

Java

1. Óra

Kiss Gábor

Objektum, osztály

- **Objektum:** Információkat tárol, és kérésre feladatokat hajt végre. Logikailag összetartozó adatok és rajtuk dolgozó algoritmusok összessége:
 - adatok
 - metódusok
- Az objektumot üzenetek (kérelmek) által lehet megkérni a feladatok elvégzésére. (Ez egy metódus végrehajtását jelenti.)
- **Osztály (class):** Objektumtípus, amely alapján példányokat, azaz objektumokat hozhatunk létre.

OO nyelvek

- Simula-67
- Smalltalk (70' eleje, XEROX)
- Hibrid nyelvek: C++, Turbo Pascal
- Java: '91-96, SUN MicroSystem

A Java főbb jellemzői

- Hordozható, platformfüggetlen
 - forráskód és bájtkód szinten
- Tisztán objektumorientált
 - csak osztályok, illetve objektumok léteznek
- Szabványos osztálykönyvtár
- Szintaktikája a C++ -on alapszik.
- Rohamos fejlődés kulcsa a JVM beépítése a Web böngészőkbe.
- Fejleszthető:
 - Alkalmazás (application)
 - Applet
 - Servlet
 - JSP (Java Server Pages)

JDK (Java Development Kit)

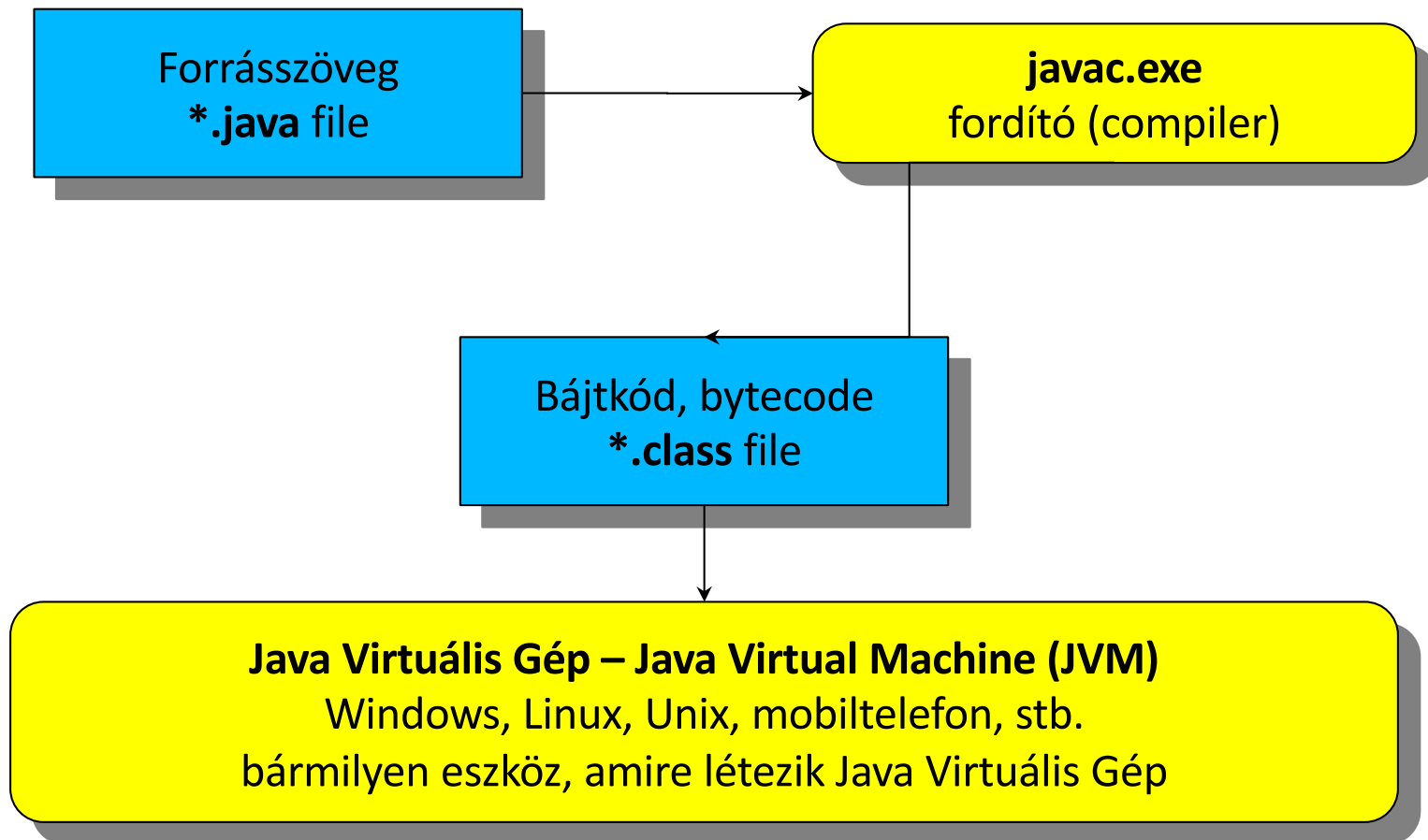
- Fejlesztői és futtató környezet, részei:
 - API (Application Programming Interface) – osztály-könyvtár (a Java egyik ereje a rengeteg kész osztály)
 - Fordító
 - Értelmező
 - Appletnéző
 - Help, stb
- Csomag (package): logikailag összetartozó osztályok. Csomag struktúra – könyvtár struktúra.

A Java család

- JavaScript
 - HTML-be ágyazott forrásszöveg
 - korlátozott eszközkészlet
 - ezzel nem foglalkozunk ebben a tárgyban
- Java Applet
 - Böngészőben futó bináris program, felhasználhatja a böngésző program által nyújtott szolgáltatásokat
 - HTML file-ból indított bájtkód
- Java Alkalmazás
 - önállóan fut a Java Virtuális Gép alatt

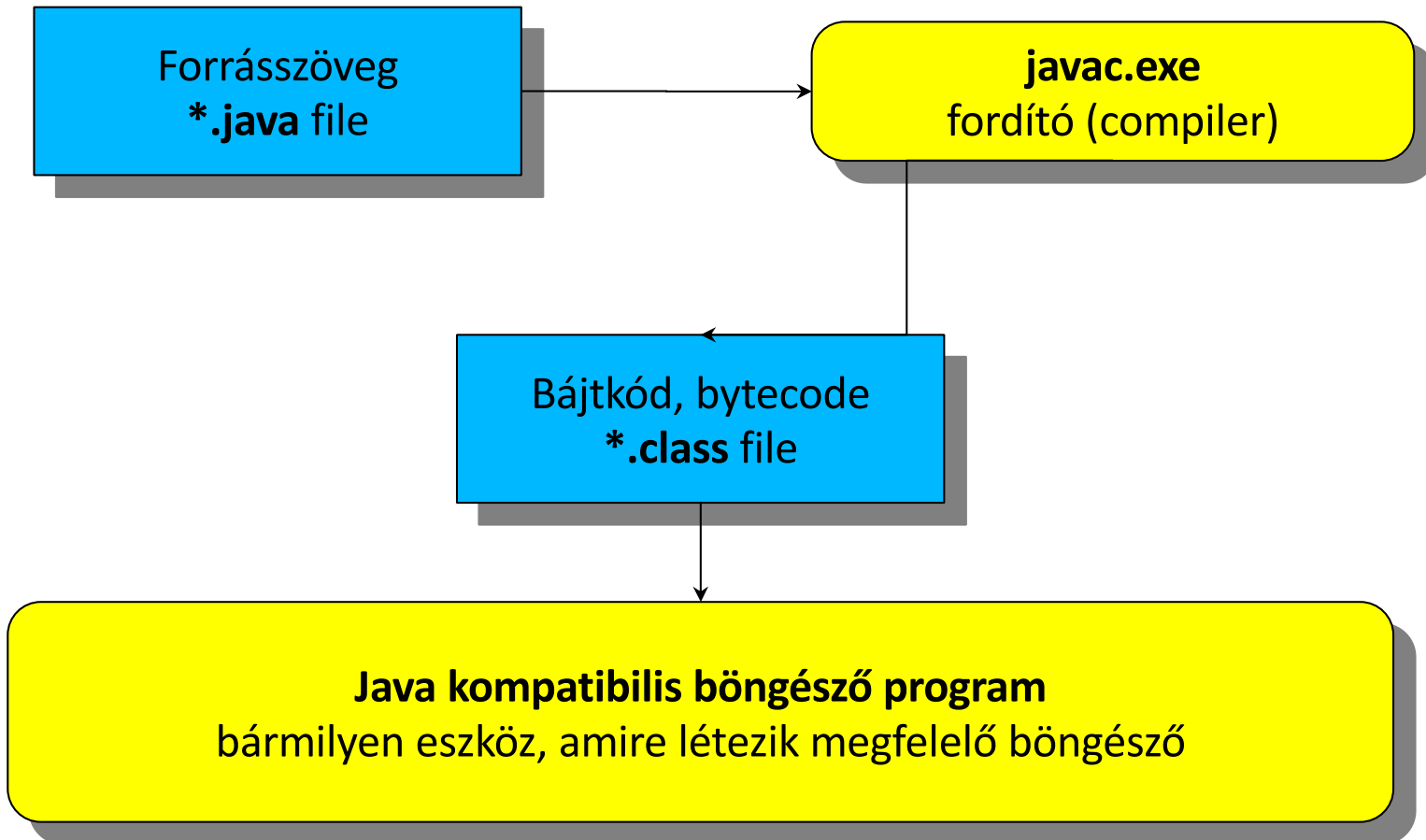
A Java programozási modell

Java alkalmazás

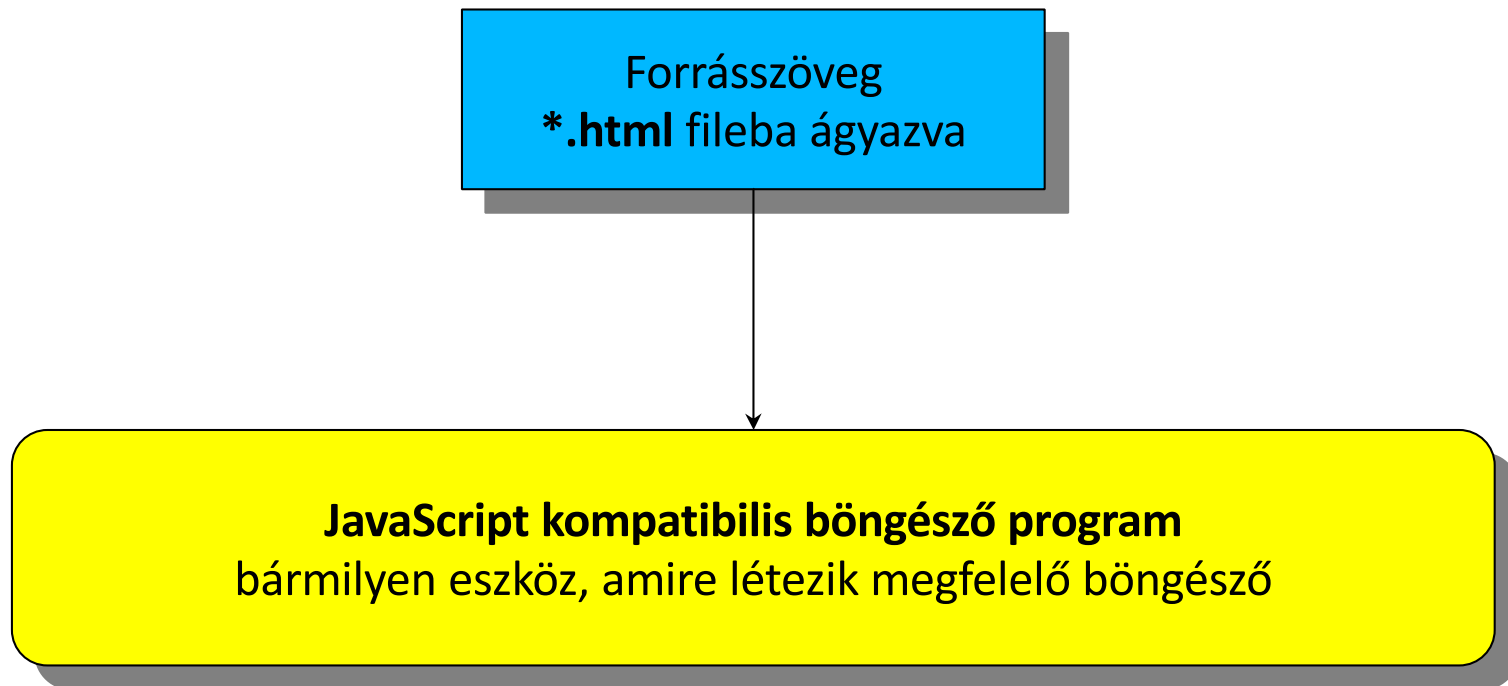


A Java programozási modell

Java kisalkalmazás (Applet)



A Java programozási modell JavaScript



Konvenció: A java osztályok neve nagy betűvel kezdődik!

Az első alkalmazás



- Forrásszöveg (HelloVilag.java):

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

- Fordítás:

```
javac HelloVilag.java
```

- Futtatás:

```
java HelloVilag
```

Notepad++-ba gépeljük be az alábbi programot:

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

Majd mentjük el Hello.java néven (a fájl nevének egyeznie kell a publikus osztály nevével (kis-nagybetű egyezőségre figyelni!))

A program futtatásához telepíteni kell a Java fejlesztői csomagot (JDK: Java Development Kit), mely letölthető a www.javasoft.com címről

A program lefordítása DOS ablakban a `javac Hello.java` utasítással tehető meg. Az elérési útvonalat meg kell adni (pl. `c:\jdk13\bin`), hogy a programot elérhessük.

A lefordított program egy `Hello.class` állományban jelenik meg. Nézzük meg a tartalmát (bytecode).

A futtatáshoz ezt a fájlt kell a `java Hello` utasítással lefuttatni.

Akkor most még egyszer Eclipse-ben 😊

- Először hozzunk létre egy új Java Project-et (File/New/Java Project) és adjunk neki nevet.
- Nézzük meg a kialakított filerendszert a létrehozás helyén a project nevével megegyező mappában. Az src mappába menti majd az ebben a projectben létrehozott java kiterjesztésű fájlokat és a bin könyvtárba a már lefordított class kiterjesztésű fájlokat. Ezek a fájlok ugyanúgy használhatók a már korábban megnézett DOS ablakból is (útvonalra figyelni).
- Az Eclipse bal oldalán legyen kiválasztva a most létrehozott projekt (ez majd folyamatosan bővülni fog (remélhetőleg otthon is 😊))
- Hozzuk létre a Hello nevű osztályt (File/New/Class) és az alsó választó boxok közül legyen kiválasztva a public static void main(String[] args)

Akkor most még egyszer Eclipse-ben 😊

- Ezután létrehozza az osztályt és a forráskód egy részét (ami mindenképpen kell) kiegészítő megjegyzésekkel létrehozza.
- Kezdjük el begépelni a main metódusba a `System.out.println` sorunk elejét.

```
System.out.println("Hello!");
```

- Nyomjuk meg a CTRL+SPACE billentyűkombinációt. Gyorsabb a fejlesztés sok esetben, ha innen választjuk ki a számunka fontos osztályokat, metódusokat és a szintaktikai hibák ejtésének valószínűsége is csökken.
- Zárjuk az utasítást ;-vel.
- Futtassuk le a Run/Run, vagy F11, vagy megfelelő ikon kiválasztásával. Az alsó ablakban elvileg kiírja a kért szöveget. Nézzük meg az src és bin mappák tartalmát újra.

Fontos

A minden publikus osztályt külön – az osztály nevének megfelelő java kiterjesztésű – fájlba kell írni (Eclipse-ben külön class-ként kell létrehozni).

Van egy kitüntetett osztály aminek van egy static main metódusa, ez lesz a főprogram!

A main metódus szignatúrája a következő KELL legyen:
`public static void main(String[] args)`

A statikus metódusok futtathatók példányoktól függetlenül.
Futtatásnál a VM –nek megmondjuk, melyik a „főosztály”, aminek a main metódusát meg kell hívnia. Ez a program indulási pontja.

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

- A `HelloVilag` nevű osztály publikus, más csomagokból is látni lehet.
- A `main` metódus a program belépési pontja.
 - Publikus
 - A `static` kulcsszó azt jelenti, hogy ez osztálymetódus
 - `void`: nincs visszatérési értéke, azaz eljárás
 - `String[] args`: a parancssor paraméterek
- A `System` egy osztály, amelyben található egy **out** objektum, melynek a `println` egy metódusa.

A Java nyelv alapjai 1.

- Megjegyzés (comment)
 - egysoros megjegyzés:
`// jeltől a sor végéig`
 - hosszabb megjegyzés:
`/*
...
*/`
 - dokumentációs megjegyzés (a **javadoc** program használja):
`/**
...
*/`

A Java nyelv alapjai 2.

- Karakterkészlet
 - ASCII helyett Unicode karakterkészlet
 - 8 bit helyett 16 bit
 - minden Unicode karakter használható a teljes forrásszövegben, akár az azonosítókban is, ha a környezet támogatja
 - óvatosan használni!!! (pl. ékezet)

A Java nyelv alapjai 3.

- Változók (itt azonosító) elnevezése
 - Betűvel kezdődik – betűvel vagy számmal folytatódik
 - Betűnek számít a \$ és az _ jel valamint az ékezetes magánhangzók is.
 - Case-sensitive, azaz különböznek a kis- és nagybetűk
 - A hossza nincs korlátozva
 - A nyelv kulcsszavai nem lehetnek változók nevei

A Java nyelv alapjai 4.

Egyszerű típusok

egyszerű típus	osztály	leírás
boolean	Boolean	logikai típus (true, false)
char	Character	16 bites Unicode karakter
byte	Integer	8 bites egész szám (-128 - +127)
short	Integer	16 bit, (-32 768 - +32 767)
int	Integer	32 bit (-2 147 483 648 - +2 147 483 647)
long	Long	64 bit (-9 223 372 036 854 775 808 - +...)
float	Float	32 bit (-3,40292347E+38 - +3,40292347E+38)
double	Double	64 bit (-1,79769313486231570E+308 - +...)

- a szabvány rögzíti az ábrázolási módot és a méretet, nem az implementáció
- hagyományos típusként és osztályként is használhatók
- osztályként a MIN_VALUE és a MAX_VALUE konstansok definiálják a határokat

Változó/azonosító deklarálás

- Deklaráló utasítás
- Adható kezdőérték is (inicializálás).
- Pl:

```
int a, b;
```

```
double x, y = 2.4;
```

A char is numerikus típus

```
char c;  
int a;  
c=65;  
System.out.println(c);  
c='A'+1;  
System.out.println(c);  
a=c+1;  
System.out.println(a);
```

Egészítsük ki a meglévő main metódusunkat ezekkel az utasításokkal és futtassuk le.