



# Záverečný test

## Zadanie



Ústav informatiky  
Prírodovedecká fakulta  
UPJŠ v Košiciach

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.

## Elektronické aukcie

**Motivácia:** Poznáte elektronické aukcie? Tí, čo chcú predáť stanovujú cenu za okamžitý predaj, ktorú keď niekto zaplatí, aukcia sa ďalej nekoná, ale kupujúci si tovar hneď kúpi. Potom stanovujú nižšiu - vyvolávaciu cenu, čo je najnižšia cena, za akú sú ochotní tovar predáť. Nakoniec ešte určujú čas, dokedy bude aukcia trvať. Záujemcovia potom licitujú o cenu a ten kto dodá nakoniec tú najvyššiu môže si tovar za túto cenu kúpiť. Naším cieľom bude vytvoriť systém simulujúci takéto aukcie.



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

- triedu **Aukcia**, ktorá bude uchovávať aktuálne informácie o jednej aukcii.
- triedu **Aukcie**, ktorá bude reprezentovať zoznam všetkých známych aukcií a umožní s nimi pracovať.

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

- nazovTovaru** (čo sa predáva v danej aukcii)
- predavajuci** (meno človeka predávajúceho tovar),
- cenaNaPredaj** (cena, za ktorú je možné tovar okamžite kúpiť v eurách, napr. "112.50"),
- vyvolavaciaCena** (cena, na ktorej sa začína dražiť, napr. "49.99")
- casKonca** (čas ukončenia dražby v počte milisekúnd od 1.1.1970 - pozn. aktuálny čas v počte milisekúnd od 1.1.1970 vracia metóda `System.currentTimeMillis()`),
- poslednaPonuka** (najvyššia cena, ktorú doteraz niekto navrhol)
- zaujemca** (meno toho, kto poslednú ponuku zadal)

*Upozornenie:* Zadanie triedy **Aukcia** 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.Aukcie`, ktorá bude uchovávať zoznam všetkých aukcií.

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

- **public** Aukcia(String nazovTovaru, String predavajuci, **double** cenaNaPredaj, **double** vyvolavaciaCena, **long** casKonca) – použije sa na vytvorenie záznamu o aukcii, o ktorú ešte nik neprejavil záujem,
- **public** Aukcia(String nazovTovaru, String predavajuci, **double** cenaNaPredaj, **double** vyvolavaciaCena, **long** casKonca, **double** poslednaPonuka, String zaujemca) – použije sa na vytvorenie záznamu o aukcii, o ktorú niekto už prejavil záujem,
- **public void** pridaj(Aukcia aukcia) – inštančná metóda v triede Aukcie, ktorá pridá nový záznam o aukcii.

### Práca so súbormi (povinné):

V triede Aukcia:

- **public static** Aukcia zoStringu(String popis) – statická metóda, ktorá vráti referenciu na novovytvorený objekt triedy Aukcia. Parameter je **String** v tvare "názov tovaru \t predávajúci \t cena na predaj \t vyvolávacia cena \t čas konca \t posledná ponuka \t záujemca", ak ide o aukciu, ktorá už má záujemcu, alebo v tvare "názov tovaru \t predávajúci \t cena na predaj \t vyvolávacia cena \t čas konca", ak ide o aukciu bez záujemcu (3 body);

*Poznámka:* Znak \t je neviditeľný znak tabulátora. Scanner-u môžete povedať, že oddeľovač má byť tabulátor zavolaním jeho metódy `useDelimiter("\t")`. Ak chcete nastaviť, aby Scanner korektne čítal čísla s desatinnou bodkou, spravte tak zavolaním metódy `useLocale(Locale.US)`.

- **public** String toString() – vráti reťazec vhodne reprezentujúci záznam o aukcii (1 bod).

V triede Aukcie:

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

### Inštančné metódy triedy Aukcie:

- **public double** najvyssiaPonuka(String zaujemca) – vráti, akú najvyššiu ponuku daný záujemca za nejaký tovar ponúkol (2 body).
- **public** Aukcia najvyhodnejsiaAukcia() – vráti takú aukciu, ktorá má najlepší (najväčší) pomer poslednej ponúkanej ceny a ceny na predaj. Ak aukcia ešte ponuku od nikoho nemá, použite vyvolávaciu cenu (3 body).
- **public int** pocetAktivnychAukcii(String predajca) – vráti, koľko aukcií má daný predajca ešte neukončených. Aukcia je ukončená ak nastal čas konca aukcie alebo tovar bol predaný za cenu na okamžitý predaj (4 body).
- **public** List<String> zoznamTovarov() – vráti zoznam všetkých tovarov z aktívnych aukcií, ktoré sa dajú v aukciách kúpiť (niektorí predávajúci môžu ponúkať rovnaký tovar). Každý tovar nech sa vo vrátenom zozname nachádza len raz. (4 body).
- **public boolean** prihod(String zaujemca, String tovar, String predajca, **double** ponukanaCena) – záujemca sa pokúša licitovať novú cenu, ak porazil predchádzajúceho záujemcu, nastúpi na jeho miesto. Ak už uplynul čas konca aukcie, nie je možné prihadzovať. Ak touto ponukou záujemca kúpil tovar, metóda vráti **true**, inak vráti **false**. (4 body).
- **public** List<String> nesutazivi() – vráti zoznam tých záujemcov, ktorí doposiaľ nakúpili tovar len za cenu za okamžitý predaj a v žiadnej neukončenej aukcii neponúkajú cenu nižšiu ako je cena za okamžitý predaj (8 bodov).
- **public** Map<String, Double> nakupene() – vráti pre každého záujemcu, za koľko už nakúpil tovary v ukončených aukciách (7 bodov).

- **public** String najsikovnejšiKupujúci()- vráti meno takého kupujúceho, ktorý zo všetkých už kúpených tovarov dosiahol najlepší (najmenší) pomer nákladov na kúpu týchto tovarov a celkového súčtu ich cien na okamžitý predaj (9 bodov).
- **public** List<String> smelinari()- vráti zoznam tých ľudí (bez opakovania), ktorí nejaký (aspoň jeden) tovar kúpili a potom ho ponúkli na predaj v neskoršej aukcii pod rovnakým názvom tovaru – nie však nutne s rovnakou cenou (8 bodov).
- **public void** najvytazenejšiDen()- vypíše na konzolu ten začiatok dňa, v ktorom sa malo/bude mať uskutočniť najviac ukončení aukcií. Pre jednoduchosť, za začiatok a koniec dňa (Greenwichského času) môžete považovať okamih, keď čas v milisekundách je deliteľný počtom milisekúnd za deň. (8 bodov).

### **Triedenie (dokopy 5 bodov):**

Rozšírte triedu `Aukcia` tak, aby implementovala rozhranie `java.util.Comparable<Aukcia>` a aby porovnávala aukcie podľa poslednej ponúknukej ceny. Ak na aukciu ešte nik nereagoval, použite vyvolávaciu cenu.

V triede `Aukcie` implementujte inštančnú metódu:

- **public void** zotriedPodlaCeny()- usporiada aukcie podľa aktuálne ponúknukej ceny.

### **Výnimky (3 body)**

Vytvorte nekontrolovanú výnimku `NeznamyClovekException` a vhodne ju použite (vyhod'ite) aspoň v jednej metóde. Pozor, nezamieňajte si `NeznamyClovekException` s výnimkou „bol zadaný prázdny reťazec alebo null“.