



# 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.

## (Nielen) vianočné koláčiky

**Motivácia:** Laura a Stanislava mali od útleho veku záľubu v koláčikoch. No nie v ich jedení, ale pečení. Ako čas plynul, chýr o ich cukrárskom talente presiahol ich  $\epsilon$ -okolie a každé Vianoce sa im začalo ozývať stále viac a viac ľudí s prosbou, či by im nenapiekli koláčiky. Neskôr to už neboli len Vianoce, ale aj oslavy či svadby. Tieto Vianoce už mali toľko objednávok, že sa rozhodli koláčikom venovať full-time a založili koláčikovú donáškovú službu. Ako správna firma vo veku digitalizácie vytvorili si aj program, ktorý im pomôže manažovať ich koláčikový biznis. Vďaka programu nikdy nezabudnú napiecť správny počet koláčikov objednaných na daný deň (pretože čerstvosť je pre nich kľúčová) a tiež zabezpečiť ich rozvoz.



**Pohľad analytika:** Pri implementácii aplikácie na správu objednávok koláčikov (cukrárenských produktov/výrobkov) budeme potrebovať:

- triedu `Objednavka`, ktorá reprezentuje objednávku na konkrétny cukrárenský produkt (kvôli zjednodušeniu predpokladáme, že každá objednávka je len na jeden typ produktu – ak chce niekto objednať rôzne druhy produktov, zadá sa viacero objednávok; jedna pre každý druh produktu),
- triedu `ZoznamObjednavok`, ktorá bude uchovávať zoznam objednávok.

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

- **produkt** (názov objednaného cukrárenského produktu, napr. „Cheesecake malinový 100g“),
- **pocetKusov** (počet kusov objednaného produktu, napr. 10),
- **klient** (meno alebo iná jedinečná identifikácia klienta, napr. „Ján Turtlák“),
- **datumObjednavky** (dátum, kedy bola objednávka prijatá, napr. „15.12.2017“),
- **datumDodania** (dátum, kedy má byť/bol výrobok hotový alebo dodaný, napr. „19.12.2017“),
- **poznamka** (poznámka k objednávke, napr. „Eko balenie, #facebook, menej cukru“),

*Upozornenie:* Zadanie pre triedu `Objednavka` 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.ZoznamObjednavok`, ktorá bude uchovávať zoznam objednávok.

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

- **public** Objednavka(String produkt, **int** pocetKusov, String klient, String datumObjednavky, String datumDodania) – použije sa na vytvorenie objednávky bez poznámky,
- **public** Objednavka(String produkt, **int** pocetKusov, String klient, String datumObjednavky, String datumDodania, String poznamka) – použije sa na vytvorenie objednávky s poznámkou,
- **public void** pridaj(Objednavka objednavka) – inštančná metóda v triede ZoznamObjednavok, ktorá pridá novú objednávku do zoznamu objednávok.

### Práca so súborami (povinné):

V triede Objednavka:

- **public static** Objednavka zoStringu(String popis) – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy Objednavka. Parameter je **String** v tvare "produkt \t pocetKusov \t klient \t datumObjednavky \t datumDodania", resp. "produkt \t pocetKusov \t klient \t datumObjednavky \t datumDodania \t poznamka", ak objednávka obsahuje poznámku (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 v objednávke (1 bod).

V triede ZoznamObjednavok:

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

### Statické pomocné metódy v triede PomocneMetody:

Vytvorte triedu PomocneMetody iba s privátnym konštruktorom (1b), ktorá bude poskytovať niekoľko užitočných statických metód:

- **public static** Map<String, Double> mapZoSuboru(File subor) – statická metóda, ktorá z uvedeného textového súboru prečíta zoznam párov reťazec-číslo. Každý pár reťazec-číslo je uložený v samostatnom riadku, pričom reťazec a číslo sú oddelené znakom tabulátora. Teda každý riadok má formát: "retazec \t cislo " (3 body).
- **public static int** porovnajDatумы(String datum1, String datum2) – statická metóda, ktorá chronologicky porovná dva dátumové reťazce podobne ako to robia komparátory. Teda v prípade, že datum1 je skôr ako datum2, metóda vráti záporné číslo; ak sú rovnaké, metóda vráti 0; ak datum1 je neskôr ako datum2, metóda vráti kladné číslo (3 body).

### Inštančné metódy triedy ZoznamObjednavok:

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

- **public int** vratPocetObjednavok(String datum) – vráti počet objednávok so zadaným dátumom dodania (1 bod).
- **public** String vratNajkratsiNazov() – vráti názov produktu, ktorý má najkratší názov a nachádza sa v zozname objednávok. Ak je takých produktov viac, vráti názov ktoréhokoľvek z nich. (3 body).
- **public** List<String> vratKlientov() – vráti zoznam klientov, pričom každý z klientov je v zozname uvedený len raz (3 body).

- **public** String najdiTopProdukt() - vráti názov najpredávanejšieho produktu, t.j. produktu, ktorého bolo celkovo objednaných najviac kusov. Ak je takých produktov viac, vráti názov ktoréhokoľvek z nich. (4 body).
- **public** int zistiPocetDodavok(String datum) - vráti počet dodávok, t.j. počet klientov, ktorým sa v daný deň dodali/majú dodať objednané cukrárenské produkty. Jeden klient môže mať na daný deň viacero objednávok (3 body).
- **public** double spocitajTrzbu(String datum, Map<String, Double> cennik) - vráti celkovú tržbu za objednávky so zadaným dátumom dodania, pričom ceny jednotlivých výrobkov sú určené mapovaním referencovaným parametrom cennik (4 body).
- **public** double vratVahuProduktov(String datum) - vráti celkovú váhu produktov v kilogramoch so zadaným dátumom dodania. Váhu produktu je možné zistiť z jeho názvu, keďže názov každého produktu je ukončený váhou produktu v gramoch – vid' príklady (7 bodov).
- **public** String vratNajLabuznika() - vráti meno klienta, ktorý si objednal najviac rôznych produktov. Ak je takých klientov viac, vráti meno ktoréhokoľvek z nich. (8 bodov)
- **public** double spocitajTrzbu(String odDatumu, String poDatum, Map<String, Double> cennik) - vráti celkovú tržbu za objednávky s dátumom dodania medzi dátumom odDatumu (vrátane) a dátumom poDatum (vrátane); pozor objednávky nemusia byť dodávané každý deň (7 bodov).
- **public** Map<String, Double> vratPodielProduktovNaTrzbe(Map<String, Double> cennik) - vráti mapovanie, ktoré každému menu produktu priradí percentuálny podiel produktu na celkovej tržbe. Parameter cennik referencuje mapovanie, ktoré každému produktu priradí jeho cenu. (7 bodov).

Poznámka k objednávke môže obsahovať akýkoľvek text. Môžu tu byť uvedené nejaké špeciálne požiadavky, upresnenia, či akékoľvek užitočné informácie. Laura a Stanislava zistili, že len dobré referencie nestačia, ak chcú rásť oveľa rýchlejšie. Rozhodli sa preto investovať do rôznych foriem marketingu. Aby vedeli, ako sa im jednotlivé formy marketingu vyplatia, rozhodli sa do poznámok písať značky (hashtagy), vďaka ktorým budú vedieť vyhodnotiť efektívnosť jednotlivých marketingových kampaní. Napríklad, ak klient spravil objednávku cez e-shop, na ktorý prišiel cez Google Adwords kampaň, tak si do poznámky k objednávke dajú značku #google. Ak prišiel cez facebookovú kampaň, do poznámky pridajú značku #facebook. Ak ide o telefonickú objednávku a podarilo sa zistiť, že nový klient ich našiel vďaka odporúčaniam, pridajú značku #referencia, atď. Každá značka (hashtag) začína znakom #, za ktorým nasledujú malé a veľké písmená anglickej abecedy. V jednej poznámke môže byť aj viacero značiek (hashtagov).

- **public** Set<String> najdiZnacky() - vráti množinu všetkých značiek, ktoré sa nachádzajú v poznámkach (8 bodov).

### Triedenie a komparátor (dokopy 3 body):

Vytvorte triedu DatumDodaniaKomparator implementujúcu rozhranie java.util.Comparator<Objednavka> s metódou (1 bod):

- **public** int compare(Objednavka objednavka1, Objednavka objednavka2) - porovná objednávky chronologicky podľa dátumu dodania.

V triede ZoznamObjednavok implementujte inštančnú metódu (2 body):

- **public** List<Objednavka> vratObjednavkyPodlaDatumuDodania() - vráti zoznam objednávok usporiadaných chronologicky podľa dátumu dodania.

### Výnimky (3 body)

Vytvorte nekontrolovanú výnimku NeznamaCenaProduktuException a vhodne ju použite aspoň v jednej metóde.