

Név: .....

Neptun-kód: .....

## Programozás 2

### – 1. zárthelyi dolgozat –

2021. okt. 13., 14.00 órás csoport

## Feladatok

1. (1 pont) Írjon egy `Processor` nevű osztályt, mellyel egy processzort (CPU-t) tudunk reprezentálni. A `Processor` osztályt egy `Processor.java` nevű állományban helyezzük el!

Az osztálynak egyetlen konstruktora legyen, amiben a processzor pontos típusát kell megadni sztringként. Az egyszerűség kedvéért tegyük fel, hogy csak a mellékelt `info.txt` állományban található processzor típusok jöhetnek számításba.

Példák:

```
Processor p1 = new Processor("Intel Core i5-2500 @ 3.30GHz");
Processor p2 = new Processor("Intel Core i5-2500K @ 3.30GHz");
Processor p3 = new Processor("Intel Core i5-2310 @ 2.90GHz");
Processor p4 = new Processor("AMD Phenom II X6 1065T");
Processor p5 = new Processor("AMD Phenom II X6 1075T");
Processor p6 = new Processor("AMD Phenom II X6 1090T");
```

Egy `Processor` típusú objektum legyen *immutable*. Vagyis miután létrehoztunk egy ilyen objektumot, azt utólag ne lehessen módosítani!

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

```
System.out.println(p1);           // Intel Core i5-2500 @ 3.30GHz
System.out.println(p4);           // AMD Phenom II X6 1065T
System.out.println(p1.isIntel()); // true
System.out.println(p6.isIntel()); // false
System.out.println(p3.isAmd());   // false
System.out.println(p5.isAmd());   // true
System.out.println(p1.getClockSpeed()); // 3.30
                                   // (3.3 is elfogadható)
                                   // az órajel ennyi GHz
System.out.println(p4.getClockSpeed()); // -1.0
                                   // AMD esetén nem tudjuk...
                                   // ...a pontos értéket

System.out.println(p3.cmpClockSpeed(p1)); // -1
System.out.println(p1.cmpClockSpeed(p2)); // 0
System.out.println(p1.cmpClockSpeed(p3)); // 1
System.out.println(p3.cmpClockSpeed(p4)); // -2
```

*Folytatás a köv. oldalon!*

### **Egy kis magyarázat**

A `cmpClockSpeed()` két processzor órajelét hasonlítja össze. Mivel csak az Intel processzorok órajelét ismerjük, ezért csakis két Intel processzor összehasonlításának van értelme.

Ha a két processzor közül bármelyik is AMD gyártmányú, akkor a függvény adjon vissza  $-2$  értéket (ezzel jelezzük a hibát, vagyis hogy a két processzor nem hasonlítható össze).

Ha két Intel gyártmányú processzort hasonlítunk össze, akkor vagy  $-1$ , vagy  $0$ , vagy  $+1$  értéket kell visszaadni. Ha a bal oldali processzor kisebb órajelű, akkor  $-1$ -et adunk vissza. Ha a két processzor azonos órajelű, akkor  $0$  a visszatérési érték. Ha a bal oldali processzor a nagyobb órajelű, akkor  $+1$ -et kell visszaadni.

2. (1 pont) Írjon egy programot, ami parancssori argumentumként kap egy szót, s kiszámítja a szó hash értékét. Hibás paraméterezés esetén írjunk ki egy hibaüzenetet és a program lépjen ki 1-es hibakóddal.

Példák:

```
$ java Main
Hiba! Adj meg pontosan egy szót!
$ echo $?
1
```

```
$ java Main Alma Dio
Hiba! Adj meg pontosan egy szót!
```

```
$ java Main Alma
69
```

```
$ java Main alma
101
```

```
$ java Main Dio
96
```

```
$ java Main Programozas_2
18
```

Egy szó (sztring) hash értékét állapítsuk meg a következőképpen: vegyük az egyes betűk ASCII-kódját<sup>1</sup> s ezt szorozzuk meg az adott betű pozíciójával (az első betű pozíciója 1, a másodiké 2, stb.). Vegyük az így kapott értékek szummáját, majd ezt az összeget modulózzuk a 103 konstans értékkel. Vagyis a hash művelet eredménye a  $[0, 1, \dots, 102]$  zárt intervallumból fog kikerülni.

Vegyük például az „Alma” szót. Az „A” ASCII-kódja: 65; az „l” ASCII-kódja: 108, stb. Az „Alma” szó hash értéke:

$$(65 * 1 + 108 * 2 + 109 * 3 + 97 * 4) \bmod 103 = 996 \bmod 103 = 69$$

A hash érték kiszámítását szervezze ki egy statikus metódusba egy Hash nevű osztályon belül. A metódust a következőképpen akarjuk használni:

```
int result = Hash.stringHash("Alma");    // 69
```

Gondoskodjon róla, hogy a Hash osztályt ne lehessen példányosítani!

A Hash osztályon belül a 103-as értéket rendelje hozzá egy nevesített konstanshoz! Ez a konstans a Hash osztályon kívülről ne látszódjon!

---

<sup>1</sup>Lehet használni a <https://asciichart.com> oldalt.

3. (1 pont) Írjon egy programot, ami interaktív módon bekér a felhasználótól egy tetszőleges szöveget.

A szövegben szerepelhetnek a magyar ékezetes betűk. Ezek akár kis- és nagybetűsen is előfordulhatnak.

A program írja ki a képernyőre az input szöveget úgy, hogy abban nem szerepel egyetlen magánhangzó sem!

Példa:

```
$ java Main
Input: Kiss István
Output: Kss stvn
```

```
$ java Main
Input: Árvíztűrő tükörfúrógép
Output: rvztr tkrfrgp
```

```
$ java Main
Input: Bíró Ica
Output: Br c
```

Az ékezetek eltávolítását szervezze ki egy statikus metódusba, amit a következőképpen akarunk meghívni (példa):

```
String clean = StringUtils.removeVowels("példa");    // pld
```

Tegyen róla, hogy a StringUtils osztályt még véletlenül se lehessen példányosítani!

4. (1 pont) Tekintsük a `people.csv` nevű szöveges állományt, ami személyek adatait tartalmazza. Minden egyes sorban a következő adatok szerepelnek:

```
address;blood_group;company;job;mail;name;residence;sex;ssn;username
```

Mint látható, az egyes adatok pontosvesszővel vannak elválasztva. Minket most csak a „name” oszlop érdekel, s innen is csak azon nevekkel akarunk dolgozni, amelyek pontosan két részből állnak (*first name*, ill. *last name*). (Az angolban a keresztnév [*first name*] megelőzi a vezetéknévet [*last name*]). Ha valakinek a neve kettőnél több részből áll (pl. „Dr. Leslie Gordon”), akkor az illetőt figyelmen kívül hagyjuk.

**Feladat:** írjunk egy olyan programot, ami kilistázza azokat a neveket, ahol a keresztnév hosszabb, mint a vezetéknév.

A program tartalmazzon hibakezelést is. Ha túl kevés, vagy ha túl sok parancssori argumentumot adunk meg, akkor legyen egy hibaüzenet. Ekkor a program 1-es hibakóddal lépjen ki.

Ha pontosan egy argumentum lett megadva, akkor feltételezhetjük, hogy az egy létező file neve.

Példa:

```
$ java Main
Hiba! Egyetlen argumentum kell, a bemeneti állomány neve!
$ echo $?
1

$ java Main people.csv na.txt
Hiba! Egyetlen argumentum kell, a bemeneti állomány neve!

$ java Main people.csv
...
Keith Cox
...
```

Keith Cox nevében a keresztnév („Keith”) hosszabb, mint a vezetéknév („Cox”).