Java és web programozás

Kovács Kristóf

Budapesti Műszaki Egyetem

2015. 02. 11.

Tárgyismertető

Elérhetőségek:

- Kovács Kristóf: kkovacs@math.bme.hu

A tárgy honlapja:

- http://wiki.math.bme.hu/view/WebProg-2014

Követelmények:

Az alábbiak közül 2-t kell teljesíteni:

- Gyakorlaton való jelenlét
- Év végi ZH
- Nagyházi

Jegy a jelenléten kívüli követelmény(ek) alapján.

Ami lesz

- Java
- Legalapabb java csomagjainak ismertetése
- Java alapú 2D grafikus alkalmazások létrehozása (értsd: játék)

Ami lesz

- Java
- Legalapabb java csomagjainak ismertetése
- Java alapú 2D grafikus alkalmazások létrehozása (értsd: játék)

Órán kívül megtanulható:

- Java alapú szerver készítése
- Adatbázis kezelés sqlite3-al

Az utóbbiakhoz lesz anyag, de nem képezik részét a gyakorlatoknak (és így a ZH-nak sem). Viszont nagyházinak elfogadok webprogramot is, hisz a tárgy neve is ez.

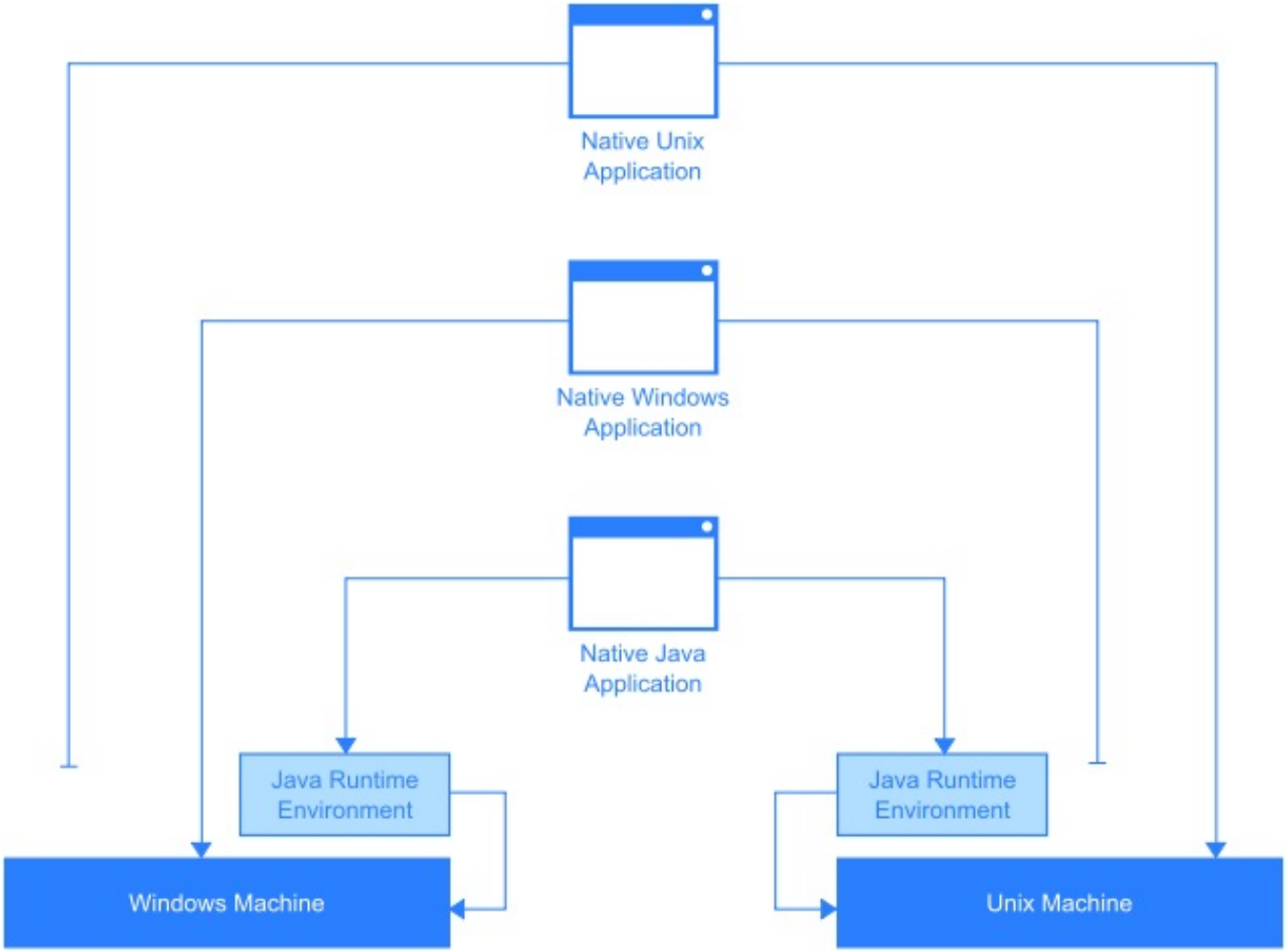
A java szerkezete

- Nem gépi kódra fordul, mint a C, C++.
- Interpretált nyelv, mint a python, azaz az utasításokat valós időben értelmezi.

A java szerkezete

- Nem gépi kódra fordul, mint a C, C++.
- Interpretált nyelv, mint a python, azaz az utasításokat valós időben értelmezi.
- Viszont alkalmaz egy valós időben fordítást, ami annyit tesz, hogy bizonyos gyakran használt kódrészeket mégis gépi kódra fordít.
- Ezáltal a sebessége, mostmár csupán 44%-al van lemaradva a C++-hoz képest.

- Cserébe, ugyanaz a kód, bármely platformon futtatható, újrafordítás nélkül.



Változók, függvények

Változó deklarálás

```
int valtozoNev;
```

Változó definiálás

```
float valtozoNev = 16.4;
```

Függvény definiálás

```
public static int duplaz(int szam) {
    return szam * 2;
}
```

- A *public* és *static* kulcsszavakról lesz szó a későbbiekben, most csak tudjuk, hogy oda kell írni őket.

Elágazások

Elágazas

```
if (feltetel) {
    ...
}
```

Többszörös elágazás

```
if (feltetel1) {
    ...
} else if (feltetel2) {
    ...
...
} else {
    ...
}
```

For ciklus

```
int i;
for (i = 0; i < 10; i++) {
    ...
}
```

For ciklus tömörebben

```
for (int i = 0; i < 10; i++) {
    ...
}
```

For ciklus, mint pythonban (*tomb* egy *int* tömb)

```
for (int elem : tomb) {
    ...
}
```

While ciklus

```
while (feltetel) {
    ...
}
```

While ciklus

```
while (feltetel) {
    ...
}
```

- Amint látjuk eddig nagyban hasonlít a többek által tanult C szintaktikára. Ez nem véletlen. A java szintaktikáját C++ alapján alakították ki.
- Viszont sok különbség is van, de ezek később kerülnek majd elő.

While ciklus

```
while (feltetel) {
    ...
}
```

- Amint látjuk eddig nagyban hasonlít a többek által tanult C szintaktikára. Ez nem véletlen. A java szintaktikáját C++ alapján alakították ki.
- Viszont sok különbség is van, de ezek később kerülnek majd elő.
- Itt még megemlíteném, hogy a java ráerőszakolja a programozóra, hogy mindent osztályokkal oldjon meg. Hasonlóan mint a python kötelező formázása, ez se véletlen.

Hello World!

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

Hello World!

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

- Elemezzük ezt a pár sor java kódot.

Hello World!

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

- A *class* kulcsszóval tudunk létrehozni új osztályt.
- Amint már említettem, javaban mindent osztályokkal kell megoldani, még egy egyszerű *Hello World!* programot se tudunk megúszni nélkülük.

Hello World!

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

- A *class* kulcsszóval tudunk létrehozni új osztályt.
- Amint már említettem, javaban mindent osztályokkal kell megoldani, még egy egyszerű *Hello World!* programot se tudunk megúszni nélkülük.
- Osztályokról már sokan tanultatok, de nem mindannyian. Az elejétől fogunk mindent áttekinteni velük kapcsolatban.
- Ezen az órán még nem foglalkozunk velük, csak 1 osztályt írunk, benne a *main* függvénnyel.

Hello World!

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

- A *main* függvény itt is a program kiindulási pozícióját jelzi.
- De a C/C++-al ellentétben itt nem tilos egy projekten belül több *main*t létrehozni.

Hello World!

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

- A *main* függvény itt is a program kiindulási pozícióját jelzi.
- De a C/C++-al ellentétben itt nem tilos egy projekten belül több *main*t létrehozni.
- Neki is kellenek a *public static* kulcsszók, ezekről majd később.
- Továbbá bemenete egy *String* tömb. Amiben megkapja a program argumentumait.
- *String*ekről lesz a későbbiekben szó, gyakorlaton is. Nagyon hasonlóan működnek mint a python stringjei.

Hello World!

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

- Végül ez az első ijesztőnek tűnő kiiratás.
- Nem kell tőle annyira megijedni, amint *Eclipse*-en dolgozunk már meglátjuk milyen könnyedén lehet használni.
- Későbbiekben lesz szó arról mi a *System*. Azon belül az *out* az outputot jelöli.

Hello World!

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

- Végül ez az első ijesztőnek tűnő kiiratás.
- Nem kell tőle annyira megijedni, amint *Eclipse*-en dolgozunk már meglátjuk milyen könnyedén lehet használni.
- Későbbiekben lesz szó arról mi a *System*. Azon belül az *out* az outputot jelöli.
- A *println* annyiban különbözik a *print*től, hogy automatikusan tesz sortörést a kiírt szöveg után.
- Hasonlóan a pythonhoz, a *print* és tarsai javaban is ki tudnak iratni minden beépített típust nem csak a *String*et.

Alapvetõ programozási „tételek”

Legyen

- t egy tömb (pythonosoknak: lista)
- i, s egész változók

Alapvetõ programozási „tételek”

Legyen

- t egy tömb (pythonosoknak: lista)
- i, s egész változók

Szummázás:

```
s = 0;
for(i = 0; i < t.length; i++) {
    s += t[i]
}
```

Alapvetõ programozási „tételek”

Legyen

- t egy tömb (pythonosoknak: lista)
- i, s egész változók

Szummázás:

```
s = 0;
for(i = 0; i < t.length; i++) {
    s += t[i]
}
```

Kiválasztás:

```
;
for(i = 0; i < t.length; i++) {
    if(t[i] % 2 == 0) {        // A párosakat választjuk ki
        System.out.println(t[i]);    // Kiírja t[i]-t
    }
}
```

Számlálás:

```
s = 0;
for(i = 0; i < t.length; i++) {
    if(100 % t[i] == 0) {       // A 100 osztója-e t[i]
        s++;                    // Vagy s += 1 vagy s = s + 1
    }
}
```