

Programozási nyelvek (BSc, 18) Java 3. gyakorlat



1. feladat

Készítsen egy csomagot `packagename` néven. A csomag tartalmazzon egy `Foo` osztályt, amelyben legyen legalább egy adattag és egy metódus; valamint tartalmazzon egy `Main` főprogramot, amely példányosítja a `Foo` osztályt, és meghívja a metódusát.

Készítsen egy másik csomagot `otherpackage` néven, amelyben legyen egy `Main` főprogram, amely szintén példányosítja a `packagename` csomagban található `Foo` osztályt.

2. feladat

Szervezze az előző órai `Point` osztályt és az őt bemutató főprogramot a `point2d` csomagba. A `Point` osztály ne látszódjon ki a csomagból. A `Point` osztály és a főprogram kerüljenek külön fordítási egységekbe.

3. feladat

Módosítsa az előző megoldást úgy, hogy a `Point` osztályt bemutató főprogramot átszervezi egy másik, `pointm` csomagba.

4. feladat

Készítsen `circle` néven csomagot; az előző feladatban elkészített pont osztály `circle.utils.Point` néven legyen része a `circle` csomagnak. A `circle` osztály tartalmazzon továbbá egy `circle.Circle` osztályt, amellyel `circle.utils.Point` típusú objektumban tároljuk egy kör középpontját, és `double` típusú adattagban a kör sugarát. Készüljön el egy `circle.Main` főprogram is.

5. feladat

Módosítsa az előző feladat megoldását úgy, hogy a `Point` és `Circle` osztályok adattagjai ne látszódnak ki az osztályból; az adattagok módosítását csak megfelelően megírt konstruktor vagy setter metódus végezheti.

6. feladat

Készítsen `stringutils` néven csomagot. A `stringutils.IterLetter` osztály konstruáláskor fogadjon egy `String` referenciát (kezeljük azt az esetet, ha ez `null`). Az osztálynak legyen egy `printNext()` metódusa, amellyel új sorban a képernyőre írjuk a sztring következő karakterét.

Ha a sztring összes karakterét kiírtuk a képernyőre, akkor a metódus többé ne írjon ki semmit. Az osztálynak legyen egy `restart()` metódusa, amely hatására a következő `printNext()` hívás a sztring elejét kezdi el kiírni. Az osztálynak legyen egy `hasNext()` metódusa, amely `true` értékkel tér vissza, ha van még kiírható elem.

Készítsen `Main` néven főprogramot, amely legyen névtelen csomagban. A főprogram példányosít egy `stringutils.IterLetter` osztályt, majd bemutatja annak használatát.

7. 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 vezetéke és keresztnéve (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 értéké 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!

1. gyakorló feladat

A `Kangaroo` osztály egy kengurut reprezentál. Az osztálynak két adattagja van, az egyik egy szöveges típusú, a kenguru nevének, a másik egész típusú és az életkorának az eltárolására szolgál.

Az osztálynak két konstruktora van. Az első egy szöveges típusú nevet és egy egész típusú életkort kap paraméterként és beállítja a megfelelő adattagokat. A második konstruktor egy egész típusú értéket kap és kiírja a kenguru lábainak számát. Az osztály rendelkezik egy `display()` metódussal is, egy szöveges típusú országnévet kap paraméterül, és kiírja a kenguru nevét, lakóhelyét (az országot), majd eggyel megnöveli az életkorát és az új életkort is kiírja.

2. gyakorló feladat

Bővítse a `stringutils` csomagot a `stringutils.IterWord` osztállyal; az osztály konstruáláskor fogadjon egy sztringet. Az osztály `printNext()` metódusa új sorban a képernyőre írja a sztring következő szavát. Az osztálynak szintén legyen `restart()` és `hasNext()` metódusa.

Készítsen `Main` néven főprogramot, amely bemutatja az osztály használatát.

3. gyakorló feladat

Készítsen `Book` osztályt, amellyel egy könyvtári könyvet reprezentálunk. Egy `Book`-nak legyen címe (`String`), szerzője (`String`), kiadás éve (`int`), oldalszáma (`int`) és egy logikai jellemzője, hogy kikölcsönözhető-e (`boolean`).

Írjon olyan konstruktort, amely minden jellemzőjét fogadja a könyvnek; illetve olyan konstruktort is, amely paraméterként egy másik `Book` referenciát vár (figyeljen arra az esetre, ha `null` referenciát kap).

Készítsen `toString()` metódust, amellyel sztringgé alakítja az objektum belső állapotát, valamint egy `equals()` metódust is, amellyel eldönti,

hogy a paraméterként kapott `Book` referencia ugyanazt a könyvet reprezentálja-e (két `Book` -ot akkor tekintünk egyenlőnek, ha ugyanaz a címe, szerzője, kiadás éve és oldalszáma; a kikölcsönözhetőség tehát nem lényeges).

Készítsen főprogramot, amellyel bemutatja az osztály használatát, valamint szervezze az osztályt és a főprogramot a `library` csomagba.

4. gyakorló feladat

Készítsen egy `game.utils.Vehicle` osztályt, amellyel egy MultiPlayer-es játék járművét reprezentáljuk. Egy járműnek van `modelid`-je (`int`), rendszáma (`String`), és két színállapota (`color1`, `color2` `int` típusú adatok). A rendszámhoz készítsen setter és getter metódusokat.

Készítsen `game.Player` osztályt, amellyel egy MultiPlayer-es játék felhasználóját (játékos) reprezentálunk. Egy játékosnak van neve (`String`), IP-címe (`String`), egészségi állapota (`int`) és lehet járműve (`game.utils.Vehicle`) (ha nincsen, akkor tároljunk `null` értéket).

A játékos osztályhoz készítsen `toString()` metódust, amellyel sztringgé alakítjuk egy játékos legfontosabb információit: nevét, IP-címét, egészségi állapotát, illetve járművének rendszámát (ha van).

Készítsen `game.Main` főprogramot, amelyben példányosít legalább 3 járművet, legalább 2 játékost; az egyik játékoshoz tartozzon jármű; írja ki a képernyőre a játékosok adatait.

