

Java tutorial

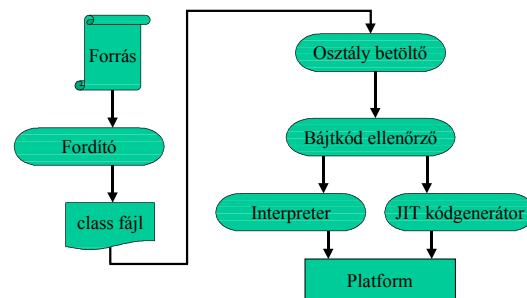
Copyright © 2000-2001, Kozsik Tamás

Java Technológia Standard Edition

A kódolás után...

- Forrásfájlok `.java`
- „Félig-meddig” lefordítjuk pl. `javac`
- Bájt kód `.class`
- Interpretáljuk pl. `java`
- Platformfüggetlenség, biztonság

Lerajzolva:



Fordítás

- Ha abban a könyvtárban vagyunk, ahol a fájl: `javac Hello.java`
- Ha egyet lejjebb, pl. `cd ..` után `javac udv/Hello.java`
- Ha több fájlt is le akarunk fordítani `javac udv/*.java`

Fordítás (több fájl)

- Írjuk össze a lefordítandó forrásfájlok nevét egy fájlba, pl. a `forrasok` fájlba.

```
$ cat forrasok
Hello.java
Szia.java
Salut.java
$ javac @forrasok
```

A fordítási egységek

- ... több típusdefiníciót is tartalmazhatnak
- Mindegyikhez keletkezik egy `.class` fájl

```
                                Csavar.java
package raktar.adatok;
public class Csavar extends Anyag {...}
class CsavarMenet {...}
```

- A fordítás során két fájl keletkezik:
`Csavar.class` `CsavarMenet.class`

A lefordított fájlok elhelyezése

- Követni kell a csomag-hierarchiát
 - különben a virtuális gép (**java**) tanácstalan lesz
- Alkönyvtárak létrehozása
- Például a `raktar.adatok.Csavar` osztály `.class` fájlját a `raktar/adatok` alkönyvtárba kell tenni

`raktar/adatok/Csavar.class`

A -d kapcsoló

- Nem kell kézzel másolgatni: bizzuk rá a `javac` fordítóra
- Megmondjuk, mi legyen a célkönyvtár
- Azon belül felépíti a szükséges könyvtárstruktúrát

```
javac -d . Csavar.java
javac -d ../classes Csavar.java
```

A források elhelyezése

- Javaslat: ugyanolyan struktúrában, mint a `.class` fájlknál, azaz követve a csomaghierarchiát. Pl.:

```
src
  raktar
    adatok
    gui
    web
classes
  raktar
    adatok
    gui
    web
```

Forrásfájlok elhelyezése (2)

- Lehetnek egy könyvtárszerkezetben a `.class` fájlokkal
- Lehetnek két különböző, de azonos szerkezetű könyvtárstruktúrában a forrás és a `.class` fájlok
- Lehetnek egész máshogy is, pl. sok könyvtárban szétszórva

Java tutorial

Copyright © 2000-2001, Kozsik Tamás

A -sourcepath kapcsoló

- Ha sok könyvtárba vannak szétszórva a forrásfájlok, akkor használhatjuk a **-sourcepath** kapcsolót
- Megadható, hogy mely könyvtárakban kell keresni forrásfájlok után

```
javac -sourcepath ~/src/java:~/projects/raktar DB.java
```

Java tutorial

Copyright © 2000-2001, Kozsik Tamás

Hivatkozások a fordítási egységen kívülre

- Ha egy fordítási egység hivatkozik egy másik fordítási egységben levő típusra, akkor a **javac** fordító:
 - megpróbálja megtalálni a hozzá tartozó **.class** fájlt - ha egyáltalán le van már az a típus fordítva
 - ha nincs lefordítva, akkor megkeresi a megfelelő **.java** fájlt, és rekurzívan azt is lefordítja

Rekurzív fordítás

- A neve alapján keresi meg a hivatkozott típust tartalmazó forrásfájlt a **javac**
 - A fájlneveknek meg kell egyeznie a típusnévvel
 - A **-sourcepath** kapcsoló szerepe...

Hol keresse a .class fájlokat?

- Alapértelmezésben
 - a szabványos könyvtárakban és
 - az aktuális munkakönyvtárban
- A **-classpath** kapcsoló segítségével más is megadható

```
javac -classpath ~/classes:. DB.java
```

A szabványos könyvtárakban, a **~/classes** könyvtárban és az aktuális munkakönyvtárban.

A -classpath kapcsoló

- Megadható több könyvtárnév, amelyekben a **.class** fájlokat keresheti a fordító
 - a könyvtárnevek elválasztásához **UNIX** alatt **:** **Windows** alatt **;** használendő
- Nem csak könyvtárnevek, hanem **.zip** és **.jar** fájlok is megadhatók, azokban is keres
- A kapcsoló rövidebb alakja: **-cp**

Hogyan történik a keresés?

`javac -classpath ~/classes:. DB.java`

- Tegyük fel, hogy hivatkozás történik a `raktar.adatok.Csavar` osztályra.
- A szabványos könyvtárakban nem találja.
 - Ott `java.` és `javax.` kezdetű típusok vannak...
- Keresi a `~/classes/raktar/adatok/Csavar.class` fájlt
- Ha az sincs, keresi a `raktar/adatok/Csavar.class` fájlt
- Ha az sincs, keresi a forrását, a `raktar/adatok/Csavar.java` fájlt, hogy lefordítsa
 - a `-sourcepath` kapcsoló alapján
 - ha nincs megadva, akkor az aktuális könyvtárban

Osztálykeresési útvonal hagyományos megadása

- A `CLASSPATH` környezeti változó
- Manapság nem illendő használni
 - könnyen összekavarodnak a dolgok, ha több projekten is dolgozunk
- Helyette a `-classpath` kapcsoló

Java tutorial

Copyright © 2000-2001, Kozsik Tamás

A javac fordító egyéb kapcsolói

- Szabványos kapcsolók, pl.
 - g nyomkövetési infók
 - O optimalizálás
 - target kód verziószám (1.1, 1.2, ...)
 - nowarn ne legyenek figyelmeztetések
 - deprecation ... erről még lesz szó...
- Nem szabványos kapcsolók
 - Későbbi/korábbi fordítók nem biztos, hogy ismerik
 - Pl. heap méretének beállítása

A virtuális gép indítása

- A java parancsnak is adhatunk kapcsolókat
- A `-classpath` ugyanúgy, mint fordításkor
 - az ott megadott helyeken keresi az osztályokat az osztálybetöltő
- Más szabványos kapcsolók, pl. `-version`
- Ha a főprogram nem névtelen csomagban van: teljes osztálynevet kell megadni

`java -classpath ~/classes:. raktar.Indit`

- Mit is jelent a `java Hello.class` parancs?

Feladat

- Nézzük meg, milyen opciói vannak a fordítónak és a virtuális gépnek!

Deprecation

- Az API néhány eleme elavulhat az újabb és újabb verziók kihozatalakor
 - osztályok, interfészek, metódusok
- A **deprecated** szóval jelölik meg a dokumentációban
- Nem illeszkedik az aktuális Java szemlélethez
 - pl. internacionalizáció
- Újabb API verziók nem biztos, hogy támogatni fogják
- Használatuk kerülendő

Deprecated feloldása

- A fordító figyelmeztetést küld, ha deprecated elemet használunk
- Fordítsuk újra a -deprecation kapcsolóval
- Hibalistát kapunk azokról az elemekről, amelyek deprecated állapotúak
- A dokumentáció megadja, mi kell helyette
- Érdemes a javítást elvégezni, hogy a programunk hosszabb életű legyen...

Feladat

- Hozzunk létre egy java.util.Date objektumot a születési időnkkel, és írassuk ki. Tüntessük el a programunkból az elavult elemeket!

Java tutorial

Copyright © 2000-2001, Kozsik Tamás

A -D kapcsoló

- A virtuális gépnek paraméterek adhatók át
- System properties - rendszerjellemzők
- A programból lekérdezhetőek
- Vannak automatikusan beállítottak, és mi is létrehozhatunk újabbakat a -D kapcsolóval

Példa

```
import java.util.*;
public class Jellemzok {
    public static void main(String[] args) {
        Properties p = System.getProperties();
        Enumeration nevek = p.propertyNames();
        while ( nevek.hasMoreElements() ) {
            String név = (String) nevek.nextElement();
            String érték = p.getProperty(név);
            System.out.println(név + " = " + érték);
        }
    }
}

java -Dzaneve=valami Jellemzok
```

Szemelvények az eredményből (1)

`java -Dezaneve=valami Jellemzok`

```
java.version = 1.3
user.timezone = Europe/Paris
java.specification.version = 1.2
java.vm.vendor = Sun Microsystems Inc.
user.home = /h/teacher/kto
java.vm.specification.version = 1.0
os.arch = sparc
...
```

Szemelvények az eredményből (2)

`java -Dezaneve=valami Jellemzok`

```
...
user.language = en
user.name = kto
path.separator = :
file.separator = /
user.dir = /h/mnt/pandora/userlv/home/teacher/kto
ezaneve = valami
...
```

Java tutorial

Copyright © 2000-2001, Kozsik Tamás

Segédprogramok

- | | |
|-----------|-------------------------|
| • javac | fordító |
| • java | virtuális gép |
| • javadoc | dokumentáció generálása |
| • jar | Java ARchieve készítése |
| • ... | |

javadoc: HTML dokumentáció

- A programszövegből a specifikációk kigyűjtése (csomagnevek, osztálynevek, adattagok, metódusok, kivételek...)
- Dokumentációs megjegyzések beillesztése
`/** ... */`
- Speciális adatok a dokumentációs megjegyzésekben belül (ún. tag-ek)

javadoc tag-ek

- | | |
|---------------|----------------------------|
| • @author | szerző neve - email címe |
| • @version | verziószám |
| • @param | paraméter dokumentálása |
| • @return | visszatérési érték dokum. |
| • @throws | kiváltott kivételek dokum. |
| • @see | hivatkozás, link |
| • @deprecated | elavultság jelzése |

```

/** Hagományos verem adattípus.
    @author Kozsik Tamás
    @version 1.0
 */
public class Verem {
    /** A verem tartalma. */
    java.util.Vector adatok = new java.util.Vector();
    /** Betesz egy elemet a verembe.
        @see #pop
        @param adat Amit betesz.
    */
    public void push(Object adat){...}
    /** Kivesz egy elemet a veremből.
        @result Object A verem tetején levő elem.
        @throws UresException Ha üres a verem.
    */
    public Object pop() throws UresException {...}
}

```

javadoc meghívása

- A különféle hozzáférési kategóriák (public, protected, package, private) szerint lehet különböző részletességű dokumentációt készíteni
- Lehet saját formátumot definiálni (doclet)
- És van egy csomó kapcsoló (-classpath, -sourcepath, -d, stb.)

jar - Java ARchieve

- Lényegében zip formátum, kis extrákkal
- A jar program használata hasonlít a tar parancsra
- Archívum készítése:

```
jar cvf classes.jar Foo.class Bar.class
```

Más fejlesztőeszközök

- Sok cég csinált már Java fordítót és virtuális gépet
- Például a böngészőprogramokban is saját szokott lenni
 - Helyette használhatunk plug-in virtuális gépet
- Fejlesztő eszközökben is szokott saját lenni
- Vannak hatékonyabbak is, mint a Sun cuccok

Java tutorial

Copyright © 2000-2001, Kozsik Tamás