

B-OOP: Úloha č. 5

Pozorne si najprv prečítajte celé zadanie!

Vytvorte balík **sk.stuba.fei.uim.oop** a v ňom novú triedu s názvom **Task5**, ktorá obsahuje nasledujúce premenné a funkcie:

- metódu **main** s korektnou signatúrou.

V metóde **main** vytvorte pole reťazcov **String[] names = new String[9]**, ktoré inicializujte s hodnotami premennej **tree** z minulej úlohy. Vytvorte dve nové inštanície typu *Course*:

- **oop** s kapacitou **7**, učiteľom **Julius Siska** a s názvom **OOP**;
- **vsa** s kapacitou **10**, učiteľom **Igor Kossaczky** a s názvom **VSA**.

Vo for cykle postupne vytvorte 9 nových inštancií triedy *Student*:

- meno nastavte postupne podľa i-tej hodnoty **names**;
- ako id vygenerujte náhodné celé číslo v rozsahu *2000-10000*;
- zapíšte študenta do oboch kurzov.

Po for cykle zavolajte **printRegisteredStudents** metódu oboch vytvorených kurzov.

Pridajte balík **entity** a v ňom nové triedy **Student** a **Course**. Trieda **Student** nech obsahuje:

- defaultný konštruktor;
- privátnu inštančnú premennú **name** typu *String*;
- privátnu inštančnú premennú **id** typu *int*;
- príslušné (verejné) getter a setter metódy inštančných premenných;
- verejnú metódu **toString** s návratovou hodnotou *String* a bez parametrov, ktorá vráti meno a identifikátor študenta ako jeden reťazec;
- verejnú metódu **registerToCourse** s parametrom **course** typu *Course* bez návratovej hodnoty, v ktorej sa zavolá metóda **registerStudent** premennej **course**, kde ako parameter pošlite referenciu na danú inštanciu študenta. Na základe návratovej hodnoty metódy **registerStudent** do konzoly vypíšete, či sa študentovi podarilo prihlásenie do kurzu.

Trieda **Course** nech obsahuje:

- privátnu inštančnú premennú **courseName** typu *String*;
- privátnu inštančnú konštantu **duration** typu *int*;
- privátnu inštančnú premennú **teacher** typu *String*;
- privátnu inštančnú premennú **capacity** typu *int*;
- privátnu inštančnú premennú **registeredStudentsCount** typu *int*;
- príslušné (verejné) getter a setter metódy inštančných premenných. Nech premenná **registeredStudentsCount** má len getter;
- privátnu inštančnú premennú **students** typu *Student[]*, bez getter/setter metód;
- verejnú metódu **printRegisteredStudents** bez parametrov a bez návratovej hodnoty, v ktorej sa do konzoly vypíšu informácie všetkých prihlásených študentov. Využite pri výpise **toString** metódu triedy **Student**;
- verejnú metódu **toString** s návratovou hodnotou *String* a bez parametrov, ktorá vráti základné informácie o kurze (názov, trvanie, učiteľa) a zostávajúci počet voľných miest;
- verejnú metódu **registerStudent** s parametrom **student** typu *Student* s návratovou hodnotou typu *boolean*, v ktorej sa do zoznamu **students** pridá nový študent v prípade, že je ešte voľná kapacita. V takomto prípade sa vráti hodnota **true** a zvýši sa hodnota premennej **registeredStudentsCount** o 1. V prípade, že už nie je voľná kapacita, funkcia vráti hodnotu **false**.
- konštruktor s parametrami: **capacity**, **teacher**, **courseName**. Nastavte inštančné premenné s rovnakými názvami podľa vstupných parametrov. Premennú **registeredStudentsCount** inicializujte na hodnotu 0, **duration** inicializujte na hodnotu 12 a vytvorte pole študentov (premenná **students**) veľkosti **capacity**.

V balíku **sk.stuba.fei.uim.oop** vytvorte novú triedu s názvom **Task5_2**,

ktorá obsahuje nasledujúce premenné a funkcie:

- metódu **main** s korektnou signatúrou

V metóde **main** vytvorte pole **coins** typu **Coins[]** dĺžky 3, kde budú jednotlivé inštanície triedy **Coin** mať hodnoty 2, 5, 10. Vytvorte inštanciu triedy **MonetaryAmount** s hodnotou 13 a zavolajte na nej metódu **canAssemble**

s poľom **coins**. Vypíšte výstup. Vymyslite ďalšie príklady nastavení poľa **coins** a inštancie triedy **MonetaryAmount**, aby ste otestovali funkčnosť metódy **canAssemble**.

V balíku **entity** vytvorte triedy **Coin** a **MonetaryAmount**. Trieda **Coin** nech obsahuje:

- privátnu inštančnú konštantu **value** typu **int**, ktorá reprezentuje hodnotu mince v centoch;
- príslušnú getter metódu ku premennej **value**, nie však setter;
- konštruktor s parametrom **value**. Nastavte inštančnú konštantu s rovnakým názvom.

Trieda **MonetaryAmount** nech obsahuje:

- privátnu inštančnú premennú **amount** typu **int**, ktorá reprezentuje aktuálny objem finančných prostriedkov;
- konštruktor bez parametrov, ktorý inicializuje premennú **amount** na hodnotu 0;
- metódu **add** s parametrom **coin** typu **Coin** bez návratovej hodnoty. Metóda pričíta hodnotu objektu **coin** ku premennej **amount**;
- metódu **sub** s parametrom **coin** typu **Coin** bez návratovej hodnoty. Metóda odčíta hodnotu objektu **coin** ku premennej **amount**;
- príslušnú getter metódu;
- metódu **copy** bez parametrov s návratovou hodnotou typu **MonetaryAmount**, ktorá vráti kópiu inštancie, nad ktorou bola zavolaná;
- metódu **canAssemble** s parametrom **Coin[] coins** a návratovou hodnotou typu **boolean**. Táto metóda vráti **true** vtedy a len vtedy, keď sa hodnota **amount** dá získať sčítaním ľubovoľného počtu mincí s hodnotami z poľa **coins**. Použite rekurziu! Napr. ak sa v poli **coins** nachádzajú mince s hodnotami 2 a 5, je možné s nimi postaviť hodnotu **amount** 9 (použili sme jedenkrát mincu s hodnotou 5 a dvakrát mincu s hodnotou 2), ale nie hodnotu 3. Ak sa v poli **coins** nachádzajú mince s hodnotami 1, 2 a 5, je možné zostaviť sumu **amount** 3 (použili sme jedenkrát mincu s hodnotou 2 a jedenkrát mincu s hodnotou 1). Zamyslite sa, ako by bolo treba upraviť vašu implementáciu, aby ste vedeli vytlačiť aj informáciu o tom, z akých mincí (a v akých počtoch) sa dá hodnota **amount** zostaviť.

Triedu **MonetaryAmount** pokryte unit testami.

Do AIS odovzdajte zdrojové súbory (s príponou **.java**): ZIP súbor priečinku **src**.