Java és web programozás

Kovács Kristóf, Rimay Zoé

Budapesti Műszaki Egyetem

2013. szeptember 11.

Jó hír

Jó hír

Java lesz!

Tárgyismertető

- Elérhetőségek:
  - Kovács Kristóf: kkovacs@math.bme.hu
  - Rimay Zoé: zrimay@math.bme.hu,
- Követelmények:
  - Év végi beadandó (facebook): 60
    - Minimum 24 pontot meg kell szerezni.
  - Hetente kisZH **vagy** házifeladat: 40
    - 4 pontot érnek, de hetente csak az számít amelyik jobban sikerült
    - Legalább 10 héten lesz mindkettő
    - Nincs minimumkövetelmény külön-külön egyikből se, de összességében meg kell lennie a 16 pontnak.
- Értékelés:

  5: 85 - 100
  4: 70 - 84
  3: 55 - 69
  2: 40 - 54
  1: 0 - 39

Ami lesz

- Java
- Java alapú szerver készítése
- HTML bemenetek használata
- Adatbázis kezelés sqlite3-al

Ami lesz

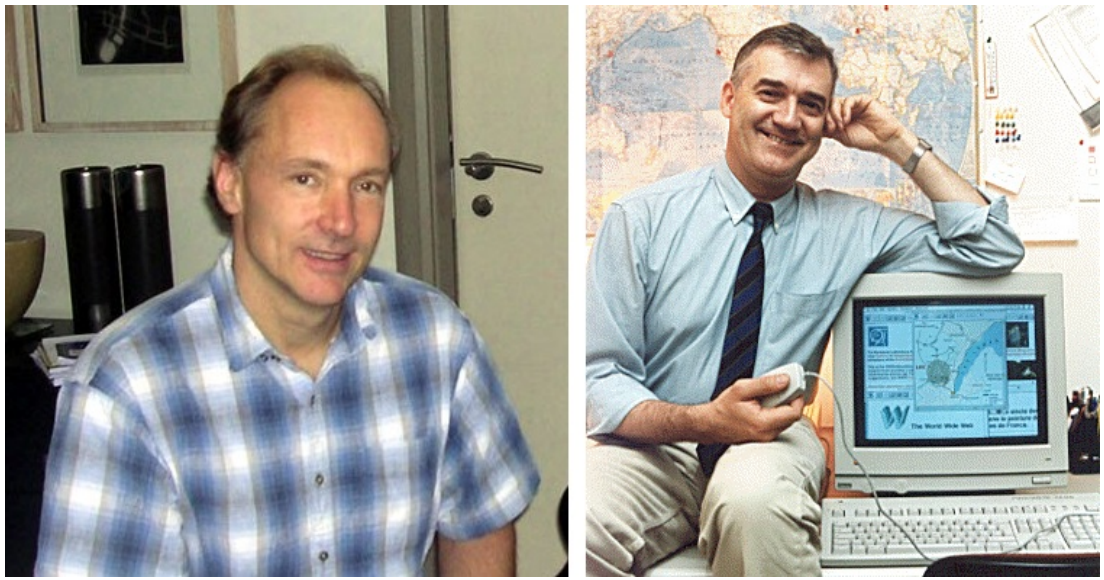
- Java
- Java alapú szerver készítése
- HTML bemenetek használata
- Adatbázis kezelés sqlite3-al

Ha marad idő:

- Reguláris kifejezések
- HTML-fa bejárás
- Javascript

World Wide Web

Egy kis történeti háttér: (nem kérdezzük vissza)



Sir Tim Berners-Lee (bal) és Robert Cailiau (jobb)

Sir Tim Berners-Lee és Robert Cailliau 1990-ben a CERN-ben tervezték meg a WWW-t, hogy egy olyan hálózatot hozzanak létre amelyben bárki hozzáférhet adott információkhoz.

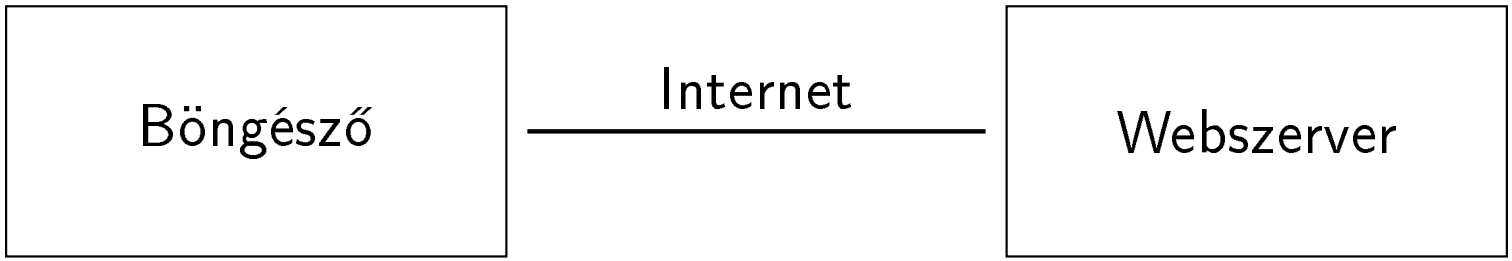
World Wide Web

Még 1990 végén elkészítették az első böngészőt, majd 1991-ben lett ténylegesen publikus a WWW az interneten.

Hogy ez effektíven működhessen bevezették az URL-t (Uniform Resource Locator), a HTML-t (HyperText Markup Language) és a HTTP-t (HyperText Transfer Protocol).

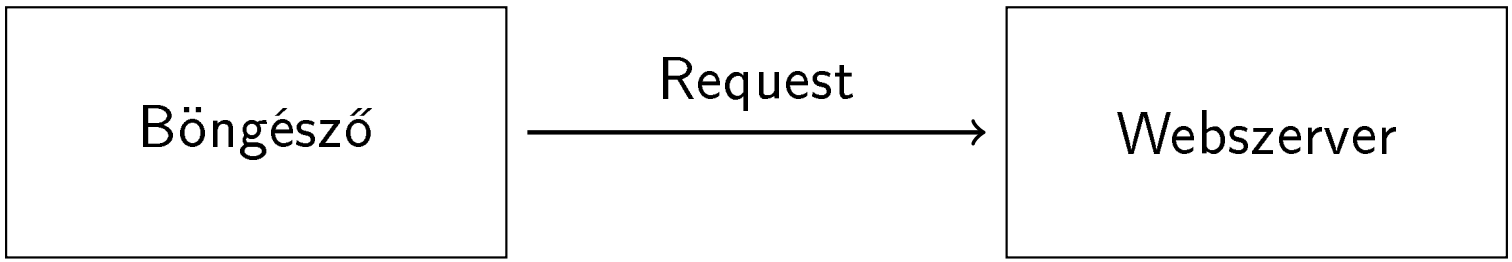
A CERN 1993-ban bejelentette, hogy a Web ingyenes lesz mindenkinek, majd később ebben az évben elkészült az első igazán elterjedt böngésző, a Mosaic.

A Web működése nagy vonalakban



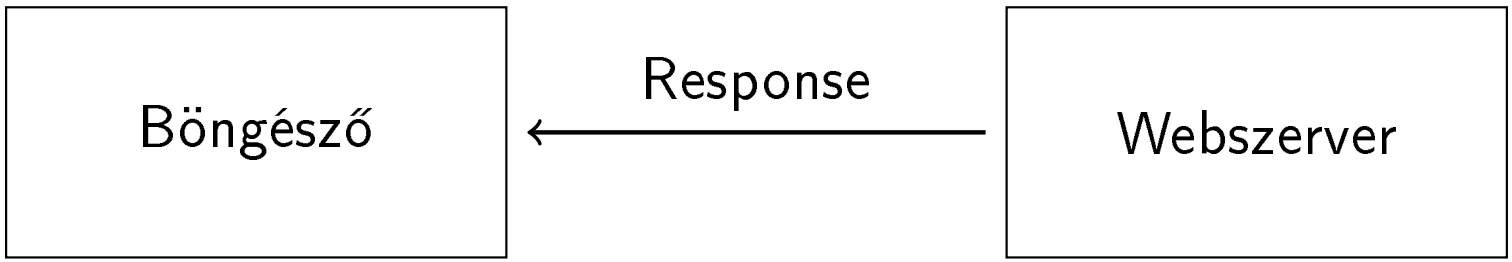
A böngészőt összeköti az internet egy webszerverrel.

A Web működése nagy vonalakban



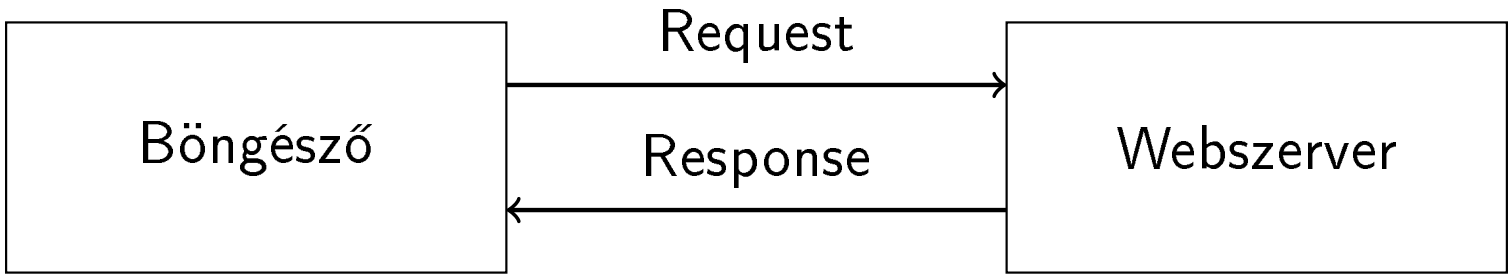
Küld a böngésző egy kérést (request) a webszervernek, hogy melyik adott oldalt szeretné lekérdezni, meg még küld egyéb dolgokat is mellette, erről egy későbbi előadáson lesz szó.

A Web működése nagy vonalakban



Kap erre egy választ (response), ami tartalmazza a HTML-t amit majd a böngésző megjelenít a felhasználónak.

A Web működése nagy vonalakban



Így folytatódik a böngészés, akárhányszor új oldalra akar lépni a felhasználó a böngésző küld egy kérést, majd erre válaszol a webszerver.


Borneo
Sumatra
Indonesia
Java Sea
Java
Bali

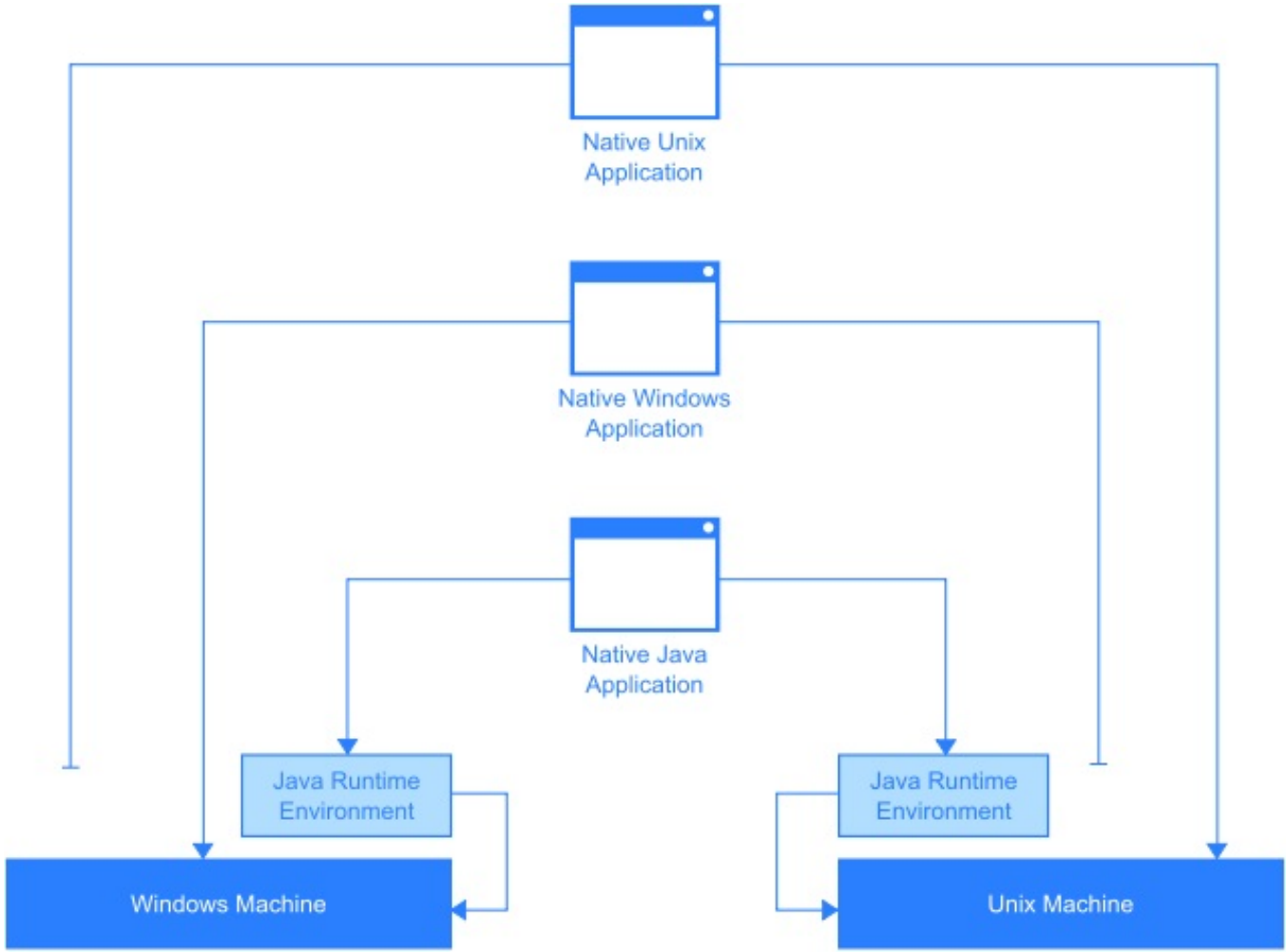
A java szerkezete

- 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.



Eddig tanult dolgok javaban

- Változó deklarálás

```
int valtozoNev;
```

- Változó definiálása

```
float valtozoNev = 16.4;
```

- Függvény definiálásá

```
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.

- 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 már 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 már 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!");
    }
}
```

- A maradék dián ezt a pár sort magyarázom el.

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.
- Osztályok már voltak pythonban. Ennek ellenére az elejétől fogunk mindent áttekinteni velük kapcsolatban, mert a működésük alig hasonlít.
- 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 az kiírt szöveg után.
- Hasonlóan a pythonhoz, a *print* és tarsai javaban is ki tudnak iratni minden beépített típust nem csak a *String*et.