

Programozás 2

– 1. zárthelyi dolgozat (Java), gyakorló feladatsor –

2022. október 26., 8.00

1. feladat (1 pont)

Írjon egy `Product` nevű osztályt, amivel egy terméket tudunk reprezentálni. A `Product` osztályt egy `Product.java` állományba helyezze el. Az osztálynak egyetlen konstruktora legyen. A konstruktornak meg kell adni a termék nevét, mennyiségét és egységárát. A termék osztálynak van még egy azonosítója is, ami megegyezik az aktuális példány számával (1. termék, 2. termék, stb.). A `Product` osztálynak négy attribútuma van: `id`, `name`, `quantity` és `unitPrice`.

```
Product p1 = new Product("Tej", 3, 400);
Product p2 = new Product("Alma", 10, 600);
Product p3 = new Product("Körte", 12, 600);
```

Egy `Product` típusú példány `id` és `name` attribútumait ne lehessen módosítani!

Példányosítás után a példányokat a következőképpen akarjuk használni. A megjegyzésekben az elvárt kimenet látható:

```
System.out.println(p1);           // termék: Tej, mennyiség: 3, egységár: 400 Ft
System.out.println(p1.getId());   // 1
System.out.println(p3.getId());   // 3
System.out.println(p3.getName()); // Körte
System.out.println(p3.setQuantity(13)); // 13
System.out.println(p3.getUnitPrice()); // 600
System.out.println(p3.getTotalPrice()); // 7800 (13 * 600)
System.out.println(p1.compare(p2)); // -1
System.out.println(p2.compare(p3)); // 0
System.out.println(p3.compare(p1)); // 1
```

Magyarázat:

A `compare()` metódus két termék árát hasonlítja össze. Ha a bal oldali terméknek kisebb az egységára, akkor `-1`-et adunk vissza. Ha a két terméknek azonos az egységára, akkor `0` a visszatérési érték. Ha a bal oldali terméknek nagyobb az egységára, akkor `+1`-et kell visszaadni.

2. feladat (1 pont)

Írjon egy programot, mely parancssori argumentumként vár két számot (feltételezhetjük, hogy a felhasználó számokat adott meg paraméterként)! A program vegye a két szám között minden szám saját magára emelt hatványát, és írja ki ezek összegét (feltételezhetjük, hogy az összeg nem haladja meg a 10 milliárdot)!

$$1^1 + 2^2 + 3^3 + \dots + n^n$$

```
$ java Main 1 3
32
```

```
$ java Main 4 9
405071285
```

Ha a felhasználó nem megfelelő mennyiségű paramétert adott meg, akkor a program írjon ki egy informatív hibaüzenetet és lépünk ki egy hibakóddal!

```
$ java Main
Hiba! Adjon meg két számot paraméterben!
$ echo $?
1
```

Ha a második megadott szám nem nagyobb, mint az első, akkor a program írjon ki egy informatív hibaüzenetet és lépünk ki egy hibakóddal.

```
$ java Main 3 1
Hiba! A második megadott szám nagyobb legyen mint az első!
$ echo $?
1
```

```
$ java Main 4 4
Hiba! A második megadott szám nagyobb legyen mint az első!
```

3. feladat (1 pont)

Írjon egy programot, ami interaktív módon bekér a felhasználótól egy szöveget és minden olyan karaktert, ami nem betű, azt egy '_' jellel helyettesíti.

A szöveg átalakítását egy függvénnyel oldja meg!

Futási példák:

```
$ java Main
Adjon meg egy szöveget: Ez az 1.ZH
Az átalakított szöveg: Ez_az__ZH
```

```
$ java Main
Adjon meg egy szöveget: Remelem 5-os lesz a ZH
Az átalakított szöveg: Remelem__os_lesz_a_ZH
```

4. feladat (1 pont)

Az `dobozok.txt` file-ban dobozok méretei szerepelnek. Írjon egy programot, ami kiszámítja a dobozokhoz szükséges csomagolópapír méretét. Ehhez szükséges lesz a doboz felületének a kiszámítására, ami

$$2 * l * w + 2 * w * h + 2 * h * l$$

Például tegyük fel, hogy az állomány a következőket tartalmazza:

```
2x3x4
1x1x10
```

Egy 2x3x4-es méretű doboz felülete: $2 * 6 + 2 * 12 + 2 * 8 = 52 \text{ cm}^2$

Egy 1x1x10 méretű doboz felülete: $2 * 1 + 2 * 10 + 2 * 10 = 42 \text{ cm}^2$

Ebben az esetben 94 cm^2 csomagolópapírra van szükség.

Futási példa egy kisebb méretű input fájl esetén:

```
$ cat pelda.txt
2x3x4
1x1x10
```

```
$ java Main
94 cm^2
```

Figyelem! A programnak a `dobozok.txt` file-t kell feldolgoznia!