Programozás 2

– 2. zárthelyi dolgozat –

2021. november 24., 8.00

1. feladat – filmek (1 pont)

Tekintsük a `movies.csv` állományt, melynek oszlopai a következők:

`Film,Genre,Lead Studio,Audience score %,Profitability,Rotten Tomatoes %,Worldwide Gross,Year`

Készítsen egy `Movie` nevű osztályt, mellyel egy filmet tud reprezentálni. Olvassa be a `movies.csv` tartalmát, s minden egyes sort alakítson át `Movie` típusú objektummá. A `Movie` objektumokat tárolja el egy listában.

Vegyük észre, hogy az input file első sora az oszlopok nevét tartalmazza. Az input file-t tilos módosítani!

Egy `Movie` objektumban elegendő csak a következő adatokat letárolni: a film címe (Film), a film műfaja (Genre), ill. a megjelenés éve (Year). Miután elkészült egy `Movie` objektum, azt utólag ne lehessen módosítani!

Írjon egy programot, aminek parancssori argumentumként meg kell adni egy műfajt. A műfaj megadásakor nem számít a kis- és nagybetű (nincs köztük különbség). A program írja ki a képernyőre az ebbe a műfajba eső filmek címét ábécésorrendben. Nem létező műfaj esetén ne legyen semmilyen kimenet sem.

Futási példák:

```
$ java Main
Hibás paraméterezés! Adj meg egy műfajt!

$ java Main one two
Hibás paraméterezés! Adj meg egy műfajt!

$ java Main sci-fi
$

$ java Main comedy
A
B
C
...

$ java Main CoMedY
A
B
C
...
```

A `sci-fi` esetén a `$` azt jelenti, hogy visszakaptuk a promptot, nem volt semmilyen kimenet sem. A `comedy` esetén nem a valódi kimenet lett feltüntetve.

2. feladat – szótár (1 pont)

Írjon egy programot, ami parancssori argumentumként egyetlen argumentumot, a bemeneti fájl nevét várja.

A program elemezze a file tartalmát, majd írjon ki egy statisztikát. Példa:

```
$ java Main
Hiba: add meg egy file nevét!

$ java Main aa bb
Hiba: add meg egy file nevét!

$ java Main one.txt
Nagybetűk száma: 12
Kisbetűk száma: 20
Szóközök száma: 5
```

Írjon egy `Analyse.process()` nevű statikus metódust, ami bemeneti paraméterként kap egy sztringet, a visszatérési értéke pedig legyen egy szótár. A szótárban három kulcs legyen: `uppercase`, `lowercase`, ill. `space`. A kulcshoz tartozó érték azt mutassa meg, hogy az adott karakterosztályba hány karakter tartozik a bemeneti sztringben. Például a "Prog. 2" sztringre felépített szótár `lowercase` kulcsához a 3-as érték fog tartozni.

Az `Analyse` osztályt tegye ki egy `Analyse.java` nevű forrásállományba!

A feladat megoldása során természetesen használja az `Analyse.process()` metódust!

3. feladat – teszt (1 pont)

A `feladat3/` mappába másolja át az előző feladatban elkészített `Analyse` nevű osztályt. Írja meg a `TestAnalyse.java` nevű forrást, melyben alaposan leteszteli az `Analyse.process()` metódust.

Néhány tipp:

- Mi történik, ha a bemenet üres sztring?
- Mi történik, ha a bemenet csupa nagybetűs?
- Mi történik, ha a bemenet csupa kisbetűs?
- Mi történik, ha a bemenet csak szóközöket tartalmaz?
- Mi történik, ha a bemenet nem üres sztring, de nem tartalmaz sem nagybetűt, sem kisbetűt, sem szóközt?
- Mi történik, ha a bemenet vegyesen tartalmaz nagybetűket, kisbetűket, szóközöket?
- Tényleg háromelemű szótárat kapunk vissza?
- A háromelemű szótár kulcsai jól vannak elnevezve?

4. feladat – fájlok (1 pont)

Adott két, ugyanannyi sorból álló állomány (`in1.txt` és `in2.txt`), melyek mérések adatait tartalmazzák. Az első fájl (`in1.txt`) az első méréssorozat eredményeit tartalmazza. Ugyanezeket a méréseket azonos sorrendben megismételtük, s ezt a második fájlban (`in2.txt`) rögzítettük. Vagyis a fájlok `i.` sorában ugyanazon kísérletre kapott két eredmény lett rögzítve.

Egy kísérletet akkor tekintünk sikeresnek, ha a két alkalommal kapott értékek összege *meghaladja* az `1.0` értéket.

Példa:

```
$ cat in1.txt
0.1
0.5
0.9
0.3
0.6

$ cat in2.txt
0.2
0.6
0.1
0.9
0.2
```

Az első kísérlet sikertelen, mivel $0.1 + 0.2 < 1.0$. A második és a negyedik kísérlet viszont sikeres lett.

Írjunk programot, ami beolvassa a két input fájlt, s egy `out.txt` nevű állományba kiírja a sikeres kísérletek sorszámát. A program adjon egy kis visszajelzést is.

A fenti példa esetén:

```
$ java Main
-> out.txt létrehozva
-> sikeres kísérletek száma: 2

$ cat out.txt
2
4
```

Ennek a jelentése: a 2. és a 4. kísérlet sikeres volt.