

1. gyakorlat anyaga

Demo

1. feladat

Készítsük el a lehető legrövidebb Java fordítási egységet.

```
public class A {}
```

Fordítsuk is le.

```
javac A.java
```

2. feladat

Készítsünk futtatható, üres törzsű programot.

```
public class Skip {  
    public static void main(String[] args) {  
    }  
}
```

Fordítsuk és futtassuk.

```
javac Skip.java  
java Skip
```

Feladatok

3. feladat

Készítsünk programot, amely a billentyűzetről kéri be a felhasználó nevét, majd üdvözlí. Használjuk az alábbi két parancsot.

```
System.console().readLine()  
System.console().printf(...)
```

4. feladat

Készítsünk programot, amely a felhasználónevet parancssori argumentumként kapja, majd üdvözlí a felhasználót. A szabványos kimenetre történő kiírás ezzel a parancssal is elvégezhető - ezt használjuk majdnem mindig.

```
System.out.println(...)
```

5. feladat

Az alábbi Java program feladata, hogy kiírja az 1-től 4-ig lévő számok felét.

Az elvárt kimenet:

```
0.5
1.0
1.5
2.0
Kiírtam 4 számot
```

Javítsa ki a programot.

```
public class Print {
    public static void main() {
        for(int i = 1; i < 4; i++) {
            System.out.println(i/2);
        }
        i = 4;
        System.out.println("Kiírtam " + i + " számot")
    }
}
```

6. feladat

Készítsünk programot, amely bekér két egész számot, és kiírja a köztük lévő egész számok felét. A beolvasás során kapott sztringeket egész számmá az alábbi konverziós függvénnyel alakíthatjuk át.

```
Integer.parseInt(...)
```

7. feladat

Készítsünk egy programot, amely kiszámolja két egész szám összegét, különbségét, szorzatát, hányadosát, és az osztási maradékot is megadja. Figyeljen a nullával való osztásra (ez esetben ne végezze el az osztást). A két számot parancssori paraméterként kell megadni. Vizsgáljuk meg azt is, hogy megfelelő számú parancssori paramétert adtunk-e át.

8. feladat

Írjuk meg az  faktoriálisát kiszámoló programot.

Gyakorló feladatok

1. gyakorló feladat

Készítsünk programot, amely a parancssori argumentumában megadott számról eldönti, hogy tökéletes-e. A számelméletben tökéletes számnak nevezzük azokat a természetes számokat, amelyek megegyeznek az önmaguknál kisebb osztók összegével (vagyis az összes osztók összege a szám kétszerese). Az első négy ilyen szám: 6, 28, 496 és 8128.

2. gyakorló feladat

Módosítsuk az előző megoldásunkat úgy, hogy 1-től a paraméterként megadott határig minden egész számra vizsgálja a tökéletességet és megadja, hogy abban az intervallumban mennyi ilyen talált. Ha nem talált egyetlen ilyen számot sem, akkor írja ki, hogy egyetlen tökéletes szám sincs a megadott intervallumban.

3. gyakorló feladat

Számoljuk ki a parancssori argumentumként kapott egész számok legnagyobb közös osztóját. (Nem csak 2 számét, hanem az összes kapott számét.)

A paraméterként kapott egész számok és az eredmény is legyen `long` típusú. Sztringeket ilyen típusú értékre a `Long.parseLong(...)` függvénnyel alakíthatunk.

4. gyakorló feladat

Írjunk programot, mely kiszámítja egy szám négyzetgyökét a babilóniai módszerrel.

Az (S) szám négyzetgyökének megállapításához számoljuk ki az alábbi (x_n) sorozat első néhány elemét. A program kérje be az (S) számot, és kérjen be egy (ϵ) pontosságot. A számolást addig folytassuk, amíg két egymást követő (x_i) érték különbsége nagyobb, mint (ϵ) . A sorozat legutolsó kiszámított elemét írjuk ki.

$$x_0 = \left\{ \frac{S}{2} \right\} \quad x_{n+1} = \left\{ \frac{1}{2} \left(x_n + \frac{S}{x_n} \right) \right\}$$

Egy sztringet a `Double.parseDouble(...)` függvénnyel konvertálhatunk `double` típusúvá.

5. gyakorló feladat

Számoljuk ki $(\sqrt{2})$ értékét közelítőleg lánctört kiértékelésével.

$$\sqrt{2} - 1 = \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \dots}}}$$

6. gyakorló feladat

Számoljuk ki (e) értékét közelítőleg lánctört segítségével.

$$e - 1 = 1 + \frac{1}{1 + \frac{1}{2 + \frac{1}{1 + \frac{1}{1 + \frac{1}{4 + \frac{1}{1 + \frac{1}{1 + \frac{1}{6 + \dots}}}}}}}}}$$