

Eötvös Loránd Tudományegyetem
Informatikai Kar

Programozási technológia I

1. gyakorlat

A Java nyelvi alapjai, a NetBeans környezet

© 2012.02.16. Giachetta Roberto

groberto@inf.elte.hu

<http://people.inf.elte.hu/groberto>

A Java nyelv alapjai

A nyelv lehetőségei

- Tisztán objektumorientált programozási nyelv, amelyet a Sun Microsystems 1995-ben publikált
 - a legnépszerűbb programozási nyelv (17%-os népszerűség a TIOBE toplistán)
 - jelenlegi verzió a Java 7, az Oracle tulajdonában
- A fejlesztési elvek:
 - egyszerű, objektumorientált és ismerős legyen
 - rugalmas és biztos legyen
 - platform-független és hordozható legyen
 - nagy teljesítménnyel fusson
 - értelmezett, többszálú és dinamikus legyen

A Java nyelv alapjai

A nyelv lehetőségei

- Célja egy C++-szerű, de könnyebben kezelhető nyelv megvalósítása, amely teljes mértékben hordozható és alkalmas webes, hálózati valamint mobil alkalmazások készítésére
- Három változatban érhető el a különböző célokra:
 - *Standard Edition* (SE),
 - *Enterprise Edition* (EE),
 - *Micro Edition* (ME),
- Az Oracle-től független fejlesztést végez a Google az *Android* platformon, amely egy Linuxra épülő saját Java virtuális gép és ré épülő osztálykönyvtár

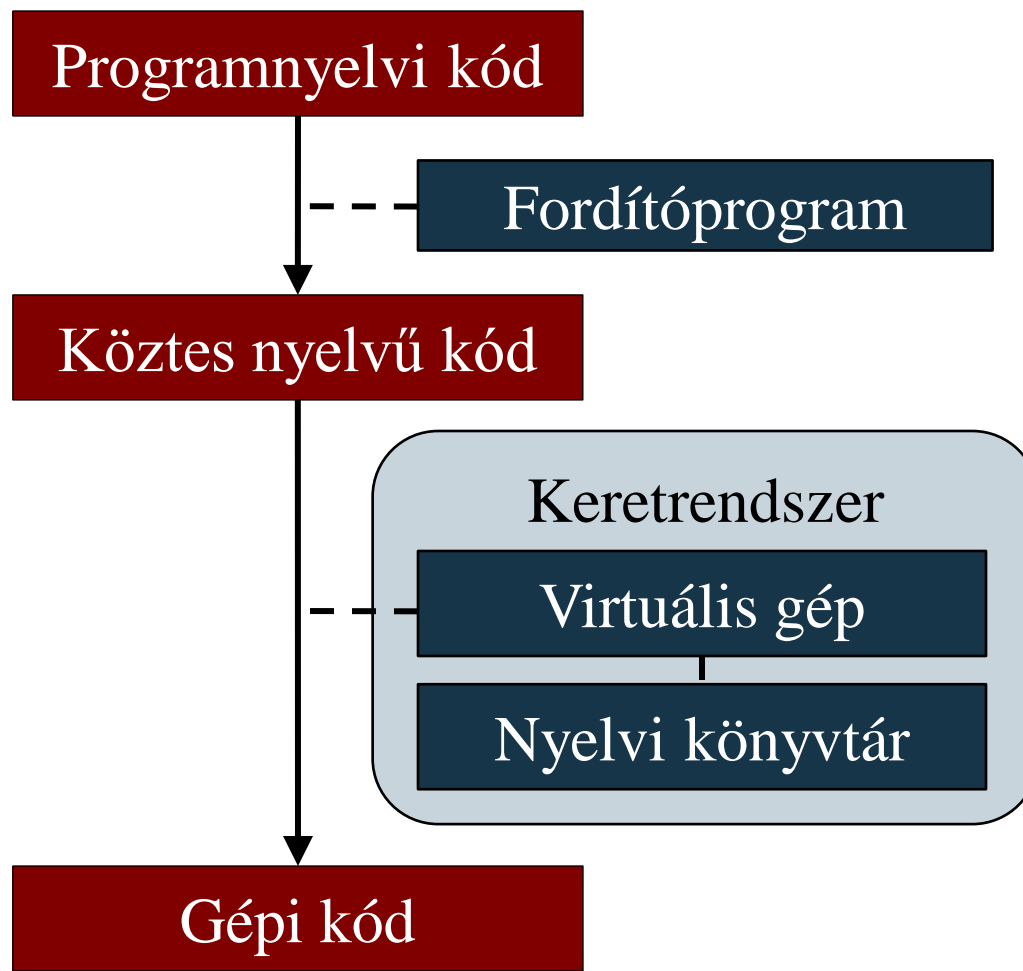
A Java nyelv alapjai

A Java környezet

- A Java egy teljes keretrendszer biztosít számos platformra a hordozhatóság végett
 - a programkód gépfüggetlen bytekódra (köztes szintű kódra) fordul
 - ezt interpretálja a futtatási környezet (*Java Virtual Machine, JVM*) futásidejű fordítással (Just In Time Compilation)
 - így elég az alkalmazásokat egyszer fordítani, és utána bárhol futtathatóak, ahol jelen van a környezet (*Java Runtime Environment*), amely magában foglalja a virtuális gépet, valamint a nyelvi könyvtárakat
 - cserébe a végrehajtás lassabb, az erőforrásigény magasabb, mint a teljesen fordított alkalmazásokban

A Java nyelv alapjai

A Java környezet



A Java nyelv alapjai

A nyelv lehetőségei

- Támogatja az objektumorientáció minden eszközét
 - osztályok, objektumok
 - adatsztruktúra, adatelrejtés, egységbe záras
 - öröklődés, polimorfizmus, dinamikus kötés
 - kivételkezelés, sablonok
- Fordítási egységek a fájlok
 - ha a fájlban van publikus típus, akkor a neve megegyezik a fájléval (tehát egy fájlban csak egy publikus típus lehet)
 - a forrásfájl kiterjesztése: **java**, a bytekód kiterjesztése: **class**
 - a bytekód fájlok összecsomagolva is futtathatóak, ekkor a kiterjesztés: **jar**

A Java nyelv alapjai

A „Hello World” program

```
import Java.io.*;
    // ki- és bementéért felelős csomagok használata

class Hello {
    // még a főprogram is osztályban van

    public static void main( String[] args ){
        // a főprogram egy statikus metódus

        System.out.println("Hello World!");
        // a kiírás a megfelelő metódushívás
    }
}
```

A Java nyelv alapjai

Karakterkódolás, literálok

- A kódoláshoz 16 bites Unicode karakterkészletet használ
- Számliterálok:
 - decimális: `377`, oktális: `0377`, hexadecimális: `0x377`
 - 64 bites tárolás: `377L`
 - lebegőpontos: `12.0`, `12.6e4`, szimpla pontosság: `12.6e4F`, dupla pontosság: `12.6e4D`
 - lebegőpontos nulla: `0.0`, `-0.0`
- Szövegliterálok:
 - karakterek: `'k'`, szöveg: `"hello"`
 - speciális karakterek: `\\`, `\n`, `\t`, `\'`, `\"`, `\b` (vissza), `\r` (sorvissza), `\f` (lapdobás)

A Java nyelv alapjai

Vezérlési szerkezetek

- Szekvencia: `<utasítás>;`
- Programblokk: `{ <utasítás> }`
 - utasítások csoportosítására
 - megszabja az élettartamot (programblokkon belül deklarált változó csak a programblokk végéig érhető el)
- Elágazások:
 - `if(<feltétel>) <utasítás>; else <utasítás>;`
 - a feltétel logikai típusú kifejezés
 - tetszőlegesen egymásba ágyazhatóak (nincs `elif`)
 - a csellengő `else` mindig az utolsó elágazáshoz tartozik
 - Triáris operátor (elágazó értékadáshoz):
`<feltétel> ? <utasítás> : <utasítás>;`

A Java nyelv alapjai

Vezérlési szerkezetek

- `switch(<változó>){`
 - `case <konstans> : <utasítások>; break;`
 - `default: <utasítások>; break;``}`
- alkalmazható egész és szöveg változókra, felsorolási típusokra
- alapértelmezett (`default`) ág nem kötelező
- Ciklusok:
 - számláló:
`for(<inicializálás>; <feltétel>; <léptetés>)`
`<utasítás>;`
 - előtesztelő: `while(<feltétel>) <utasítás>;`

A Java nyelv alapjai

Vezérlési szerkezetek

- utántesztelő: `do <utasítás>; while(<feltétel>);`
- iteráló (tetszőleges gyűjtemény bejárásához):
`for (<típus> <változó>: <adatszerkezet>)
 <utasítás>`
- ciklusból azonnali kilépés a `break` utasítással
- feltétel kiértékeléshez ugrás a `continue` utasítással
- Pl.:

```
public static void main(String[] args){  
    int fakt = 1; // faktoriális számítás  
    for (int i = 1; i <= 5; ++i)  
        fakt = fakt * i;  
    System.out.println(fakt); // eredmény: 120  
}
```

A Java nyelv alapjai

Operátorok

- Az operátorok olyan függvények, amelyek speciális meghívással rendelkeznek, vezérlőkaraktereken, vagy kulcsszavakon keresztül
 - a meghívás rögzített formában történik (*prefix*, *postfix* vagy *infix* jelölés mellett)
 - az operátorok *precedenciával* rendelkeznek, amely halmozás esetén megszabja a hívási sorrendet
 - az operandusok száma rögzített, vannak egy-, két-, illetve háromoperandusú műveletek (a + és - operátortoknak van egy-, illetve kétoperandusú változata is)
- Minden beépített típus rendelkezik operátorokkal, amelyek a megadott működést végzik

A Java nyelv alapjai

Operátorok

- Aritmetikai:
 - pozitivitás ($+a$), negáció ($-a$)
 - értéknövelés ($a++$, $++a$), értékcsökkentés ($a--$, $--a$)
 - összeadás ($a + b$), kivonás ($a - b$), szorzás ($a * b$), osztás (a / b), maradékképzés ($a \% b$)
- Értékadás:
 - egyszerű ($a = b$)
 - összetett ($a += b$, $a -= b$, $a *= b$, $a /= b$, $a \% = b$, $a << = b$, $a >> = b$, $a >>> = b$, $a \& = b$, $a |= b$, $a \wedge = b$)
 - feltételes ($a ? b : c$)
- Típusazonosítás (`instanceof`)

A Java nyelv alapjai

Operátorok

- Logikai:
 - érték összehasonlítás ($a < b$, $a > b$, $a \leq b$, $a \geq b$, $a == b$, $a != b$)
 - tagadás ($!a$), és ($a \&\& b$), vagy ($a || b$)
- Bitenkénti:
 - eltolás balra ($a \ll b$), eltolás jobbra ($a \gg b$), előjel nélküli eltolás jobbra ($a \ggg b$)
 - komplementer képzés ($\sim a$), bitenkénti és ($a \& b$), bitenkénti vagy ($a | b$), különbségképzés ($a \wedge b$)

A Java nyelv alapjai

Operátorok precedenciája

- Az operátorok precedenciája meghatározza, a műveletek halmozása esetén milyen sorrendben történik a végrehajtás
 - a precedencia típusfüggetlen, és rögzített
 - a magasabb precedenciájú hajtódik előbb végre
 - zárójelek befolyásolják a végrehajtási sorrendet
- Java precedenciák:
 1. ++ (postfix), -- (postfix)
 2. + (unáris), - (unáris), !, ~, ++ (prefix), -- (prefix),
 3. *, /, %
 4. +, -
 5. <<, >>, >>>
 6. >, <, >=, <=, instanceof

A Java nyelv alapjai

Operátorok precedenciája

7. ==, !=

8. &

9. ^

10. |

11. &&

12. ||

13. ?:

14. =, +=, -=, *=, /=, %=, <<=, >>=, &=, |=, ^=

- Pl.:

$a * b - c == 7$ jelentése: $((a * b) - c) == 7$

$c = a == b \% 2$ jelentése: $c = (a == (b \% 2))$

$++a++$ jelentése: $++(a++)$

A Java nyelv alapjai

Csomagok

- A programok tagolása magasabb szinten csomagokkal történik, egy csomagban tetszőleges számú további csomag, illetve osztály helyezkedhet el
 - a csomagok az osztályok logikai összetartozását fejezik ki
 - egy típus pontosan egy csomagba tartozik
 - alapértelmezett a névtelen csomag
- Csomag definíció:
 - `package <csomagnév>;`
 - `package <csomagnév>.<alcsomagnév>;`
- Csomag használat:
 - `import package <csomagnév>.*;`
 - `import package <csomagnév>.<típusnév>;`

A Java nyelv alapjai

Típusrendszer

- Erősen típusos a nyelv
 - automatikus konverzió altípusra
 - explicit konverzió saját típusokra futásidejű ellenőrzéssel
- Típusok kezelése:
 - *primitív típusok*:
 - érték szerint, automatikus kezdőértékkel
 - blokk végig, vagy az objektum törléséig élnek
 - *összetett típusok*:
 - referencia (igazából mutató) szerint, `null` kezdőértékkel
 - a `new` operátorral példányosíthatóak
 - a törlést a szemétgyűjtő végzi, nincs manuális törlés

A Java nyelv alapjai

Típusrendszer

- Primitív típusok:
 - **boolean** (true vagy false)
 - **char** (16 bites Unicode karakter)
 - **byte** (8 bites előjeles egész)
 - **short** (16 bites előjeles egész)
 - **int** (32 bites előjeles egész)
 - **long** (64 bites előjeles egész)
 - **float** (32 bites lebegőpontos szám)
 - **double** (64 bites lebegőpontos szám)
- Összetett típusok: osztályok, interfészek, felsorolási típusok, tömbök

A Java nyelv alapjai

Tömbök és felsorolási típus

- Tömbök:
 - indexelés 0-tól hossz-1-ig
 - szétválasztódik a deklaráció és a definíció:
`<típus>[] <név>; <név> = new <típus>[<méret>];`
 - lehet inicializálni:
`<típus>[] <név> = { <kezdőértékek> };`
 - tisztában van a hosszával: `<név>.length`
 - másolás az `arraycopy` művelettel
 - több dimenziós előállítható tömbök tömbjeként
- Felsorolási típus: `enum <típusnév> { <értékek> }`
 - rendelkezhet attribútumokkal, függvényekkel, konstruktorral

A Java nyelv alapjai

Tömbök és felsorolási típus

- Pl. egy vektor (egy dimenziós tömb):

```
int[] arr = new int[10];  
for (int i = 0; i < arr.length; i++)  
    arr[i] = i + 1; // tömb értékei 1..10 lesznek
```

- Pl. egy mátrix (két dimenziós tömb):

```
String[][] names = {{"Mr. ", "Mrs. ", "Ms. "},  
                    {"Smith", "Jones"}};  
  
// mátrix, aminek első sora a megszólítások  
// (3 hosszú), második a nevek (2 hosszú)  
System.out.println(names[0][0] + names[1][0]);  
// eredménye: Mr. Smith  
System.out.println(names[0][2] + names[1][1]);  
// eredménye: Ms. Jones
```

A Java nyelv alapjai

Tömbök és felsorolási típus

- Pl. felsorolási típus:

```
enum Napok { Hétfő, Kedd, Szerda, Csütörtök };  
// napokat tartalmazó felsoroló típus  
Napok maiNap;  
// mai nap bekérése...  
switch (maiNap) {  
    case Hétfő: System.out.println("Még 5 nap");  
                break;  
    case Kedd: System.out.println("Még 4 nap");  
                break;  
    ...  
    default: System.out.println("Nem munkanap");  
                break;  
}
```