

# Programozási nyelv Java

---

## 1. gyakorlat

### Félév tematikája

- Bevezetés, alapok
- Vezérlési szerkezetek
- Eljárások, függvények
- Csomagok
- Tömbök, adatbevitel
- Osztályok, objektumok,
- Interfészek
- Öröklődés, absztrakt osztály
- Fájlkezelés, Kivételekezés
- Gyűjtemények
- Sablonok
- GUI, Adatbázis kezelés

### Követelmények

- 2 nagy beadandó
- 2 központi zh
- min. 8 kis beadandó:  
kis beadandók heti rendszerességgel az óra végén kerülnek kiadásra  
határidejük mindig a következő óra eleje

### Java bevezető

#### Történet

- James Gosling kezdte fejleszteni a Sun Microsystemsnél a 90-es évek elején
- Első kiadás 1995-ös, azóta sok minden változott, bővült a nyelv
- A Java 2 Platform, Standard Edition (SE) 5.0 2004-es

## Jellemzők

- Objektorientált nyelv (annyira, hogy a primitív típusokon kívül minden az Object osztály leszármazottja)
- C++ szerű szintaxis
  - a fordító bájtkódra fordít, a futtatásért a Java Virtual Machine (JVM) felel
  - a bájtkód hordozható platformok között (Windows, Unix, Linux, Macintosh), ami nagyon nagy előny
  - a programok relatíve lassúak: persze ez sokszor nem számít vagy nem érzékelhető
  - könnyebben elsajátítható nyelv, mint a C++

## A nyelvről

- Garbage Collector (automatikus szemétyűjtés) van
  - Így nehéz elrontani a memória felszabadítást
  - Néha azért felgyűlik a „szemét”
- Nincs memóriafelszabadító utasítás (mint a delete a C++ esetén)
- 8 primitív típus (boolean, char, byte, short, int, long, float, double)
- Az Object osztály (java.lang.Object) minden további osztály őse
- Javadoc: viszonylag kényelmes dokumentációs kommentezést tesz lehetővé nagy hangsúly van a kivételkezelésen

## Java programok

- Egy java fájl tartalmaz egy azonos nevű osztályt (pl: a Program.java fájlban van egy Program osztály)
- A függvények osztálydefinícióban fordulhatnak elő
- Más osztály függvényeinek használatához importálni kell a megfelelő osztályt
- Fordításkor a Program.java-ból készül egy Program.class fájl ('javac' paranccsal)
- Ha az osztálynak van 'main()' metódusa (belépési pontja), akkor végrehajtható (a 'java' paranccsal)
- Az osztályok úgynevezett 'package'-ekbe (csomagokba) szerveződnek (pl. program.gui, program.engine, stb.)
  - egy package-be tartozó osztályok látják egymást (nem kell importálni)
  - egy package-be tartozó osztályok alapértelmezésben látják egymás adattagjait

## Fordítás és futtatás (parancssorból)

A .java forrásfájlokból a Java fordító .class kiterjesztésű fájlt állít elő. Ez a fájl tartalmazza a program bájtkódját. Egy forrásfájl neve a fájlban implementált osztály nevével egyezik meg. Ha a

megvalósított osztály valamilyen package-be (névtérbe) tartozik, akkor a forrásfájlt a package nevével azonos alkönyvtárba kell elhelyezni.

## Környezet beállítása Konzolban:

Ha nem ismeri fel a konzolban a javac parancsot, akkor fel kell venni környezeti változóba a Java JDK bin könyvtárának útvonalát.

```
set PATH=%PATH%;c:\Program Files\Java\jdk1.6.0_20\bin\
```

## Fordítás

- javac paranccsal történik
- -d argumentummal megadható, hogy milyen könyvtárba kerüljenek a class fájlok
- -encoding argumentummal beállítható a karakterkódolás
- Példa: javac HelloWorld.java (abszolút elérési út v relatív az adott helytől)

## JavaDoc készítése

- javadoc paranccsal történik
- -d argumentummal megadható, hogy milyen könyvtárba kerüljön a dokumentáció
- -encoding argumentummal beállítható a karakterkódolás
- Példa: javadoc -d doc -encoding utf-8 alakzat/\*

## Futtatás

- java parancs
- -cp argumentummal beállítható, hogy honnan vegye a class fájlokat
- Példa: java HelloWorld

Ha nem találja az operációs rendszer a fenti parancsokat, akkor előfordulhat, hogy telepíteni kell a Javat vagy a Java bin könyvtárát fel kell venni a PATH környezeti változóba.

## Ingyenes fejlesztő környezetek

Mi gyakorlaton nem fogunk IDE-t használni.

- Eclipse:
  - <http://eclipse.org/>
  - rengeteg plugin tölthető le hozzá
- NetBeans:
  - <http://netbeans.org/>
  - használható beépített GUI szerkesztője van (de mi nem fogjuk használni)

## Kódolási konvenciók

- Fontos betartani valami kódolási konvenciót.
- Fő szempontok: olvashatóság, egyértelműség, értelmezhetőség
- Oracle által javasolt kódolási konvenciók a Java nyelvre:  
<http://www.oracle.com/technetwork/java/codeconv-138413.html>

## Dokumentáció

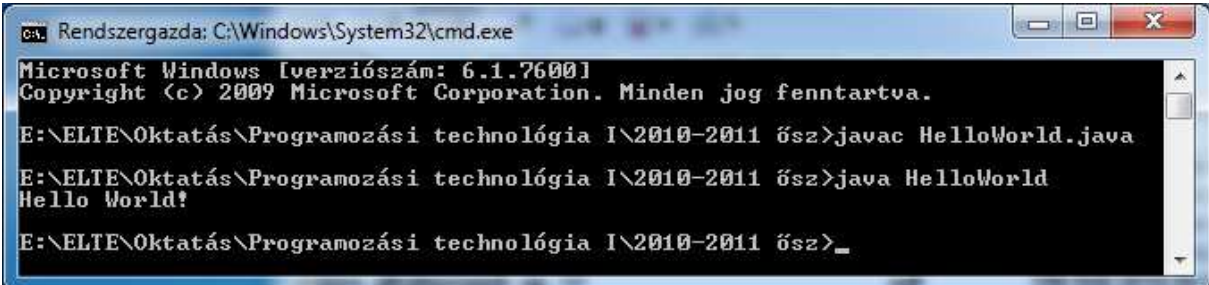
- Az előadás honlapja: <http://people.inf.elte.hu/sike/PT1C/pt1c.html>
- Java tutorial: <http://docs.oracle.com/javase/tutorial/>
- Javadoc:
  - <http://docs.oracle.com/javase/6/docs/>
  - <http://docs.oracle.com/javase/6/docs/api/>
  - <http://docs.oracle.com/javase/7/docs/>
  - <http://docs.oracle.com/javase/7/docs/api/>

## Hello World program

```
public class HelloWorld{  
    public static void main( String[] args ){  
        System.out.println( "Hello world!" );  
    }  
}
```

A **class** kulcsszóval definiáltunk egy HelloWorld nevű osztályt, aminek 1 darab metódusa a **main**. Ez a program belépési pontja. A System.out.println paranccsal tudunk a konzolra kiíratni szöveget.

A kimenet a következő lesz (parancssoros fordítással + futtatással):



```
ca. Rendszergazda: C:\Windows\System32\cmd.exe  
Microsoft Windows [verziószám: 6.1.7600]  
Copyright (c) 2009 Microsoft Corporation. Minden jog fenntartva.  
E:\ELTE\Oktatás\Programozási technológia I\2010-2011 őszi>javac HelloWorld.java  
E:\ELTE\Oktatás\Programozási technológia I\2010-2011 őszi>java HelloWorld  
Hello World!  
E:\ELTE\Oktatás\Programozási technológia I\2010-2011 őszi>_
```

# Típusok

## Primitív típusok

- <http://download.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>
- byte, short, int, long, float, double, char, boolean
- default értékekkel rendelkeznek (0, false, etc.)
- oktális (int octVal = 01), hexa érték (byte hexVal = 0xff), scientific notation (double d = 1.23e4)
- wrapper osztályok (Byte, Short, Integer, ... )
  - automatikus boxing, unboxing

Ezen kívül vannak további típusok, mint például tömbök, String típus, amikkel majd a későbbi gyakorlatokon fogtok megismerkedni.

## Konverziók általában

- bővítés automatikus
- szűkítés típuskényszerítéssel pl: byte b = (byte)10

## Szövegkonverzió

- Stringgé: String s = "" + 1; (precedenciára figyelni!) vagy a toString() függvény
- Stringből: Integer.parseInt("1"), Double.parseDouble("2"), ...

## Operátorok

- ==, !=, &&, ||, %, ++, ...
- Részletes táblázat:  
<http://java.sun.com/docs/books/tutorial/java/nutsandbolts/operators.html>

Összehasonlító operátor **!=** oldalára lehetőleg **konstanst** írjunk a kellemetlen helyzetek elkerülése érdekében:

```
boolean b = true;
if (b!=false) {

}
```

# Vezérlési szerkezetek

## Elágazások

```
if ( <boolean feltétel> ) {  
  
} else if ( <boolean feltétel> ) {  
  
} else if ( <boolean feltétel> ) {  
  
} else {  
  
}
```

Switch: byte, short, char, int típusokra (ill. ezek wrapper osztályaira is: Character, Byte, Short, Integer) használható (long-ra nem).

Példa:

```
final int month = 8;  
switch (month) {  
    case 1: System.out.println("Jan"); break;  
    case 2: System.out.println("Feb"); break;  
    case 3: System.out.println("Mar"); break;  
    case 4:  
    case 5:  
    case 6: System.out.println("Apr, Maj vagy Jun"); break;  
    default: System.out.println("Egyeb honap");break;  
}
```

## Ciklusok

Az előltesztelős while ciklus:

```
while ( <boolean feltétel> ) {  
  
}
```

A hátul tesztelős while ciklus:

```
do {  
  
} while ( <boolean feltétel> );
```

A for ciklus:

```
for (<inicializálás>; <termináló feltétel>; <léptetes>) {  
  
}
```

A foreach ciklus:

```
for(<T típusú változó> : <T típusú kollekción>) {  
  
}
```

A végtelen ciklusok:

```
for ( ; ; ) {  
  
}  
  
while (true) {  
  
}
```

## Branching kifejezések

- break: kugrik a vezérlés a ciklusból
- continue: továbblép a következő iterációra
- return: visszatérés egy függvényből vagy eljárásból

## Feladatok

1. Írd át a mintaprogramot (HelloWorld-t) úgy, hogy a saját nevedet írja ki. Pl: „Hello Dorian!”
2. Adott egy int típusú változóban egy egész szám. Írjatok egy programot, ami kiírja magát a számot és azt, hogy az páros vagy páratlan. Például: „A(z) 2 páros!”
3. Készítsünk egy hőmérséklet-konvertáló programot! Vegyünk fel egy logikai és egy valós értéket! Ha a logikai érték „Igaz”, akkor konvertáljuk a valós értéket Celsiusról Fahrenheit fokra a lenti képlet alapján, egyébként Fahrenheitről Celsiusra.

$$C = \frac{(F - 32) * 5}{9}$$

4. Készítsünk egy minimális konzolos számológépet! Adjunk meg három szám paramétert! Ha az első szám:

- 1, akkor adjuk össze
- 2, akkor vonjuk ki
- 3, akkor szorozzuk össze
- 4, akkor osszuk el a másik két paramétert egymással
- Minden egyéb esetben írjuk ki, hogy nem értelmezett művelet

A megoldáshoz használjunk switch-case szerkezetet!

5. Írj egy programot, amely kiírja a számokat 1-től N-ig (N<100, ezt ellenőrizzük is)! Azon számokhoz, amelyek hárommal oszthatók, a szám helyett "Fizz"-t írjon ki, és azok helyett, amelyek ötten oszthatók, "Buzz"-t. Azon számok helyett, amelyek mind hárommal, mind ötten oszthatók, "FizzBuzz"-t írjon ki a program!