# 1. gyakorlat

Kezdés: 8:30

Vége: 10:00

Netbeans: <http://sourceforge.net/projects/nbportable/> vagy egyszerűen <http://netbeans.org/> (Iskolai gépekre nem fér fel, átlépjük vele a kvótát. Szerencse, hogy én laptoppal járok.)

Gyakorlatvezető: Lengyel Zsolt ([lengyel.zsolt@informatics.elte.hu](mailto:lengyel.zsolt@informatics.elte.hu))

Honlap: <http://lengyel.web.elte.hu/> (<http://lengyel.web.elte.hu/index.php?q=node/3>, <http://lengyel.web.elte.hu/index.php?q=node/60>)

Objektum elvű, összetettebb rendszerek tervezése és implementálása.

Konzultáció: Gyakorlat és előadás között. 10:00-10:45

Gyakorlat látogatása kötelező.

## Követelményrendszer

Csak gyakorlati jegyet kell szerezni. Vizsga nincs.

Gyakorlati jegyért:

* 2 beadandó
* 2 ZH
* Pluszminusz

### ZH

Időpontok: A honlapon.

Helyszín: Itt, vagy a lovardában írjuk, a gyakorlat idejében.

Első ZH: Őszi szünet előtti utolsó órán. (Hétfői csoporttal közösen, mert nekik szünet lenne.)

Második ZH utolsó gyakorlaton 8:00-11:00

Javító és pót ZH: Ha az egyik ZH nem sikerül. Ha mindkét ZH sikertelen, nincs gyakjegy. Minden segédeszköz használható.

### Beadandó

Dokumentáció és program beküldése e-mailben, határidőre. Késni nem lehet. Önálló munka legyen.

Összetettebb feladat lesz.

Így félév végére lesz 4 db jegyünk.

### Pluszminusz

vagy jár érte, nincs.

Óra végén lesz kiadva, honlapon is fent lesz. Egy hetünk van rá mindig.

Ha a pluszminusz jól áll, legalább nullára, akkor a fenti 4 jegy alapján kapunk osztályzatot. Ha a pluszminusz nagyon pozitívban van, a jegy felfelé kerekül.

## Félév első fele: Konzolos alkalmazások

Középkori csatában az egységeket objektumként kezeljük. Ezeknek lesznek funkcióik, tulajdonságaik, stb.

Hogyan tudjuk közölni a felhasználóval a program futása során történő eseményeket? Kiírjuk a konzolra az információkat.

Később grafikus alkalmazásokat készítünk. (A félév második felében)

# JAVA

Operációs rendszer mindegy.

A modellező és tervező eszközök, amiket használni fogunk is JAVA-ban lesznek, azoknak is mindegy az OS.

Legalább 1.5-öst érdemes használni. (Nekem 1.7)

## Runtime Environment (RTE)

Java futtatási környezet.

Miért van erre szükség? Hogy ne kelljen újrafordítani Linux és Windows között.

A JAR bájtkódot változtatás nélkül tudjuk futtatni mindenhol.

Ehhez kell a virtuális gép. Futásidejű értelmezés történik, nem fordításidejű.

## Java compiler

Első próbálkozásra nem működik parancssorból. Be kell állítani.

Itt találjuk:   
(JDK=JAVA Development Kit)

Példa program fordítása:

Vagy:

Ha nincs hiba, nem ír ki semmit.

Fordítás eredménye bájtkód:

Futtatása:

Ha lefut, lesz látható a konzolon. Ha hiba történik a futás során, kiírja. Lehet például stack trace.

## Hello World

public class Hello**{**

public static void main **(**String**[]** args**){**

System**.**out**.**println**(**"Hello World!"**);**

**}**

**}**

Sorvégi pontosvessző kirakása nem fontos.

Nyitó kapcsos zárójelet mindig a sor végére rakjuk, nem új sorba.

Osztály nevét mindig nagybetűvel kezdjük.

A fájl neve legyen ugyan az, mint az osztály neve.

Egy forrásfájlba mindig csak egy osztályt rakunk. (Kivéve a beágyazott osztályok.)

Csomagokra is hasonló lesz igaz, ezt majd később látjuk. (Ugyan olyan mapparendszerbe rakjuk a csomagban levő osztályokat, mint ahogy az a forráskódban van felépítve.)

## Fejlesztői környezet: NetBeans

Van több száz ilyen program, mi ezt fogjuk használni.

Projekt neve legyen mindig leíró, érdemes nagybetűvel kezdeni.

Csomag neve ugyan az lesz, mint az osztály neve, csak kisbetűs.

Különbség: A forrásfájl tele van kommentekkel. (Dokumentációs komment) Van előtte package deklaráció.

Nyilvános osztálydeklaráció.

### Importálás

import java.util.Scanner;

### Equals függvény

Összehasonlít.

(Az operátor csak a memóriacímeket hasonlítaná össze. Ami jelen esetben nem gond, szerintem.)

"str"**.**equals**(**"str"**)**

object**.**equals**(**object**)**

### Primitív típusok

int**,** byte**,** long**,** float**,** double**,** boolean**,** char**,** void

### Heap

Dinamikus változók helye, **new** kulcsszóval hozzuk létre ezeket, csak referenciát kapunk rá.

Scanner sc **=** **new** Scanner**(**System**.**in**);**

s **=** sc**.**nextLine**();**

### Egyforma stringek

Csak egy példányban vannak tárolva.

### Parancssori paraméterek beállítása

Jobb egérgomb a projekten, tulajdonságok legalul (Properties), majd futtatás (Run), végül paraméterek (Arguments).

## HF

Prímtényezős felbontás. (Konzultáció végén bemutattam, pluszt kaptam rá.)