5. feladatsor

Kezdés: márc 8, 15:24

Kvízinstrukciók

1. kérdés **0 pont**

Programozási nyelvek (BSc, 18) Java 5. feladatsor

1. feladat

Készítsen egy `IntegerMatrix` nevű osztályt a következő metódusokkal.

Egy konstruktor, mely 3 paramétert vár:

`int rowNum` (A mátrix sorainak száma)
`int colNum` (A mátrix oszlopainak száma)
`int[] linearData` (Egy, a mátrix elemeit sorfolytonosan tároló tömb)

Egy `toString()` metódus, mely egyetlen karakterláncba felsorolja a mátrix elemeit. A karakterláncban az egy sorban szereplő elemeket a `,` karakterrel válassza el; a sorokat a `;` karakterrel válassza el!

Például `linearData = {1,2,3,4,5,6}` esetén az `IntegerMatrix(2,3,linearData)` konstruktorhívás hatására a következő mátrix készül:

```
[1 2 3]
[4 5 6]
```

Ez esetben objektum `toString()` metódusa a következő sztringgel tér vissza: `"1,2,3;4,5,6"`.

2. feladat

Készítse el a `WildAnimal.java` fájlba a `WildAnimal` felsorolási típust (`enum`-ot), amelyben legyen négy felsorolási tag: majom, elefánt, zsiráf és mosómedve. Az állatok konstruktorában első paraméternek megkapják azt, hogy melyik gyümölcsöt szeretik enni, második paraméterként pedig azt, hogy mennyi lenne ideális esetben egy napi adagjuk az adott gyümölcsből.

Készítse el a `listAllAnimals()` metódust, amely egy ilyen formátumú szöveggel tér vissza:

"*A vadállat sorszáma*: *a vadállat neve* szeretne enni *a vadállat gyümölcse* egy héten."

Például, ha az elefánt megadott napi mennyisége 30 málna volt:
"2: Elefánt szeretne enni 210 málnát egy héten."

Az `enum` elemeinek bejárásához használja a `values()`, illetve a sorszám lekérdezéséhez az `ordinal()` metódust.

Készítsen saját `toString()` metódust, amely az adott `enum` elem által meghívott állatról írja ki az információkat.

Próbálja ki az elkészített felsorolási típust és a hozzátartozó metódusokat egy `Main` osztályban.

3. feladat

Írjon dokumentációs megjegyzést az 1. feladat függvényeihez, amiben leírja röviden a funkcionalitását. Tartalmazza legalább az alábbi címkéket:

- `@param`
- `@return`

Készítsen az osztályhoz is dokumentációs megjegyzést. Tartalmazza az `@author`, `@version`, `@since` tageket.

A `javadoc` program segítségével generáljon HTML dokumentációt a Java programhoz.

4. feladat

Javítsuk ki a HIBÁS programo(ka)t!

Készítsünk a `util` csomagon belül egy `IntVector` osztályt, amely egészek sorozatát ábrázolja! Legyen egy tömb adattagja, amely a sorozatot tárolja. Adjunk az osztályhoz egy konstruktort, amely egy egészekből álló tömböt vár paraméterül (ennek tartalmát másolja le). Vegyünk fel egy `add()` metódust, mely a sorozat minden eleméhez hozzáad egy paraméterül kapott egész számot! Készítsünk egy `toString()` metódust is, mely felsorolja a számokat szóközzel elválasztva (használjon `StringBuilder`-t). Például: `[1 2 3]`

util/IntVector.java:

```
package util;

public class IntVector {
    int[] numbers;

    IntVector(int[] numbers) {
        numbers = numbers;
    }

    public void add(int n) {
        for (int i = 0; i < numbers.length-1; i++)
            numbers[i] += n;
    }

    public String toString() {
        return Arrays.toString(numbers);
    }
}
```

IntVectorDemo.java:

```
class IntVectorDemo {
    public static void main(String[] args) {
        int[] ns = new int{1,2,3};
        IntVector v = new IntVector(ns);
        IntVector v2 = new IntVector(ns);

        System.out.println(new int{1,2,3});
        System.out.println(v);
        System.out.println(v2);

        System.out.println("v.add(1);");
        v.add(1);
        System.out.println(v);
        System.out.println(v2);

        System.out.println("ns[0] = 10;");
        ns[0] = 10;
        System.out.println(v);
        System.out.println(v2);
    }
}
```

1. gyakorló feladat

Készítsünk egy, a nemek ábrázolásához használt `Gender` nevű felsorolási típust! Ebben szerepeljen két érték, amelyek rendre `Gender.MALE` (férfi) és `Gender.FEMALE` (nő). Készítsünk `Person` névvel egy olyan osztályt, amelyben nyilvántartjuk a személyi adatokat! A rögzíteni kívánt adatok: a személy vezeték és keresztneve (mindkettő `String`), foglalkozása (`String`), neme (`Gender`) és születési éve (`int`).

Legyen a `Person` osztálynak egy olyan konstruktora, mely ezeket az adatokat paraméterként kapja.

Egészítsük ki a `Person` osztályt egy `toString()` metódussal, amely `String` típusú értékké alakítja az adott objektum belső állapotát! Készítsünk egy `equals()` nevű metódust a `Person` osztályhoz, amely eldönti a paraméterként megadott másik `Person` objektumról, hogy megegyezik-e az aktuális példánnyal. Vigyázzunk arra, hogy mivel referenciát adunk át paraméterként, az lehet (többnyire véletlenül) `null` érték is! Ilyenkor értelemszerűen az eredménye hamis lesz.

Tegyük az eddigi osztályokat a `person` csomagba és készítsünk hozzá egy főprogramot, amelyben létrehozunk két `Person` objektumot, megvizsgáljuk, hogy ugyanarról a két személyről van-e szó és az eredményt kiírjuk a szabványos kimenetre! A főprogram kerüljön a `main` csomagba!

2. gyakorló feladat

Készítsünk egy `basics.Matrix` osztályt (valós számokat tartalmazó kétdimenziós tömb mint mátrix segítségével), amelynek a következő műveletei vannak: M × N méretű nullmátrix konstruálása, M × N méretű mátrix konstruálása M × N méretű tömb segítségével, N × N dimenziós egységmátrix létrehozása (az eredmény mátrix legyen visszatérési érték), mátrix transzponáltjának ill. két mátrix összegének, különbségének

kiszámítása, a mátrix sztringként történő ábrázolása (`java.lang.StringBuilder`-t használjunk a szöveg előállításához).

Készítsünk főprogramot (Main.java) is, amely teszteli ezen műveleteket!

3. gyakorló feladat

Készítsen egy `TelevisionShop` felsorolási típust. A felsorolási tagok legyenek `SAMSUNG`, `LG`, `SKYWORTH`, `SONY`, `SHARP`. A konstruktorukban az első tag legyen, hogy hány db készülék van az adott márkából raktáron, a második és a harmadik az elérhető átmérők minimuma és maximuma legyen. Készítsen hozzá olyan metódusokat, amelyekkel ki tudja íratni az összes lehetséges kapható méret minimumát és maximumát típustól függetlenül (statikus) és olyat, amely adott márkára kiírja, hogy mekkora méretű tévéket lehet kapni. Készítsen statikus metódust, amellyel kiírja a rendelkezésre álló készletről minden tudhatót!

Használja a `final` kulcsszót, ahol lehet!

4. gyakorló feladat

Bővítse a 4. feladatsor 2. gyakorló feladat megoldását a következő metódusokkal:

A vektorhoz lehessen hozzáadni egy újabb elemet. Itt figyeljünk rá, hogy a mérete dinamikusan növekedjen (ha megtelt a tömb, akkor csináljunk egy segédtömböt 2x akkora mérettel, másoljuk át az elemeket és állítsuk át a számokat tároló tömb referenciáját a segédtömbére).

Legyen egy statikus `sum()` függvénye, amely vár két `IntVector` objektumot és összeadja őket, majd visszatér az eredmény referenciájával.

5. gyakorló feladat

Készítsen egy tetszőlegesen választott gyakorló feladat forráskódjához JavaDoc dokumentációs megjegyzéseket. A kommentek tartalmazzanak információkat a metódusok paramétereiről és visszatérési értékeiről. Generáljon HTML fájlt a `JavaDoc` programmal.

Feltöltés Fájl kiválasztása

Kvíz mentve ekkor: 10:48 Kvíz beadása