body):

- Vytvorte kontrolovanú výnimku `ZamestnanecNeznamyException` a použite ju v metóde `zoznamDetiZamestnanca(...)`.
- Vytvorte nekontrolovanú výnimku `ZapornaSumaZaDietaException` a použite ju v metóde `potrebnaSumaPenazi(...)`.

### Triedenie a komparátor (dokopy 5 bodov):

Vytvorte triedu *PorovnavacDeti*, ktorá implementuje rozhranie `java.util.Comparator<Dieta>` s metódou:

- **public int** `compare(Dieta d1, Dieta d2)` – porovná deti podľa veku a deti rovnakého veku podľa pohlavia.

V triede *Dieta* implementujte inštančnú metódu:

- **public void** `zoradPodlaVekuAPohlavia()` – usporiada deti podľa veku a deti rovnakého veku podľa pohlavia.