

Programozás 2

1. Zárthelyi dolgozat

2021. október 12.

1. (1 pont) Írjon egy Nevek nevű osztályt, mellyel egy neveket tartalmazó adatszerkezetet valósít meg. A Nevek osztályt egy Nevek.java nevű állományban helyezze el!

Példányosítás után az alábbiaként akarjuk használni az osztályt. A megjegyzésekben az elvárt kimenet várható.

```
System.out.println(nevek1);           // < [] <
System.out.println(nevek1.size());     // 0
System.out.println(nevek1.isEmpty());  // true
nevek1.append(„Péter”);
nevek1.append(„József”);
nevek1.append(„Ajtony”);
System.out.println(nevek1);           // < [Péter, József, Ajtony] <
System.out.println(nevek1.size());     // 3
System.out.println(nevek1.isEmpty());  // false
int count = nevek1.countHasLetterJ();
System.out.println(count);             // 2
System.out.println(nevek1);           // < [Péter, József, Ajtony] <
System.out.println(nevek1.size());     // 3
nevek1.appendAll(List.of(„Károly”, „Ferenc”)) // megjegyzés: List.of() itt egy 2 elemű
                                           // listát készít
System.out.println(nevek1);           // < [Péter, József, Ajtony, Károly, Ferenc] <
```

2. (1 pont) Írjon egy programot, ami interaktívan bekér a felhasználótól egy szót, majd keresse meg benne az első „@” karaktert, és az előtte lévő szó részt a program tegye a szó végére. Tájékoztatásul írja ki a sztring hosszát is.

A szó megváltoztatását egy `StringUtils.switchAtKukac()` nevű metódussal valósítsa meg! A metódus kap egy sztringet, és a szó első „@” karaktere előtti részt tegye át a szó végére. A metódus adja vissza a módosított sztringet. Példák:

Szó: Pe@rec
A sztring hossza: 6
Új sztring: @recPe

Szó: @Alma
A sztring hossza: 5
Új sztring: @Alma

Szó: Eper@
A sztring hossza: 5
Új sztring: @Eper

Szó: Utazás
A sztring hossza: 0
Új sztring:

Az utolsó példánál egy üres sztringet kaptunk vissza.

3. (1 pont) Írjon egy programot, ami parancssori argumentumként megkap három egész számot. Az első kettő közül a kisebbik egy intervallum alsó, a nagyobb egy felső korlátját határozza meg, harmadik egy lépésközt. A program írja ki az intervallumot meghatározó két szám közötti értékeket a lépésközzel úgy, hogy az alsó korlát része az eredménynek, a felső nem. A kimenetet formázottan jelenítsük meg. Példa:

Argumentumok: 10 3 1

Kimenet: 1. elem: 3, 2. elem: 4, 3. elem: 5, 4. elem: 6, 5. elem: 7, 6. elem: 8, 7. elem: 9

Terminálos futtatás esetén:

\$ java Main 10 3 1

Argumentumok: 3

Kimenet: Hibás paraméterezés!

A program pontosan három darab paramétert vár. Hibás paraméterezés esetén írjunk ki egy hibaüzenetet és a program lépjen ki 1-es hibakóddal. Helyes paraméterezés esetén feltételezhetjük, hogy három egész szám lett megadva.

4. (1 pont) A tablázat.txt fájlban egy táblázat adatai szerepelnek, minden sorban néhány egész számmal. A számok tabulátor jellel vannak egymástól elválasztva. A táblázat minden sorában határozza meg a legnagyobb és legkisebb szám szorzatát, majd vegyük a szorzatok összegét.

Például legyen adott a következő táblázat:

3	2	1	4
6	4	3	
2	4	8	

Ekkor az első sorban a szorzat $1 * 4 = 4$ lesz. A soronkénti szorzatok összege pedig: $4 + 18 + 16 = 38$.
Írassuk ki az input táblázatból kapott végeredményt!