

Tartalomjegyzék

- 1 Bemelegítő feladatok
 - ◆ 1.1 ArrayList
 - ◆ 1.2 HashSet
 - ◆ 1.3 HashMap
- 2 Hasznos feladat

Bemelegítő feladatok

Ezeket mind lehet csak egy **main**-be írni, egy amúgy üres osztályba. Ezt ne felejtsetek az elejére írni minden file-nak:

```
import java.util.*;
```

ArrayList

- Készítsetek egy **String**eket tároló **ArrayList**-et. Adjátok hozzá a következő **String**eket:

```
Java
Unix
Oracle
C++
Perl
```

- Majd iterátorral járjátok be a listát és írjátok ki az elemeit külön sorokba (tehát a kimenet legyen az ami pont a fenti felsorolás).
- Töröljétek a C++ és Oracle elemeket. Majd írjátok ki még egyszer a maradék listát.
- Kérdezzétek le, hogy hanyadik indexen található a Unix. Állítsátok ezt át **Linux**-ra. Végül írjátok ki csak ezt a módosított elemet a **get** metódust használva.

HashSet

- **HashSet** segítségével határozzatok meg prímeket.
- Készítsetek egy **int**eket tároló **HashSet**-et. Adjátok hozzá a számokat 2-től 100-ig. Valamint készítsetek még egy **HashSet**-et, amibe nem töltünk még elemeket.
- Kérjete el egy iterátort és iteráljatok végig az elemeken. A cikluson belül kérjete el egy újabb iterátort. Majd az így megírt belső cikluson belül vizsgáljátok, hogy a külsőben figyelt elem osztható-e a belső elemmel, ha igen és nem egyezik meg ezzel az elemmel, akkor rakjátok a második **HashSet**-be.
- Amikor lefutott ez az egymásba ágyazott két ciklus, akkor a második **HashSet**-ben megkaptuk az összetett számokat. Töröljétek ezeket az első **HashSet**-ből. (Iteráljatok végig a 2.-on, és hívjátok meg a **remove** metódust az adott elemmel az első **HashSet**-en.)

Gyakorlat4

- Gondolkozzatok el rajta hogyan lehetett volna ezt sokkal optimálisabban megoldani.

HashMap

- Készítsetek egy **HashMap**et, ami Stringekhez számokat tud hozzárendelni. Töltsétek ezt fel a következő kulcs-érték párokkal:

```
one: 1
two: 2
three: 3
four: 4
five: 5
```

- Írjátok ki csak a kulcsokat, majd csak az értékeket. Végül a kulcs-érték párokat úgy ahogy fent is látszik. Ehhez segítség:

```
Set<String> keys = hm.keySet();
for (String key: keys) {
    System.out.println("Value of "+key+" is: "+hm.get(key));
}
```

- hm itt egy **HashMap**.
- Az első sorban lekérjük a kulcsait tartalmazó halmazt.
- Majd egy ciklussal végigiterálunk a kulcsokon. (Ez az a for ciklus amit első előadáson mutattam, C és C++ban nincs ilyen.)
- A cikluson belül megy a kiírás.

Hasznos feladat

- Írjátok át a felhasználórendszert, hogy tömbök helyett **ArrayList**eket használjon.
- Írjátok át a **PersonalInfo** osztály, hogy Stringhez Stringet rendelő **HashMap**ben tárolja az adatokat.

```
HashMap<String, String> hm = new HashMap<String, String>();
```

- Legyen egy statikus **HashSet** adattagja a **PersonalInfo** osztálynak, amiben tárolva vannak a lehetséges tulajdonságok (a **HashMap** lehetséges kulcsai).
- Úgy működjön az utóbb említett **HashSet**, hogy amikor új tulajdonságot adunk hozzá egy emberhez, akkor automatikusan rakja bele a **HashSet**be a tulajdonságot, ha még nem tartalmazta eddig.
- Ez azt is jelenti, hogy mostantól a tulajdonság hozzáadása a **PersonalInfo**hoz (és ezáltal egy **User**hez) egy metódussal működjön amiben csak a kulcs és értéket kell megadni.