

Programozási nyelvek – Java

C-ből Javába



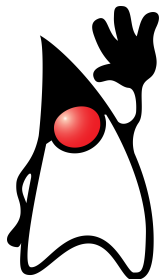
Kozsik Tamás

ELTE Eötvös Loránd Tudományegyetem

A Java nyelv

- C-alapú szintaxis
- Objektumelvű (object-oriented)
 - Osztályalapú (class-based)
- Imperatív
 - Újabban kis FP-beütés
- Fordítás bájt kódra, JVM
- Erősen típusos
- Statikus + dinamikus típusrendszer
- Generikus, konkurens nyelvi eszközök

Java Language Specification



- Könnyű/olcsó szoftverfejlesztés
- Gazdag infrastruktúra
 - Szabványos és egyéb programkönyvtárak
 - Eszközök
 - Kiterjesztések
 - Dokumentáció
- Platformfüggetlenség (JVM)
 - Write once, run everywhere
 - **Compile** once, run everywhere
- Erőforrásintenzív

JavaZone videó



James Gosling és mások, 1991 (SUN Microsystems)

Oak → Green → **Java**

- Java 1.0 (1996)
- Java Community Process (1998)
- Java 1.2, J2SE (1998)
- J2EE (1999)
- J2SE 5.0 (2004)
- JVM GPL (2006)
- Oracle (2009)
- Java SE 8 (2014)
- Java SE 11 (2018) LTS
- Java SE 15 (2021)
- Game of Codes, Javazone 2014



- Alacsonyszintű nyelv: bájtkód
- Sok nyelv fordítható rá (Ada, Closure, Eiffel, Jython, Kotlin, Scala...)
- Továbbfordítható
 - Just In Time compilation
- Dinamikus szerkesztés
- Kódmobilitás

Java Virtual Machine Specification



C és Java hasonlósága

```
int lnko( int a, int b ){  
    while( b != 0 ){  
        int c = a % b;  
        a = b;  
        b = c;  
    }  
    return a;  
}
```



C és Java különbsége

```
double sum( double array[] ){  
    double s = 0.0;  
    for( int i=0; i<array.length; ++i ){  
        s += array[i];  
    }  
    return s;  
}
```



C és Java különbsége - hangsúlyosabban

```
double sum( double[] array ){  
    double s = 0.0;  
    for( double item: array ){  
        s += item;  
    }  
    return s;  
}
```



Hello World!

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



- 1 Objektumelvű programozás
- 2 Java programok fordítása és futtatása

Objektumelvű programozás

Object-oriented programming (OOP)

- Objektum
- Osztály
- Absztrakció
 - Egységbe zárás (encapsulation)
 - Információ elrejtése
- Öröklődés
- Altípusosság, altípusos polimorfizmus
- Felüldefiniálás, dinamikus kötés



Egységbezárás: objektum

Adat és rajta értelmezett alapl műveletek (v.ö. C-beli struct)

- “Pont” objektum
- “Racionális szám” objektum
- “Sorozat” objektum
- “Ügyfél” objektum

```
p.x = 0;
```

```
p.y = 0;
```

```
p.move(3,5);
```

```
System.out.println( p.x );
```



Metódus

Java

```
p.x = 0;  
p.y = 0;  
p.move(3,5);
```

C

```
p.x = 0;  
p.y = 0;  
move(p,3,5);
```



Osztály

Objektumok típusa

- “Pont” osztály
- “Racionális szám” osztály
- “Sorozat” osztály
- “Ügyfél” osztály

```
class Point {  
    int x, y;  
    void move( int dx, int dy ){...}  
}
```



Példányosítás (instantiation)

- Objektum létrehozása osztály alapján
- Javában: mindig a heapen

```
Point p = new Point();
```



Példa: szövegek

```
String name = "James Arthur Gosling";  
String[] names = name.split(" ");  
String abbrev = names[names.length-1] + ", "  
                + names[0].charAt(0) + ".";
```



- 1 Objektumelvű programozás
- 2 Java programok fordítása és futtatása

Java programok felépítése

(első blikkre)

- [modul (module)]
- csomag (package)
- osztály (class)
- tag (member)
 - adattag (mező, field)
 - metódus (method)



Java forrásfájl

- Osztálynévvel
- .java kiterjesztés
- Fordítási egység
- Csomagjának megfelelő könyvtárban
- Karakterkódolás



Fordítás, futtatás

- A „tárgykód” a JVM bájtkód (.class)
- Nem szerkesztjük statikusan
- Futtatás: bájtkód interpretálása + JIT

Parancssorban

```
$ ls
HelloWorld.java
$ javac HelloWorld.java
$ ls
HelloWorld.class HelloWorld.java
$ java HelloWorld
Hello world!
$
```



Java programok futása

- Végrehajtási verem (execution stack)
 - Aktivációs rekordok
 - Lokális változók
 - Paraméterátadás
- Dinamikus tárhely (heap)
 - Objektumok tárolása

