

Prog2, Java befejezés

Magasszintű programozási nyelvek 2 mérnök informatikus BSc előadás

Dr. Bátfai Norbert

egyetemi adjunktus

<http://www.inf.unideb.hu/~nbatfai/>

Debreceni Egyetem, Informatikai Kar,
Információ Technológia Tanszék

batfai.norbert@inf.unideb.hu

Skype: batfai.norbert

Prog2_3.ppt, v.: 0.0.5, 2011. 12. 11.

<http://www.inf.unideb.hu/~nbatfai/>

<http://nehogy.fw.hu/>

Az óra blogja: <http://progpater.blog.hu/>

A Nokia Ovi store-ban is elérhető: <http://store.ovi.com/content/100794>

Felhasználási engedély

Bátfai Norbert

Debreceni Egyetem, Informatikai Kar, Információ Technológia Tanszék
<nbatfai@inf.unideb.hu, nbatfai gmail com>

Copyright © 2011 Bátfai Norbert

E közlemény felhatalmazást ad önnek jelen dokumentum sokszorosítására, terjesztésére és/vagy módosítására a Szabad Szoftver Alapítvány által kiadott GNU Szabad Dokumentációs Licenc 1.2-es, vagy bármely azt követő verziójának feltételei alapján. Nem változtatható szakaszok: A szerzőről.

Címlap szövegek: Programozó Páternoszter, Bátfai Norbert, Gép melletti fogyasztásra.

Hátlap szövegek: GNU Jávácska, belépés a gépek mesés birodalmába.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with the Invariant Sections being: A szerzőről, with the Front-Cover Texts being: Programozó Páternoszter, Bátfai Norbert, Gép melletti fogyasztásra, and with the Back-Cover Texts being: GNU Jávácska, belépés a gépek mesés birodalmába.

Kapcsoldó videók, videómagyarázatok és blogok

Az előadás és a
labor posztja ez(ek)

1) Embernek néznek ki, izzadnak, bűdös a leheletük*:

[http://progpater.blog.hu/2011/10/01/embernek neznek ki izzadnak budos
a leheletuk](http://progpater.blog.hu/2011/10/01/embernek_neznek_ki_izzadnak_budos_a_leheletuk)

*: Cameron, James, T E R M I N A T O R, <http://www.imdb.com/title/tt0088247/>,
<http://www.imsdb.com/scripts/Terminator.html> , 1984.

Célok és tartalom

Előadás

- a) A Java kivételkezelése,
- b) Párhuzamosság Javában, Swing és eseménykezelés
- c) A RoboCup 2D szimulációs liga folyt., Java fejlesztés: Apache Maven folyt.
- d) Java és kriptográfia
- e) Java kiadások szerint áttekintett nyelvi és API elemek

Labor (2 alkalomra tervezve)

- a) a Javát tanítók példáinak felélesztése
genomi pontmátrix, Mandelbrotosok, sejtautomatásak
- b) Atan alapú robotfoci:
Aranycsapat FC, Marvellous Magyars FC, Mighty Magyars FC, Golden Team FC

Laborkártyák

- a) Kivételes kártyák, polimorf kártyák, metrikus kártyák

Otthoni opcionális feladat

- a) RSA, Bouncy Castle
- b) Hibásan implementált RSA törése
- c) Saját Atan 1.0 alapú RCSS csapat fejlesztése
- d) Sonar, XRadar kísérletezés a megfelelő Maven bővítménnyel

Minimális gyakorlati cél

- a) A hallgató ki tudja számítani az alapvető OO metrikákat a projektjére.
- b) Egy példán keresztül be tudja mutatni a SwingWorker osztály használatát.
- c) Vázoljon egy olyan szituációt, amikor a program felülete „lefagy”.
- d) (Például saját focicsapatának) tudjon SVN tárolót létrehozni, azt használni (pl., co, commit stb.)

Minimális elméleti cél

- 1) Java nyelv kapcsán: kivételkezelés, párhuzamosság.
- 2) Ismerje a SwingWorker osztályt, az API-ját (<http://download.oracle.com/javase/7/docs/api/index.html?javax/swing/SwingWorker.html>) tudja olvasni.
- 3) Adott esetre tudjon saját kivételosztályt tervezni, írni.

Ismétlés: vissza a kivételkezeléshez

p1/7. prezi

```
try {  
    Osztaly1 peldany1;  
    Osztaly2 peldany2 = new Osztaly2();  
    ...  
    throw "hiba";  
    ...  
    delete peldany2;  
} catch (char * hiba) {  
    ...  
}
```

Mi a helyzet a **peldany?**-ekkel?

Ismétlés: vissza a kivételkezeléshez

p1/7. prezi

```
try {  
    ...  
    throw KivetelOsztaly();  
    ...  
} catch (KivetelOsztaly vagy annak őse) {  
    ...  
    ha nem tudja kezelni, tovább dobhatja: throw  
    ...  
} catch (...) {  
  
}
```


A Java kivételkezelés bevezetése

```
/*
 * Labirintus.java
 *
 * DIGIT 2005, Javat tanítok
 * Bátfai Norbert, nbatfai@inf.unideb.hu
 *
 */
package javattanitok.labirintus;
/**
 * A labirintust leíró osztály.
 *
 * @author Bátfai Norbert, nbatfai@inf.unideb.hu
 * @version 0.0.1
 */
public class Labirintus {
    /** A labirintus szélessége. */
    protected int szélesség;
    /** A labirintus magassága. */
    protected int magasság;
    /** A labirintus szerkezete: hol van fal, hol járat. */
    protected boolean[][] szerkezet;
}
```

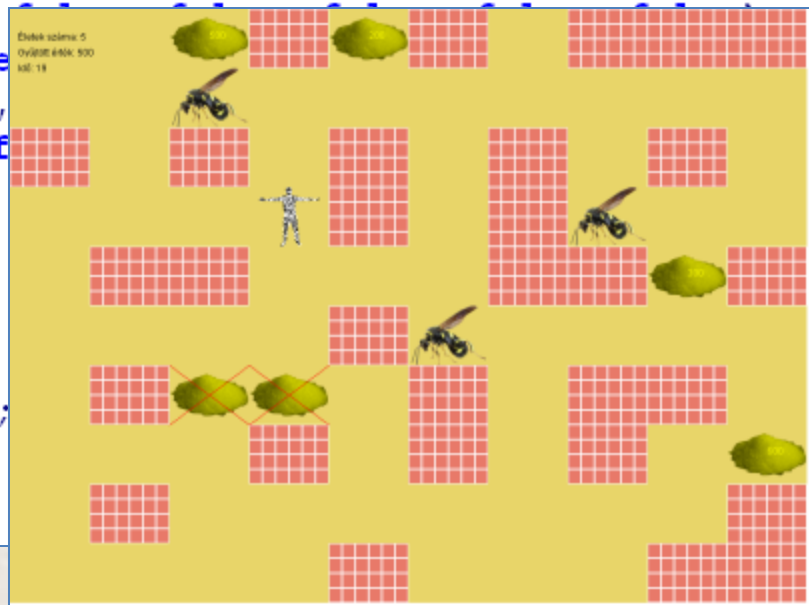
Kivételkezelés bevezetése

```
/**
 * Létrehoz egy megadott szerkezetű
 * {@code Labirintus} objektumot.
 */
public Labirintus() {
    szerkezet = new boolean[][]{

        {false, false, false, true, false, true, false, true, true, true },
        {false, false, false, false, false, false, false, false, false, false},
        {true, false, true, false, true, false, true, false, true, false},
        {false, false, false, false, true, false, true, false, false, false},
        {false, true, true, false, false, false, true, true, false, true },
        {false, false, false, false, true, false, false, false, false, false},
        {false, true, false, false, false, false, false, false, false, false},
        {false, false, false, true, false, false, false, false, false, false},
        {false, true, false, false, false, false, false, false, false, false},
        {false, false, false, false, true, false, false, false, true, false},

    };

    magasság = szerkezet.length;
    szélesség = szerkezet[0].length;
}
}
```



Kivételkezelés bevezetése

```
/**
 * Létrehoz egy paraméterben kapott szerkezetű <code>Labirintus</code>
 * objektumot.
 *
 * @param kincsekSzama a kincsek száma a labirintusban.
 * @param szörnyekSzama a szörnyek száma a labirintusban.
 * @exception RosszLabirintusKivétel ha a labirintust definiáló tömb
 * nincs elkészítve.
 */
public Labirintus(boolean[][] szerkezet, int kincsekSzama, int szörnyekSzama)
throws RosszLabirintusKivétel {

    if(szerkezet == null)
        throw new RosszLabirintusKivétel("A labirintust definiáló tömb nincs elkészítve");

    this.szerkezet = szerkezet;

    magasság = szerkezet.length;
    szélesség = szerkezet[0].length;

    kincsekSzörnyek(kincsekSzama, szörnyekSzama);
}
```

```

* // A labirintus szerkezetét megadó állomány, szerkezete a következő:
* // a kincsek száma
* // a szörnyek száma
* // a labirintus szélessége
* // magassága
* // fal=1 járat=0 ...
* // .
* // .
* // .

```

```

* 6
* 3
* 10
* 10
* 0 0 0 1 0 1 0 1 1 1
* 0 0 0 0 0 0 0 0 0 0
* 1 0 1 0 1 0 1 0 1 0
* 0 0 0 0 1 0 1 0 0 0
* 0 1 1 0 0 0 1 1 0 1
* 0 0 0 0 1 0 0 0 0 0
* 0 1 0 0 0 1 0 1 1 0
* 0 0 0 1 0 1 0 1 0 0
* 0 1 0 0 0 0 0 0 0 1
* 0 0 0 0 1 0 0 0 1 1

```

```

* </pre>

```

```

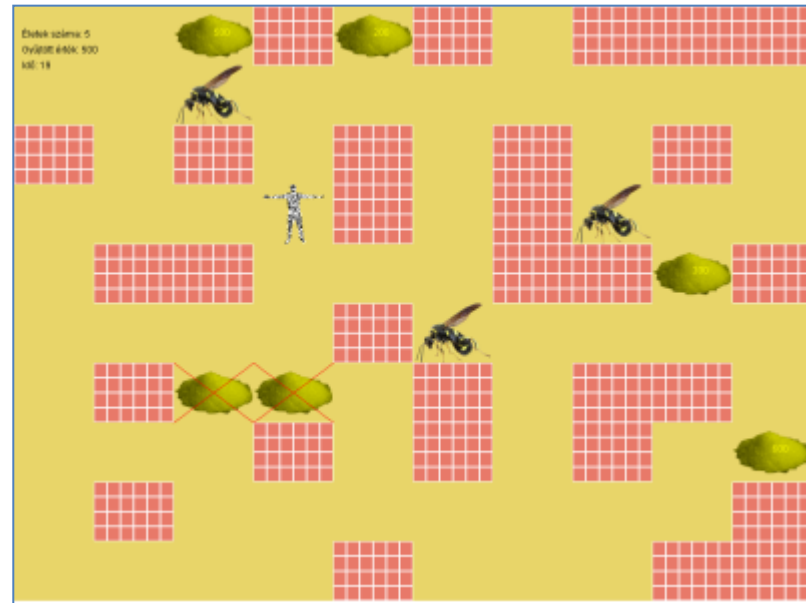
*
* @param      labirintusFájlNév      a labirintust definiáló, megfelelő
* szerkezetű szöveges állomány neve.
* @exception  RosszLabirintusKivétel  ha a labirintust definiáló állomány
* nincs meg, nem a megfelelő szerkezetű, vagy gond van az olvasásával.
*/

```

```

public Labirintus(String labirintusFájlNév) throws RosszLabirintusKivétel {

```



Kivételkezelés bevezetése

```
// Csatorna a szöveges állomány olvasásához
java.io.BufferedReader szovegesCsatorna = null;

try {
    szovegesCsatorna = new java.io.BufferedReader(
        new java.io.FileReader(labirintusFajlNév));

    String sor = szovegesCsatorna.readLine();

    while(sor.startsWith("//"))
        sor = szovegesCsatorna.readLine();

    try {

        kincsekSzama = Integer.parseInt(sor);

        sor = szovegesCsatorna.readLine();
        szornyekSzama = Integer.parseInt(sor);

        sor = szovegesCsatorna.readLine();
        szelesség = Integer.parseInt(sor);

        sor = szovegesCsatorna.readLine();
        magasság = Integer.parseInt(sor);

        szerkezet = new boolean[magasság][szelesség];
    }
}
```

Kivételkezelés bevezetése

```
try {  
  
    kincsekSzama = Integer.parseInt(sor);  
  
    sor = szovegesCsatorna.readLine();  
    szornyekSzama = Integer.parseInt(sor);  
  
    sor = szovegesCsatorna.readLine();  
    szelesség = Integer.parseInt(sor);  
  
    sor = szovegesCsatorna.readLine();  
    magasság = Integer.parseInt(sor);  
  
    szerkezet = new boolean[magasság][szelesség];  
  
} catch(java.lang.NumberFormatException e) {  
  
    throw new RosszLabirintusKivétel("Hibás a kincsek, szörnyek száma, széles  
  
}
```

Kivételkezelés bevezetése

```
for(int i=0; i<magasság; ++i) {  
  
    sor = szovegesCsatorna.readLine();  
  
    java.util.StringTokenizer st =  
        new java.util.StringTokenizer(sor);  
  
    for(int j=0; j<szélesség; ++j) {  
        String tegla = st.nextToken();  
  
        try {  
  
            if(Integer.parseInt(tegla) == 0)  
                szerkezet[i][j] = false;  
            else  
                szerkezet[i][j] = true;  
  
        } catch(java.lang.NumberFormatException e) {  
  
            System.out.println(i+" sor "+j+" oszlop "+e);  
            szerkezet[i][j] = false;  
  
        }  
    }  
}
```

Kivételkezelés bevezetése

```
} catch(java.io.FileNotFoundException e1) {  
    throw new RosszLabirintusKivétel("Nincs meg a fájl: " + e1);  
} catch(java.io.IOException e2) {  
    throw new RosszLabirintusKivétel("IO kivétel történt: "+e2);  
} catch(java.util.NoSuchElementException e3) {  
    throw new RosszLabirintusKivétel("Nem jó a labirintus szerkezete: "  
        +e3);  
} finally {  
    if(szovegesCsatorna != null) {  
        try{  
            szovegesCsatorna.close();  
        } catch(Exception e) {}  
    }  
}
```


Kivételkezelés bevezetése

```
} catch(java.io.FileNotFoundException e1) {  
  
    throw new RosszLabirintusKivétel("Nincs meg a fájl: " + e1);  
  
} catch(java.io.IOException e2) {  
    try {  
        szövegesCsatorna = new java.io.BufferedReader(  
            new java.io.FileReader(labirintusFájlNév));  
    }  
} catch(java.util.NoSuchElementException e3) {  
  
    throw new RosszLabirintusKivétel("Nincs meg a labirintus szerkezete: "  
        +e3);  
  
} finally {  
  
    if (szövegesCsato  
  
        try{  
            szöveges  
        } catch(Exce  
  
    }
```

Constructor Detail

FileReader

```
public FileReader(String fileName)  
    throws FileNotFoundException
```

Creates a new `FileReader`, given the name of the file to read from.

Parameters:

`fileName` - the name of the file to read from

Throws:

`FileNotFoundException` - if the named file does not exist, is a

<http://download.oracle.com/javase/7/docs/api/index.html?java/io/FileReader.html>

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-1-1-080904-3>

Kivételkezelés bevezetése

```
    } catch(java.io.FileNotFoundException e1) {  
  
        throw new RosszLabirintusKivétel("Nincs meg a fájl: " + e1);  
    } catch(java.io.IOException e2) {  
        for(int j=0; j<szélesség; ++j) {  
            String tegla = st.nextToken();  
  
            throw new RosszLabirintusKivétel("IO kivétel történt: " + e2);  
        }  
    } catch(java.util.NoSuchElementException e3) {  
  
        throw new RosszLabirintusKivétel("Nem jó a labirintus szerkezete: "  
            + e3);  
    } finally {  
  
        if(szövegesCsatorna != null) {  
            try{  
                szövegesCsatorna.close();  
            } catch(Exceptio
```

nextToken

```
public String nextToken()
```

Returns the next token from this string tokenizer.

Returns:

the next token from this string tokenizer.

Throws:

`NoSuchElementException` - if there are no more tokens in this tokenizer's string.

<http://download.oracle.com/javase/7/docs/api/index.html?java/util/StringTokenizer.html>

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-1-1-080904-3>

Kivételkezelés bevezetése

```
/*
 * RosszLabirintusKivétel.java
 *
 * DIGIT 2005, Javat tanítok
 * Bátfai Norbert, nbatfai@inf.unideb.hu
 *
 */
package javattanitok.labirintus;
/**
 * Ha állomány alapján készítjük a labirintust, akkor az állomány szerkezetének
 * hibáit jelzi ez az osztály.
 *
 * @author Bátfai Norbert, nbatfai@inf.unideb.hu
 * @version 0.0.1
 * @see javattanitok.labirintus.Labirintus
 */
public class RosszLabirintusKivétel extends java.lang.Exception {
    /**
     * Elkészít egy <code>RosszLabirintusKivétel</code> kivétel objektumot.
     *
     * @param hiba a hiba leírása
     */
    public RosszLabirintusKivétel(String hiba) {
        super(hiba);
    }
}
```

Kivételkezelés bevezetése

```
/**
 * A <code>LabirintusVilág</code> objektum elkészítése.
 *
 * @param labirintusFájlNév a labirintust definiáló, megfelelő
 * szerkezetű szöveges állomány neve.
 * @exception RosszLabirintusKivétel ha a labirintust definiáló állomány nem
 * a megfelelő szerkezetű
 */
public LabirintusVilág(String labirintusFájlNév)
throws RosszLabirintusKivétel {

    // A labirintus elkészítése állományból
    labirintus = new Labirintus(labirintusFájlNév);

    // A hős elkészítése és a kezdő pozíciójának beállítása
    hős = new Hős(labirintus);
    hős.sor(9);
    hős.oszlop(0);

    // A játékbeli idő folyását biztosító szál elkészítése és indítása
    new Thread(this).start();
}
```

Kivételkezelés bevezetése

```
/**
 * Átveszi a játék indításához szükséges paramétereket, majd
 * elindítja a játék világának működését.
 *
 * @param args a labirintus tervét tartalmazó állomány neve az első
 * parancssor-argumentum.
 */
public static void main(String[] args) {

    if(args.length != 1) {

        System.out.println("Indítás: java LabirintusVilág labirintus.txt");
        System.exit(-1);
    }

    try {

        new LabirintusVilág(args[0]);
    } catch(RosszLabirintusKivétel rosszLabirintusKivétel) {

        System.out.println(rosszLabirintusKivétel);
    }
}
```

Kivételkezelés bevezetése

```
/**
 * Átveszi a játék indításához szükséges paramétereket, majd
 * elindítja a játék világának működését.
 *
 * @param args a labirintus tervét tartalmazó állomány neve az első
 * parancssor-argumentum.
 */
public static void main(String[] args) {

    if (args.length != 1) {

        System.out.println("Indítás: java LabirintusVilág labirintus.txt");
        System.exit(-1);
    }

    try {

        new LabirintusVilág(args[0]);
    } catch (RosszLabirintusKivétel rosszLabirintusKivétel) {

        System.out.println(rosszLabirintusKivétel);
    }
}
```

Kivételkezelés

- 1) Mi a kivétel?
Valamilyen futási idejű hiba.
- 2) Mi egy Java kivétel?
„Javában minden objektum”, a kivételes szituáció (a futási idejű hiba) egyfajta leírása.
- 3) Elkülönül a hibakezelő kód
- 4) A kivétel keletkezik, lehet dobni, tovább dobni, deklarálni, specifikálni, elkapni, kezelni, származtatni: **try, catch, throw, throws, finally**

Kivételkezelés

1) Elkülönül a hibakezelő kód

<http://www.inf.unideb.hu/~nbatfai/p1/forrasok-SVN/bevezetes/halo/>

```
if ((osztott_memoria =
    shmget (ftok (".", 44), 2*sizeof(int), IPC_CREAT | S_IRUSR | S_IWUSR)) == -1)
{
    perror ("shmget");
    exit (EXIT_FAILURE);
}
if ((osztott_memoria_terulet = (int *)shmat (osztott_memoria, NULL, 0)) < 0)
{
    perror ("shmat");
    exit (EXIT_FAILURE);
}
*osztott_memoria_terulet = 42;
*(osztott_memoria_terulet+1) = 0;
```

<http://www.inf.unideb.hu/~nbatfai/p1/forrasok-SVN/bevezetes/vedes/>

```
std::fstream beFile (inFile, std::ios_base::in);

if(!beFile) {
    std::cout << inFile << " nem letezik..." << std::endl;
    usage();
    return -3;
}
```


Kivételkezelés

1) Elkülönül a hibakezelő kód

```
try {  
  
    if(Integer.parseInt(tegla) == 0)  
        szerkezet[i][j] = false;  
    else  
        szerkezet[i][j] = true;  
  
} catch(java.lang.NumberFormatException e) {  
  
    System.out.println(i+" sor "+j+" oszlop "+e);  
    szerkezet[i][j] = false;  
  
}
```

```
if(...jó a téglá megadása...) {  
  
    if(...járat?...)  
        ...  
    else if(...fal?...)  
        ...  
  
} else {  
  
    ...hibaág...  
  
}
```

Kivételkezelés

1) Elkülönül a hibakezelő kód

```
} catch(java.io.FileNotFoundException e1) {  
    throw new RosszLabirintusKivétel("Nincs meg a fájl: " + e1);  
} catch(java.io.IOException e2) {  
    throw new RosszLabirintusKivétel("IO kivétel történt: "+e2);  
} catch(java.util.NoSuchElementException e3) {  
    throw new RosszLabirintusKivétel("Nem jó a labirintus szerkezete: "  
        +e3);  
} finally {  
    if(szovegesCsatorna != null) {  
        try{  
            szovegesCsatorna.close();  
        } catch(Exception e) {}  
    }  
}
```

Java kivételek szervezése

Nem ellenőrzött

- java.lang.**Throwable** (implements java.io.Serializable)
 - java.lang.**Error**
 - java.lang.**AssertionError**
 - java.lang.**LinkageError**
 - java.lang.**BootstrapMethodError**
 - java.lang.**ClassCircularityError**
 - java.lang.**ClassFormatError**
 - java.lang.**UnsupportedClassVersionError**
 - java.lang.**ExceptionInInitializerError**
 - java.lang.**IncompatibleClassChangeError**
 - java.lang.**AbstractMethodError**
 - java.lang.**IllegalAccessError**
 - java.lang.**InstantiationError**
 - java.lang.**NoSuchFieldError**
 - java.lang.**NoSuchMethodError**
 - java.lang.**NoClassDefFoundError**
 - java.lang.**UnsatisfiedLinkError**
 - java.lang.**VerifyError**
 - java.lang.**ThreadDeath**
 - java.lang.**VirtualMachineError**
 - java.lang.**InternalError**
 - java.lang.**OutOfMemoryError**
 - java.lang.**StackOverflowError**
 - java.lang.**UnknownError**

Ellenőrzött

- java.lang.**Exception**
 - java.lang.**CloneNotSupportedException**
 - java.lang.**InterruptedException**
 - java.lang.**ReflectiveOperationException**
 - java.lang.**ClassNotFoundException**
 - java.lang.**IllegalAccessException**
 - java.lang.**InstantiationException**
 - java.lang.**NoSuchFieldException**
 - java.lang.**NoSuchMethodException**
 - java.lang.**RuntimeException**
 - java.lang.**ArithmeticException**
 - java.lang.**ArrayStoreException**
 - java.lang.**ClassCastException**
 - java.lang.**EnumConstantNotPresentException**
 - java.lang.**IllegalArgumentException**
 - java.lang.**IllegalThreadStateException**
 - java.lang.**NumberFormatException**
 - java.lang.**IllegalMonitorStateException**
 - java.lang.**IllegalStateException**
 - java.lang.**IndexOutOfBoundsException**
 - java.lang.**ArrayIndexOutOfBoundsException**
 - java.lang.**StringIndexOutOfBoundsException**
 - java.lang.**NegativeArraySizeException**
 - java.lang.**NullPointerException**
 - java.lang.**SecurityException**
 - java.lang.**TypeNotPresentException**
 - java.lang.**UnsupportedOperationException**

Nem ellenőrzött

Java kivételek szervezése

- java.lang.**Exception**

- java.lang.**CloneNotSupportedException**
- java.lang.**InterruptedException**
- java.lang.**ReflectiveOperationException**
 - java.lang.**ClassNotFoundException**
 - java.lang.**IllegalAccessException**
 - java.lang.**InstantiationException**
 - java.lang.**NoSuchFieldException**
 - java.lang.**NoSuchMethodException**
- java.lang.**RuntimeException**
 - java.lang.**ArithmeticException**
 - java.lang.**ArrayStoreException**
 - java.lang.**ClassCastException**
 - java.lang.**EnumConstantNotPresentException**
 - java.lang.**IllegalArgumentException**
 - java.lang.**IllegalThreadStateException**
 - java.lang.**NumberFormatException**
 - java.lang.**IllegalMonitorStateException**
 - java.lang.**IllegalStateException**
 - java.lang.**IndexOutOfBoundsException**
 - java.lang.**ArrayIndexOutOfBoundsException**
 - java.lang.**StringIndexOutOfBoundsException**
 - java.lang.**NegativeArraySizeException**
 - java.lang.**NullPointerException**
 - java.lang.**SecurityException**
 - java.lang.**TypeNotPresentException**
 - java.lang.**UnsupportedOperationException**

- java.lang.**Throwable** (implements java.io.Serializable)

- java.lang.**Error**
 - java.util.**ServiceConfigurationError**
- java.lang.**Exception**
 - java.io.**IOException**
 - java.util.**InvalidPropertiesFormatException**
 - java.lang.**RuntimeException**
 - java.util.**ConcurrentModificationException**
 - java.util.**EmptyStackException**
 - java.lang.**IllegalArgumentException**
 - java.util.**IllegalFormatException**
 - java.util.**DuplicateFormatFlagsException**
 - java.util.**FormatFlagsConversionMismatchException**
 - java.util.**IllegalFormatCodePointException**
 - java.util.**IllegalFormatConversionException**
 - java.util.**IllegalFormatFlagsException**
 - java.util.**IllegalFormatPrecisionException**
 - java.util.**IllegalFormatWidthException**
 - java.util.**MissingFormatArgumentException**
 - java.util.**MissingFormatWidthException**
 - java.util.**UnknownFormatConversionException**
 - java.util.**UnknownFormatFlagsException**
 - java.lang.**IllegalStateException**
 - java.util.**FormatterClosedException**
 - java.util.**IllformedLocaleException**
 - java.util.**MissingResourceException**
 - java.util.**NoSuchElementException**
 - java.util.**InputMismatchException**
 - java.util.**TooManyListenersException**



Párhuzamosság

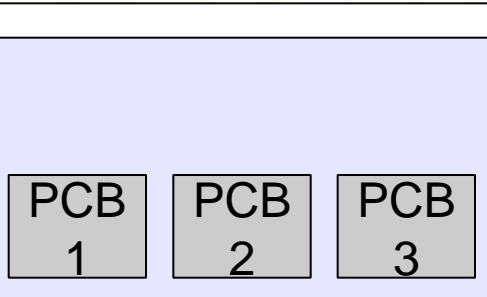
Ismétlés: szálak

MM 61

Memória

p1/3. prezi

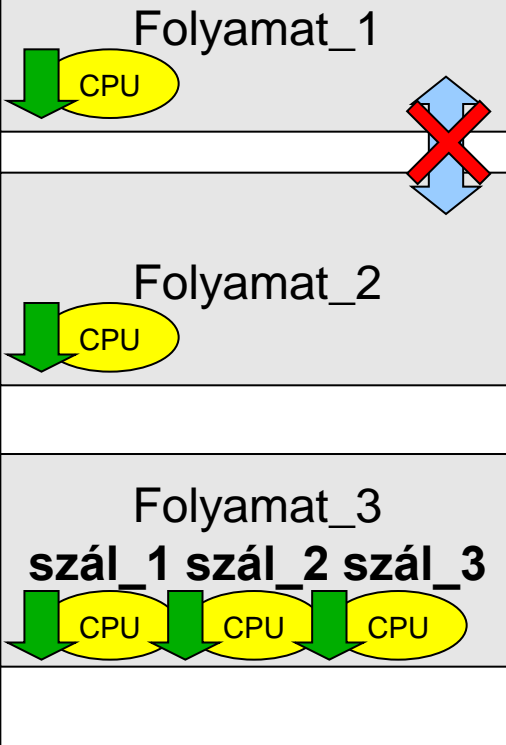
Kernel címtér



```
PTHREADS(7)                Linux Programmer's Manual
NAME
    pthreads - POSIX threads

DESCRIPTION
    A single process can contain multiple threads,
    all of which are executing the same program.
    These threads share the same global memory (data
    and heap segments), but each thread has its
    own stack (automatic variables).
    ...
```

Felhasználói címtér



Folyamat_3

Verem szál_1	Verem szál_2	Verem szál_3
Heap		
Data		
Text		

Lokális változók,
paraméterátadás.
malloc()
Adatok
Kód

Isméltés: (POSIX threads) pthreads könyvtár,

p1/3. prezi

pthread_

PTHREADS (7)

Linux Programmer's Manual

PTHREADS (7)

NAME

pthread - POSIX threads

DESCRIPTION

POSIX.1 specifies a set of interfaces (functions, header files) for threaded programming commonly known as POSIX threads, or Pthreads. A single process can contain multiple threads, all of which are executing the same program. These threads share the same global memory (data and heap segments), but each thread has its own stack (automatic variables).

PTHREAD_CREATE (P)

POSIX Programmer's Manual

PTHREAD_CREATE (P)

NAME

pthread_create - thread creation

SYNOPSIS

```
#include <pthread.h>
```

```
int pthread_create(pthread_t *restrict thread,  
                  const pthread_attr_t *restrict attr,  
                  void *(*start_routine) (void*), void *restrict arg);
```

DESCRIPTION

The `pthread_create()` function shall create a new thread, with

Ismétlés: pthreads könyvtár, mutex zárok, pthreads_

p1/4. prezí

PTHREAD_MUTEX_LOCK(P)

POSIX Programmer's Manual

PTHREAD_MUTEX_LOCK(P)

```
...  
#define SZALAK_SZAMA 100  
int szamlalo = 0;
```

```
...  
  
int  
main(void)  
{  
    pthread_t sz[SZALAK_SZAMA];  
    ...  
}
```

The mutex object
pthread_mutex_lock
thread shall block
shall return with
state with the ca

PP 67

```
.  
. .  
. .  
Szal: 98, 1622116  
Szal: 96, 1589346  
Szal: 98, 1622116  
Szal: 96, 1589346  
Szal: 96, 1589346  
A szamlalo vegul: -2
```

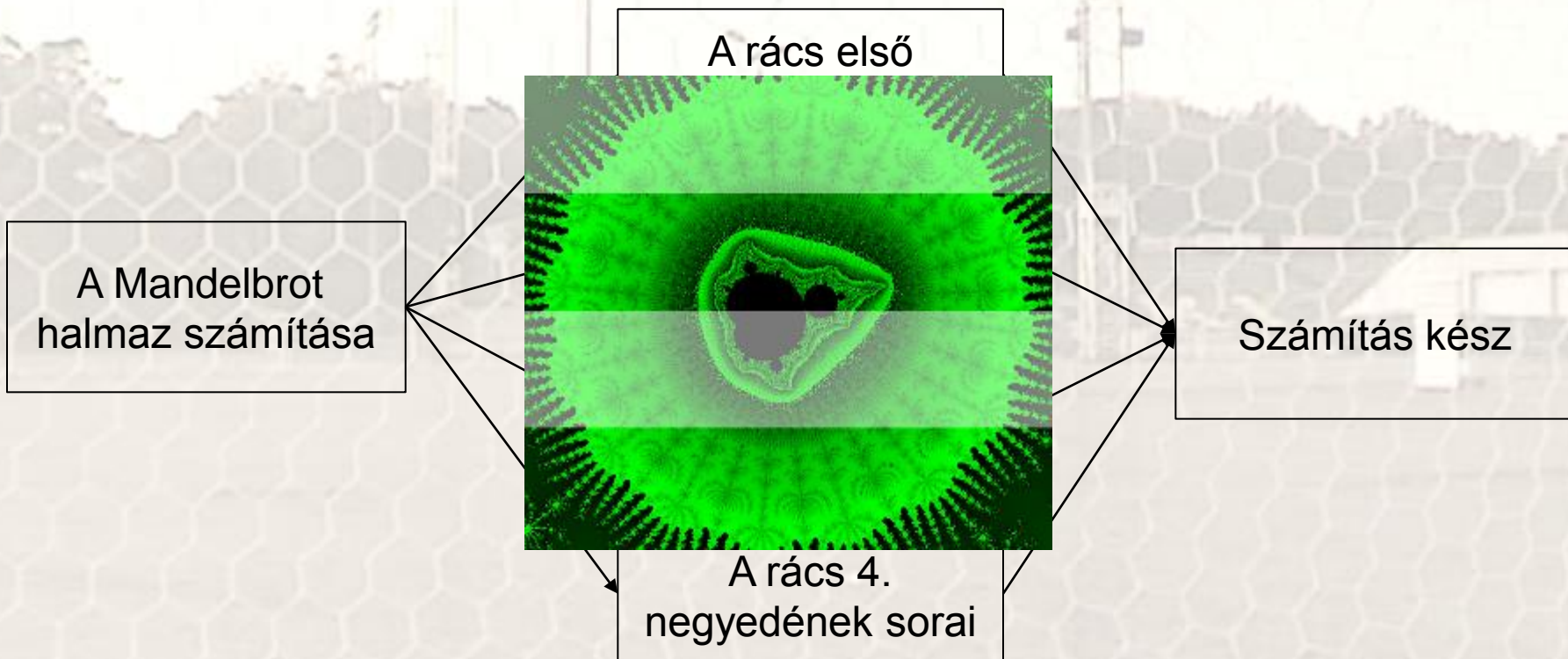
pthread_mutex_trylock, pthread_mutex_unlock - lock

```
void *  
novel_szal(void *id)  
{  
    int i;  
    for(i=0; i<100; ++i)  
    {  
        printf("Szal: %d, %d\n", *(int *)id, pthread_self());  
        fflush(stdout);  
        var();  
        szamlalo = szamlalo + 1;  
    }  
    return id;
```

```
void *  
csokkent_szal(void *id)  
{  
    int i;  
    for(i=0; i<100; ++i) {  
        printf("Szal: %d, %d\n", *(int *)id, pthread_self());  
        fflush(stdout);  
        var();  
        szamlalo = szamlalo - 1;  
    }  
    return id;  
}
```


Ismétlés: Labor, párhuzamos programozás

p1/7. prezi



Ismétlés: Párhuzamos programozás, P-szálak

p1/7. prezi

A rács első
negyedének sorai

A rács 2.

A Mandelbrot
halmaz szá

```
void *mandel_resz_szamitas(void *id)
{
    int mettol = *(int *)id * (magassag / SZALAK_SZAMA);
    int meddig = mettol + (magassag/SZALAK_SZAMA);

    // a számítás
    double dx = (b-a)/szelesseg;
    double dy = (d-c)/magassag;
    double reC, imC, reZ, imZ, ujreZ, ujimZ;
    // Hány iterációt csináltunk?
    int iteracio = 0;
    std::cout << *(int *)id << ". szal szamitas indul";
    // Végigzongorázzuk a szélesség x magasság rácsot:
    for (int j=mettol; j<meddig; ++j) {
        //sor = j;
        for (int k=0; k<szelesseg; ++k) {
            // c = (reC, imC) a rács csomópontjainak
            // megfelelő komplex szám

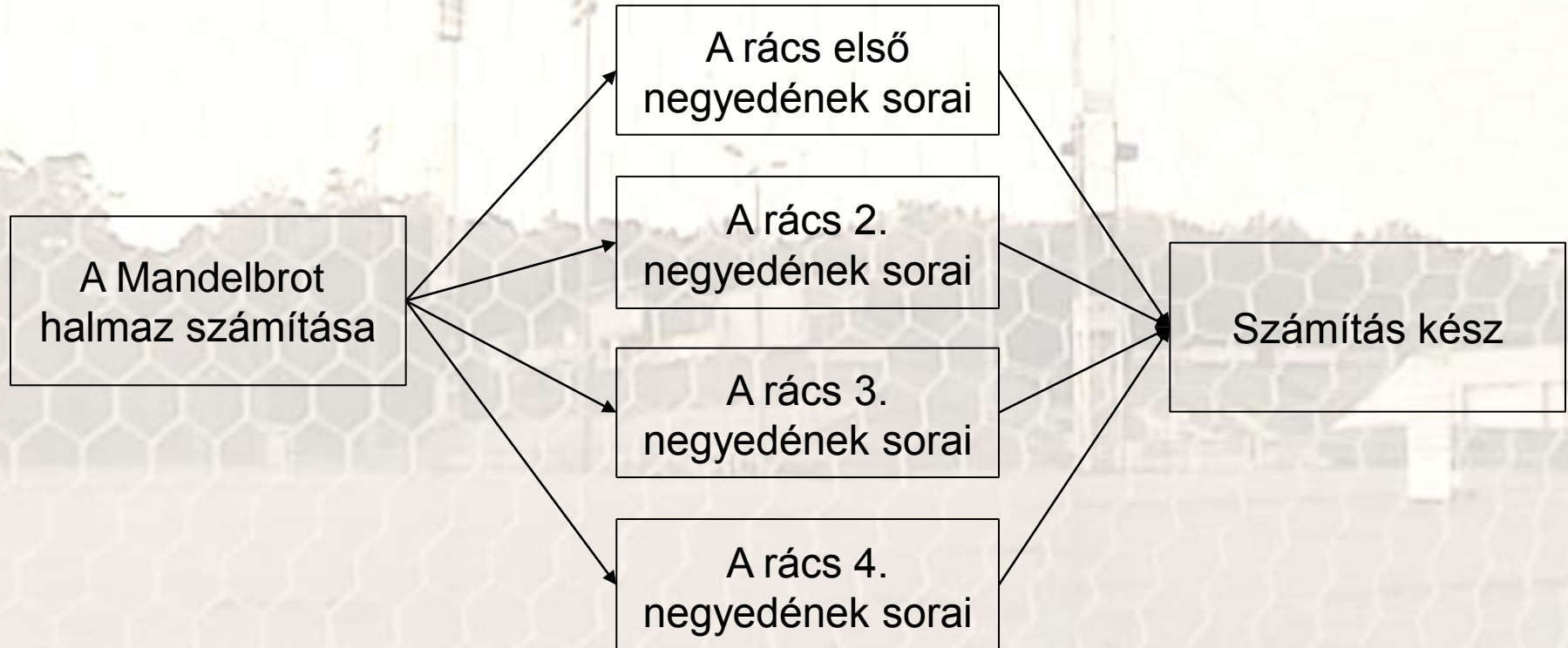
```

tás kész

Ismétlés: Párhuzamos programozás, OpenMP

p1/7. prezi

Open Multi-Processing



```
// Végigzongorázzuk a szélesség x magasság rácsot:
```

```
#pragma omp parallel for
```

```
for (int j=0; j<magassag; ++j) {
```

```
    //sor = j;
```

```
    for (int k=0; k<szelesseg; ++k) {
```

```
        // c = (reC, imC) a rács csomópontjainak
```

```
        // megfelelő komplex szám
```

```
        reC = a+k*dx;
```

Ciklus párhuzamosítása

Ismétlés: Szekvenciális vs. párhuzamos

p1/7. prezi

The image displays several overlapping terminal windows from a Linux system. The top-most window shows the output of the `top` command, which lists system statistics and a table of running processes. In this table, the `%CPU` column for processes `16302 norbi`, `16303 norbi`, and `16304 norbi` is circled in red, showing values of 93.1, 92.7, and 14.9 respectively. Below this, another terminal window shows the execution of a C++ program `ompmandelpngt` with the `time` command. The output shows a real time of `0m30.07s` and a user time of `0m29.98s`, with the `0m30.07s` value underlined in red. A third terminal window shows the compilation of the program with `g++` and the use of OpenMP flags `-fopenmp -O3`. The compilation time is shown as `0m8.597s` (real) and `0m18.025s` (user), with the `0m8.597s` value underlined in red. The background of the slide features a faint image of a classical building.

Ismétlés: Szekvenciális vs. párhuzamos

p1/7. prezi

```
[norbi@sgu exor]$ gcc t.c -O3 -o t -std=c99
```

```
[norbi@sgu exor]$ time ./t <titkos.szoveg |grep 59934434
```

```
59934434Már komoly mennyiségű munkát elvégeztünk ebben a félévben (eddig is, de még sok élmény vár :), ezért aki nem készült heti gyakorisággal, az úgy érezheti: elmélyült körülötte a víz. Itt a tavaszi szünet holnaptól, az érintve érzett hallgatónak azt ajánlom, hogy pótolják be az elmaradt munkát.
```

```
élményeket, tapasztalatokat, azaz programozzanak. Kezdjék ennek a blognak az elején és fogják fel olyan on-line tutoriálnak, ahol (lehetőleg) gyorsan választ is adunk a kommentben felmerült kérdésekre! Ha a programozás helyett a kurzust átszervező elmélkedésbe kezdenél,
```

```
real    3m34.691s
user    3m33.855s
sys     0m0.069s
```

```
[norbi@sgu exor]$ gcc t.c -fopenmp -O3 -o t -std=c99
```

```
[norbi@sgu exor]$ time ./t <titkos.szoveg |grep 59934434
```

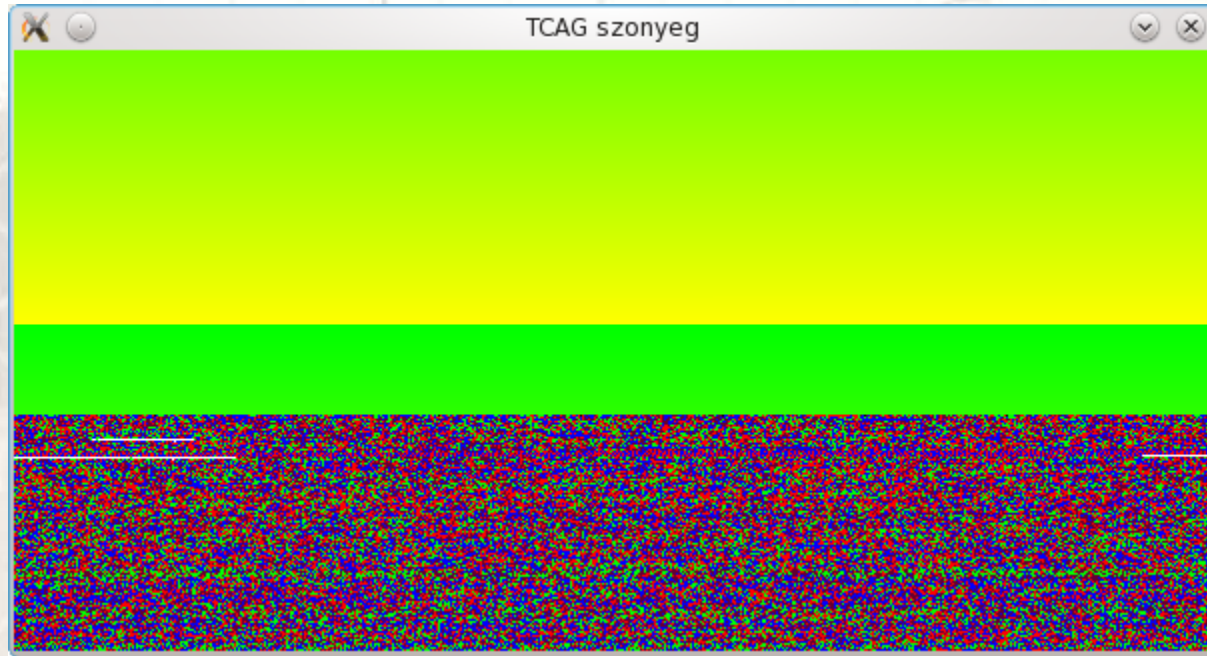
```
59934434Már komoly mennyiségű munkát elvégeztünk ebben a félévben (eddig is, de még sok élmény vár :), ezért aki nem készült heti gyakorisággal, az úgy érezheti: elmélyült körülötte a víz. Itt a tavaszi szünet holnaptól, az érintve érzett hallgatónak azt ajánlom, hogy pótolják be az elmaradt munkát, élményeket, tapasztalatokat, azaz programozzanak. Kezdjék ennek a blognak az elején és fogják fel olyan on-line tutoriálnak, ahol (lehetőleg) gyorsan választ is adunk a kommentben felmerült kérdésekre! Ha a programozás helyett a kurzust átszervező elmélkedésbe kezdenél, akkor kezd ezzel :)
```

```
real    1m10.466s
user    3m50.372s
sys     0m0.057s
```

Ismétlés: Szőnyegen a humán genom

p1/5. prezi

http://progpater.blog.hu/2011/03/05/szonyegen_a_human_genom



OO szempontból teljesen ua., mint az előző.

Ismétlés: Szőnyegen a humán genom

p1/5. prezi

http://progpater.blog.hu/2011/03/05/szonyegen_a_human_genom

```
class GenSzal : public QThread
{
    Q_OBJECT

public:
    GenSzal(const char *fajlNev, int szelesseg, GenAblak *genAblak);
    ~GenSzal();
    void run();

protected:
    int szelesseg;
    GenAblak *genAblak;
    int *egySor;
    QFile *fajl;
};
```

genszal.h

Ismé

p1/5. prezi <http://>

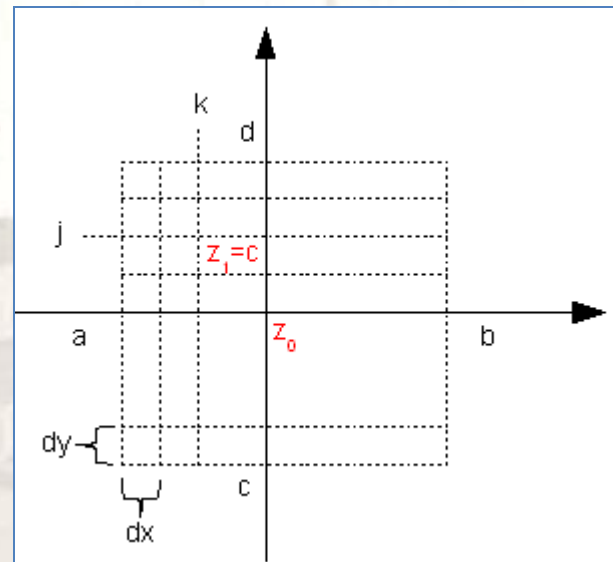
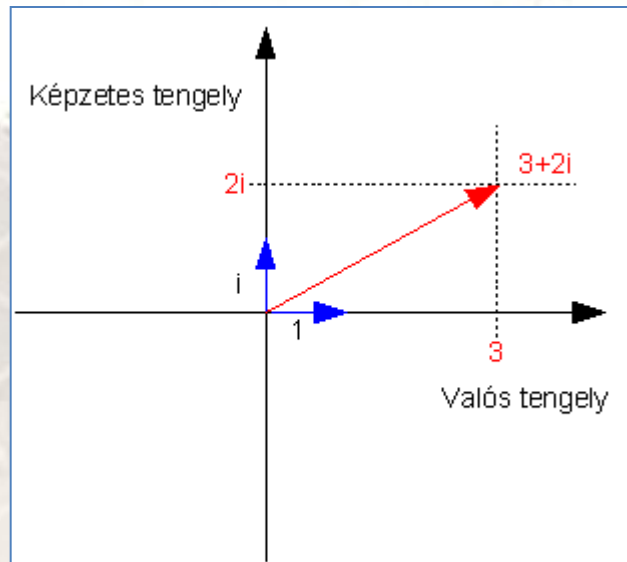
```
#include "g  
  
GenSzal::Gen  
{  
    this->s  
    this->g  
  
    egySor =  
  
    fajl =  
    fajl->op  
  
}
```

```
void GenSzal::run()  
{  
    while(true) {  
        QThread::msleep(80);  
  
        char c;  
        for(int i=0; i < szelesseg && (fajl->getChar(&c) != -1); ++i) {  
  
            if(c==0x0a)  
                continue;  
  
            switch(c)  
            {  
                case 'T':  
                    egySor[i] = qRgb (255, 0, 0);  
                    break;  
                case 'C':  
                    egySor[i] = qRgb (0, 255, 0);  
                    break;  
                case 'A':  
                    egySor[i] = qRgb (0, 0, 255);  
                    break;  
                case 'G':  
                    egySor[i] = qRgb (100, 0, 50);  
                    break;  
                default:  
                    egySor[i] = qRgb (255, 255, 255);  
                    break;  
            }  
        }  
        genAblak->vissza(egySor, szelesseg);  
    }  
}
```


Ismétlés: Mandelbrot halmaz

http://progpater.blog.hu/2011/02/26/tan_csodallak_amde_nem_ertelek_de_kepzetem_hegyvolgyedet_bejarja

p1/5. prezi



```
void FraksZal::run()
{
    // A [a,b]x[c,d] tartományon milyen sűrű a
    // megadott szélesség, magasság háló:
    double dx = (b-a)/szelesseg;
    double dy = (d-c)/magassag;
```

<http://www.tankonyvtar.hu/informatika/javat-tanitok-2-2-080904-1>

Ismétlés: Mandelbrot halmaz

p1/5. prezi

```
// Hány iterációt csináltunk?  
int iteracio = 0;  
// Végigzongorázzuk a szélesség x magasság hálót:  
for(int j=0; j<magassag; ++j) {  
    //sor = j;  
    for(int k=0; k<szelesseg; ++k) {  
        // c = (reC, imC) a háló rácspontjainak  
        // megfelelő komplex szám  
        reC = a+k*dx;  
        imC = d-j*dy;  
        // z_0 = 0 = (reZ, imZ)
```



```
    = 0;  
    // z_n = z_{n-1} * z_{n-1} + c iterációk  
    // száma, amíg |z_n| < 2 vagy még  
    // nem elük el a 255 iterációt, ha  
    // nem elértük, akkor úgy vesszük,  
    // hogy kiindulási c komplex számra  
    // iteráció konvergens, azaz a c a  
    // Mandelbrot halmaz eleme  
    if (reZ*reZ + imZ*imZ < 4 && iteracio < iteraciosHatar) {  
        reZ+1 = reZ*reZ - imZ*imZ + reC;  
        imZ+1 = 2*reZ*imZ + imC;  
        ujreZ = reZ+1;  
        ujimZ = imZ+1;  
        iteracio++;  
    }  
}
```

Ismétlés: Mandelbrot halmaz

p1/5. prezi

```
class FraksZal : public QThread
{
    Q_OBJECT

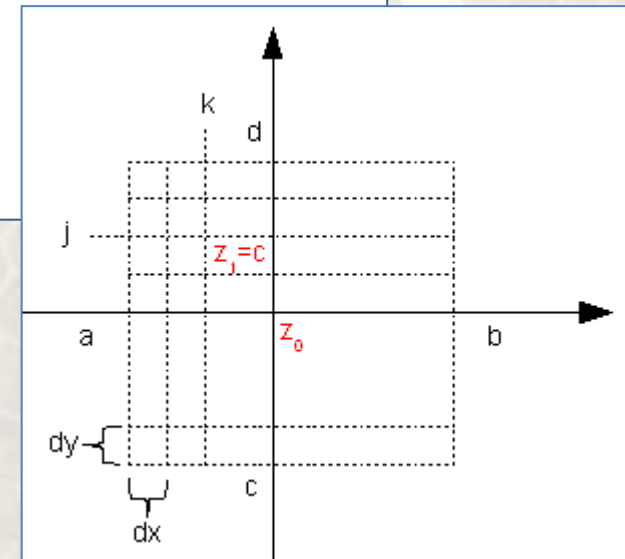
public:
    FraksZal(double a, double b, double c, double d,
            int szelesseg, int magassag, int iteraciosHatar, FrakAblak *frakAblak);
    ~FraksZal();
    void run();

protected:
    // A komplex sík vizsgált tartománya [a,b]x[c,d].
    double a, b, c, d;
    // A komplex sík vizsgált tartományára feszített
    // háló szélessége és magassága.
    int szelesseg, magassag;
    // Max. hány lépésig vizsgáljuk a  $z_{n+1} = z_n * z_n + c$  iterációt?
    // (tk. most a nagyítási pontosság)
    int iteraciosHatar;
    // Kinek számolok?
    FrakAblak* frakAblak;
    // Soronként küldöm is neki vissza a kiszámoltakat.
    int* egySor;
};
```

frakszal.h

A Java párhuzamosság bevezetése

```
/*
 * MandelbrotHalmaz.java
 *
 * DIGIT 2005, Javat tanítok
 * Bاتفai Norbert, nbatfai@inf.unideb.hu
 *
 */
/**
 * A Mandelbrot halmazt kiszámoló és kirajzoló osztály.
 *
 * @author Bاتفai Norbert, nbatfai@inf.unideb.hu
 * @version 0.0.1
 */
public class MandelbrotHalmaz extends java.awt.Frame implements Runnable {
    /** A komplex sík vizsgált tartománya [a,b]x[c,d]. */
    protected double a, b, c, d;
    /** A komplex sík vizsgált tartományára feszített
     * háló szélessége és magassága. */
    protected int szélesség, magasság;
```



A J

[http://download](http://download.oracle.com/javase/6/docs/api/java/lang/Runnable.html)

public in

The Runnable

This interface
simply mean

In addition, R
Thread inst
methods. Thi

Since:

JDK1.0

See Also:

Thread,

Method

Methods

Modifier

void

```
package java.lang;
```

```
/**
```

```
* The Runnable interface should be implemented by any  
* class whose instances are intended to be executed by a thread. The  
* class must define a method of no arguments called run.  
* 

* This interface is designed to provide a common protocol for objects that  
* wish to execute code while they are active. For example,  
* Runnable is implemented by class Thread.


```

```
...
```

```
*
```

```
* @author Arthur van Hoff  
* @see java.lang.Thread  
* @see java.util.concurrent.Callable  
* @since JDK1.0
```

```
*/
```

```
public  
interface Runnable {
```

```
/**
```

```
* When an object implementing interface Runnable is used  
* to create a thread, starting the thread causes the object's  
* run method to be called in that separately executing  
* thread.
```

```
* 
```

```
* The general contract of the method run is that it may  
* take any action whatsoever.
```

```
*
```

```
* @see java.lang.Thread#run()
```

```
*/
```

```
public abstract void run();
```

```
}
```

A Java párhuzamosság bevezetése

```
/**
 * A Mandelbrot halmaz számítási algoritmus.
 * Az algoritmus részletes ismertetését lásd például a
 * [BARNSELY KÖNYV] (M. Barnsley: Fractals everywhere,
 * Academic Press, Boston, 1986) hivatkozásban vagy
 * ismeretterjesztő szinten a [CSÁSZÁR KÖNYV] hivatkozásban.
 */
public void run() {
    // A [a,b]x[c,d] tartományon milyen sűrű a
    // megadott szélesség, magasság háló:
    double dx = (b-a)/szélesség;
    double dy = (d-c)/magasság;
    double reC, imC, reZ, imZ, ujreZ, ujimZ;
    int rgb;
    // Hány iterációt csináltunk?
    int iteráció = 0;
    // Végigzongorázzuk a szélesség x magasság hálót:
    for(int j=0; j<magasság; +
        sor = j;
        for(int k=0; k<széless
            // c = (reC, imC)
            // megfelelő kompl
            reC = a+k*dx;
            imC = d-j*dy;
            // Ablak tulajdonságai
            setTitle("A Mandelbrot halmaz");
            setResizable(false);
            setSize(szélesség, magasság);
            setVisible(true);
            // A számítás indul:
            számításFut = true;
            new Thread(this).start();
        }
    }
```

A Java párhuzamosság bevezetése

```
// A billentyűzetről érkező események feldolgozása
addKeyListener(new java.awt.event.KeyAdapter() {
    // Az 's', 'n' és 'm' gombok lenyomását figyeljük
    public void keyPressed(java.awt.event.KeyEvent e) {
        if(e.getKeyCode() == java.awt.event.KeyEvent.VK_S)
            pillanatfelvétel();
        // Az 'n' gomb benyomásával pontosabb számítást végzünk.
        else if(e.getKeyCode() == java.awt.event.KeyEvent.VK_N) {
            if(számításFut == false) {
                MandelbrotHalmaz.this.iterációsHatár += 256;
                // A számítás újra indul:
                számításFut = true;
                new Thread(MandelbrotHalmaz.this).start();
            }
        }
    }
});
```

A Java párhuzamosság általában

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-1-8-080904>

Szál objektumok

Thread objektumok

Runnable objektumok

Párhuzamos példa – bevezető (1)

```
public class EgyszeruSzerver {  
  
    public static void main(String[] args) {  
        try {  
            java.net.ServerSocket serverSocket =  
                new java.net.ServerSocket(2005);  
            while (true) {  
                java.net.Socket socket = serverSocket.accept();  
                java.io.PrintWriter kimenoCsatorna =  
                    new java.io.PrintWriter(socket.getOutputStream());  
                kimenoCsatorna.println(  
                    new java.util.Date().toString());  
                kimenoCsatorna.flush();  
                socket.close();  
            }  
        } catch (java.io.IOException ioE) {  
            ioE.printStackTrace();  
        }  
    }  
}
```

Párhuzamos példa – bevezető (2)

```
public class EgyszeruTobbszaluSzerer {  
  
    public static void main(String[] args) {  
        try {  
            java.net.ServerSocket serverSocket =  
                new java.net.ServerSocket(2005);  
            while (true) {  
                java.net.Socket socket = serverSocket.accept();  
                new KiszolgaloSzal(socket);  
            }  
        } catch  
    }  
  
    class KiszolgaloSzal implements Runnable {  
  
        public void run() {  
            try {  
                java.io.BufferedReader bejovoCsatorna =  
                    new java.io.BufferedReader(  
                        new java.io.InputStreamReader(socket.getInputStream()));  
                java.io.PrintWriter kimenoCsatorna =  
                    new java.io.PrintWriter(socket.getOutputStream());  
                String vizshang = bejovoCsatorna.readLine();  
                kimenoCsatorna.println(vizshang);  
                kimenoCsatorna.flush();  
                socket.close();  
            } catch (java.io.IOException ioE) {  
                ioE.printStackTrace();  
            }  
        }  
    }  
}
```

Párhuzamos példa

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-2-3-080904>

C:\ Parancssor - java CsevegőSzerver

```
Microsoft Windows XP [verziószám: 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
```

```
C:\Documents and Settings\Bátfai Norbert>javac CsevegőSzerver.java
```

```
C:\Documents and Settings\Bátfai Norbert>java CsevegőSzerver
```

C:\ Telnet localhost

```
alma fönt be
alma fönt be
körte lent be
```

C:\ Telnet localhost

```
alma fönt be
körte lent be
körte lent be
```

Párhuzamos példa

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-2-3-080904>

```
/** A csevegő szerver elindítása. */
public static void main(String [] args) {
    new CsevegőSzerver();
}
```

```
/**
 * A csevegő szerver.
 *
 * @author Bátfai Norbert, nbatfai@inf.unideb.hu
 * @version 0.0.1
 */
public class CsevegőSzerver {
    /** Maximum hány csevegő fér be. */
    public static final int MAX_CSEVEGŐ = 20;
    /** Itt tartjuk az éppen csevegést lebonyolító szálakat. */
    private java.util.List<Kiszolgáló> csevegők;
    /** Itt tartjuk az éppen csevegést nem bonyolító szálakat. */
    private java.util.List<Kiszolgáló> nemCsevegők;
    /** A <code>CsevegőSzerver</code> objektumot felépítő konstruktor. */
```

Párhuzamos példa

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-2-3-080904>

```
/** A <code>CsevegőSzerver</code> objektumot felépítő konstruktor. */
public CsevegőSzerver() {
    nemCsevegők = new java.util.ArrayList<Kiszolgáló>();
    csevegők = new java.util.ArrayList<Kiszolgáló>();
    // Csevegő szálak elkészítése
    for(int i=0; i<MAX_CSEVEGŐ; ++i)
        nemCsevegők.add(new Kiszolgáló(this));
}
```

```
class Kiszolgáló implements Runnable {
    /** A csevegő szerver, aki ezt a kiszolgáló szálát készítette és
     * használja. */
    CsevegőSzerver szerver;
    /** A kiszolgálást ezen a TCP/IP kapcsolatot absztraháló
     * objektumon keresztül végezzük éppen. */
    java.net.Socket socket;
    /** A kapcsolat feletti kimenő csatorna (látnia kell a többi
     * kiszolgálást végző szálak is) */
    java.io.PrintWriter kimenőCsatorna;
    /** Dolgozik éppen vagy sem az objektum? */
    boolean kiszolgál = false;
}
```

Párhuzamos példa

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-2-3-080904>

```
/**
 * A {@code Kiszolgáló} objektumot felépítő konstruktor.
 *
 * @param  szerver a csevegő szálakat összefogó szerver.
 */
public Kiszolgáló(CsevegőSzerver szerver) {

    this.szerver = szerver;
    // Készítjük a szálát és indítjuk, az ennek
    // megfelelő run() módszerben pedig azonnal
    // altatjuk majd az indított szálát.
    new Thread(this).start();
}
}
```

```
/**
 * A kliensek kiszolgálását végző szál.
 */
public synchronized void run() {
    for(;;) {
        // Tétlenül várakozik addig, amig nincs kliens
        while(!kiszolgál)
            try{
                // a várakozásból a notify() hívás ébreszti majd fel.
                wait();
            } catch(InterruptedException e) {}
    }
}
```

Párhuzamos példa

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-2-3-080904>

```
public CsevegőSzerver() {
    nemCsevegők = new java.util.ArrayList<Kiszolgáló>();
    csevegők = new java.util.ArrayList<Kiszolgáló>();
    // Cs
    for(i = 0; i < 10; i++) {
        // Sz
        try {
            j = serverSocket.accept();
            w = new java.net.Socket(j.getInetAddress(), j.getPort());
            while(true) {
                java.net.Socket socket = serverSocket.accept();
                Kiszolgáló szál = beszáll();
                if(szál == null) {
                    // Ha betelt a csevegő szoba, akkor egy
                    // rövid tájékoztató üzenet a kliensnek:
                    java.io.PrintWriter kimenőCsatorna =
                        new java.io.PrintWriter(socket.getOutputStream());
                    kimenőCsatorna.println("A csevegő szoba tele van!");
                    socket.close();
                } else // Ha van szabad csevegő szál, akkor
                    // azzal elkezdődik a csevegő kliens kiszolgálása.
                    szál.kiszolgál(socket);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
public synchronized Kiszolgáló beszáll() {
    if(!nemCsevegők.isEmpty()) {
        Kiszolgáló kiszolgáló = nemCsevegők.remove(0);
        csevegők.add(kiszolgáló);
        return kiszolgáló;
    }
    return null;
}
```

Párhuzamos példa

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-2-3-080904>

```
/**
 * A szál kiszolgálását indító függvény.
 *
 * @param socket kapcsolat a TCP/IP-s klienssel.
 */
public void indit() {
    try {
        wait();
    } catch (InterruptedException e) {}
    // Kimenő és bejövő csatorna objektumokat elkészítjük.
    try {
        java.io.BufferedReader bejövőCsatorna =
            new java.io.BufferedReader(
                new java.io.InputStreamReader(socket.getInputStream()));
        java.io.OutputStream kimenőCsatorna =
            socket.getOutputStream();
    } catch (IOException e) {}

    /**
     * Üzenet a kliensnek.
     *
     * @param üzenet amit el köldünk a kliensnek.
     */
    public void üzenet(String üzenet) {
        kimenőCsatorna.println(üzenet);
        kimenőCsatorna.flush();
    }

    // SZ
    for (Kiszolgáló csevegő: csevegők)
        csevegő.üzenet(üzenet);
} while (true);
}
```


Párhuzamos példa

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-2-3-080904>

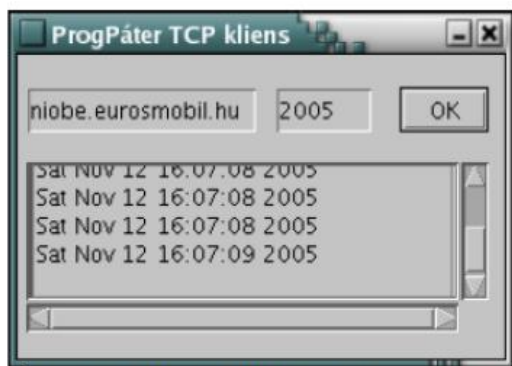
```
    } catch(java.io.IOException ioE) {  
        ioE.printStackTrace();  
    }  
    finally {  
        // ...  
    }  
}
```

```
/**  
 * A kliensek kiszolgálását végző szál.  
 */  
public synchronized void run() {  
    for(;;) {  
        // Tétlenül várakozik addig, amig nincs kliens  
        while(!kiszolgál)  
            try{  
                // a várakozásból a notify() hívás ébreszti majd fel.  
                wait();  
            } catch(InterruptedException e) {}  
    }  
}
```

```
*  
 * @param csevegő szál, aki éppen befejezte működését.  
 */  
public synchronized void kiszáll(Kiszolgáló csevegő) {  
    csevegők.remove(csevegő);  
    nemCsevegők.add(csevegő);  
}
```

Párhuzamossági buktatók

Oprendszer emlékeztető



Ábra 25: AWT TCP kliens Linux alatt a hálózati rész szerveihez.



Ábra 24: AWT TCP kliens a hálózati rész szerveihez.



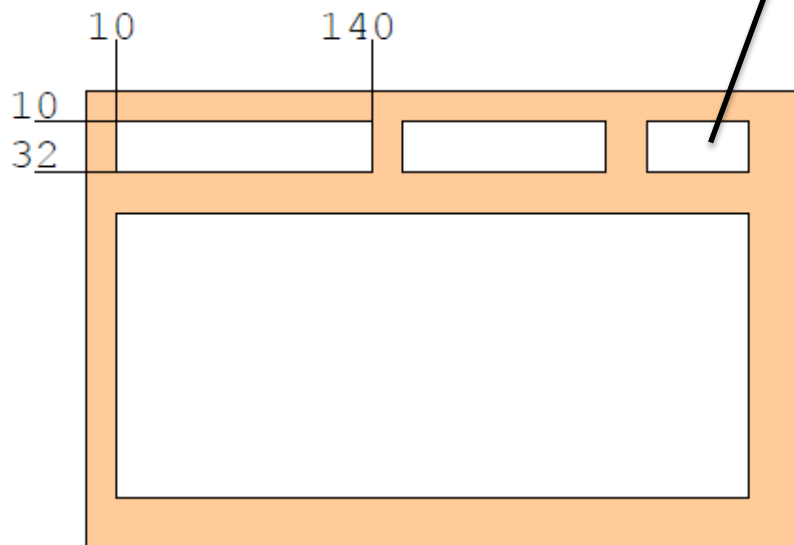
Ábra 23: Mivel blokkoltuk az eseménykezelő szálát, így a felület átmenetileg "lefagy".

VIII.1.1.3 ProgPáter TCP kliens

A legtöbb Java GUI-s példában mindenféle elhelyezé
ismerkedhetünk m
leskicceljük és dire

Építsük fel

```
okButton.addActionListener(  
    new java.awt.event.ActionListener() {  
        public void actionPerformed(java.awt.event.ActionEvent  
e) {  
            kattint(e);  
        }  
    });
```



Ábra 22: A felület skicce

Párhuzamosági buktatók

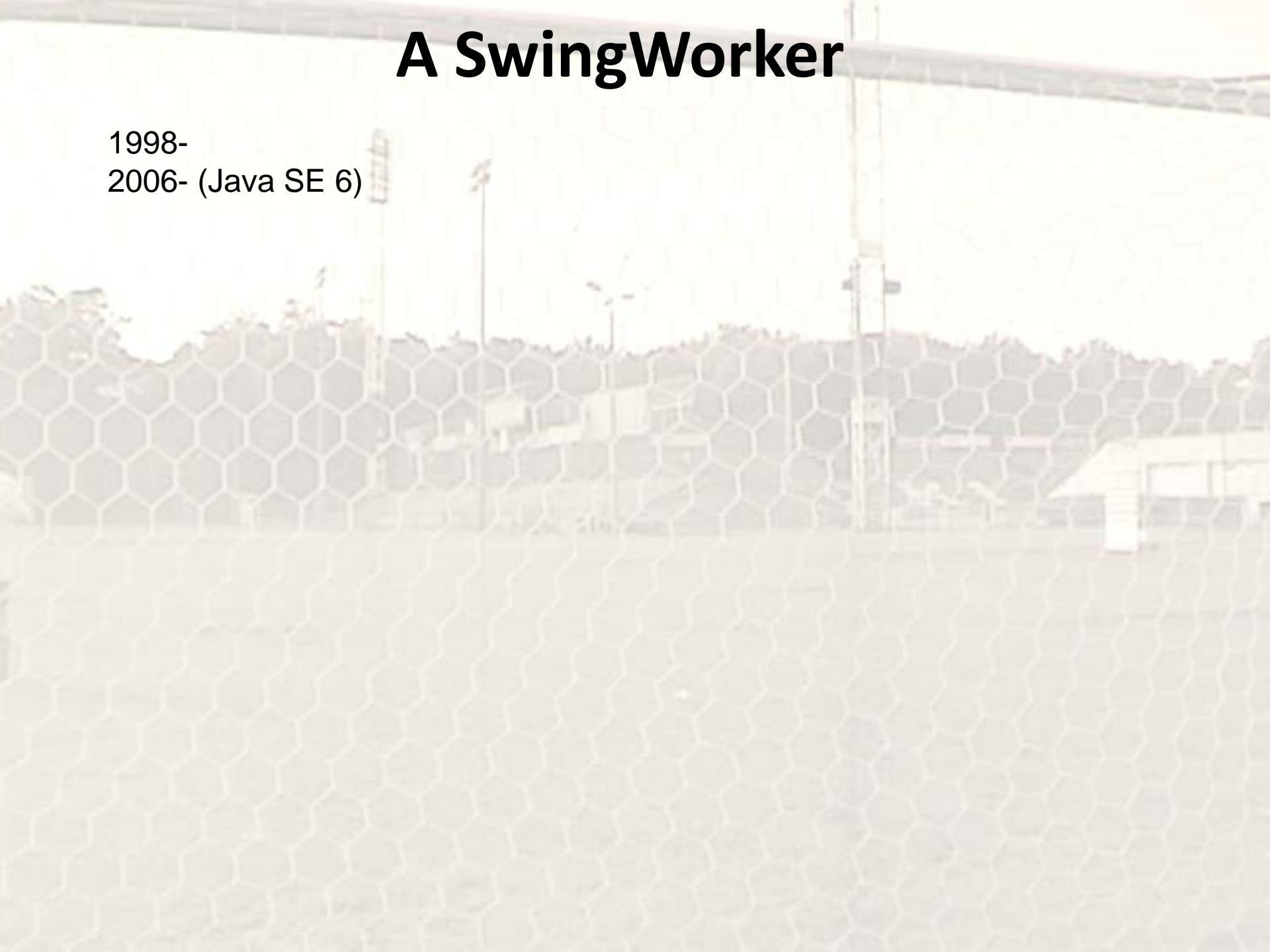
```
okButton.addActionListener(  
    new java.awt.event.ActionListener() {  
        public void actionPerformed(java.awt.event.ActionEvent  
e) {  
            kattint(e);  
        }  
    });
```

```
private void kattint(java.awt.event.ActionEvent esem) {  
    try {  
        int port = Integer.parseInt(portTextField.getText());  
        java.net.Socket socket =  
            new java.net.Socket(hosztTextField.getText(), port);  
        java.io.BufferedReader br =  
            new java.io.BufferedReader(  
                new java.io.InputStreamReader(socket.getInputStream()));  
        valaszTextArea.append(br.readLine()+"\n");  
        socket.close();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

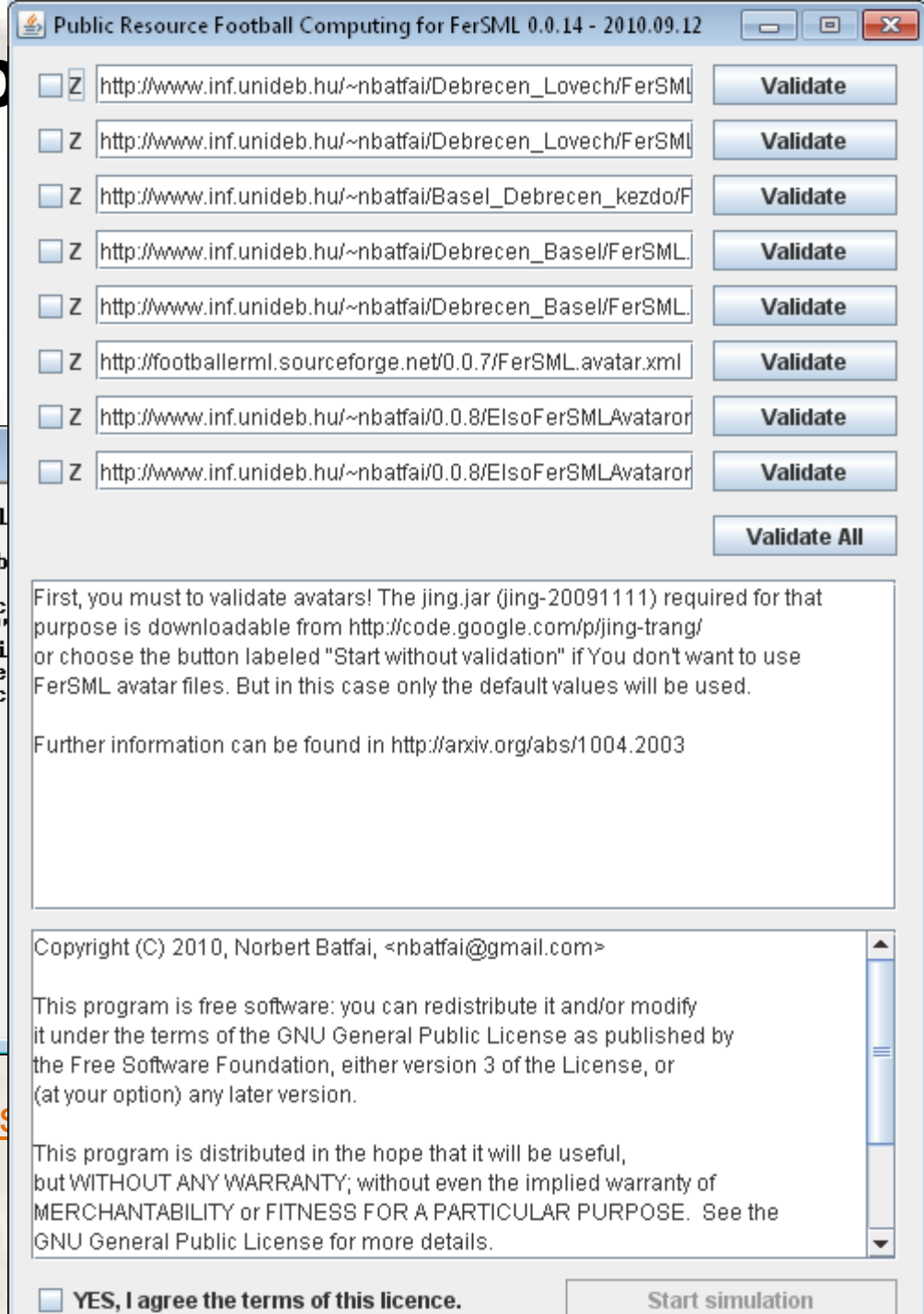
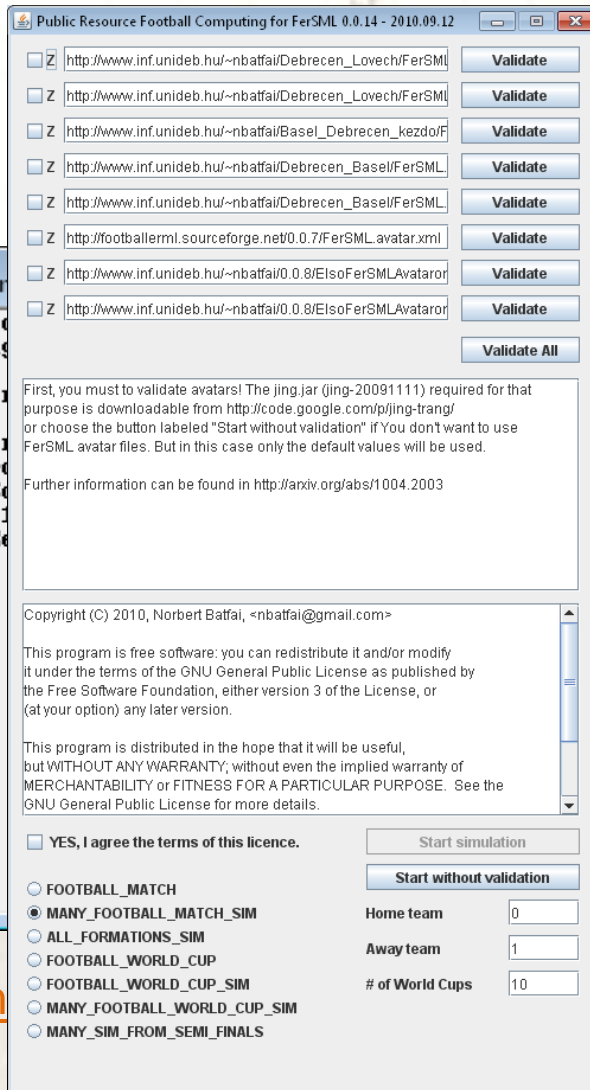
A SwingWorker

1998-

2006- (Java SE 6)



A SwingWo



SwingWorker

<http://download.oracle.com/javase/7/docs/api/index.html?javax/swing/SwingWorker.html>

PublicResourceFC\src\main\java\hu\javacska\jsfa\JSFA.java

```
package hu.javacska.jsfa;  
  
public class JSFA extends javax.swing.JFrame {  
  
    public final String FERSML_RNC_URL_STR = "http://footballerml.sourceforge.net/0.0.
```

```
private void validate(int whichTeam, String url) {  
    new hu.javacska.jsfa.ValidatingThread(whichTeam, url, FERSML_RNC_URL_STR,  
        this).execute();  
}
```

SwingWorker

<http://download.oracle.com/javase/7/docs/api/index.html?javax/swing/SwingWorker.html>

PublicResourceFC\src\main\java\hu\javacska\jsfa\ValidatingThread.java

```
package hu.javacska.jsfa;
```

```
public class ValidatingThread extends javax.swing.SwingWorker {  
//public class ValidatingThread extends javax.swing.SwingWorker<hu.javacska.xml.Avat
```

```
String xml;  
String rnc;  
JSFA callBack;  
int whichTeam;
```

```
public ValidatingThread(int whichTeam, String xml, String rnc, JSFA callBack) {  
    this.xml = xml;  
    this.rnc = rnc;  
    this.callBack = callBack;  
    this.whichTeam = whichTeam;  
}
```


SwingWorker

<http://download.oracle.com/javase/7/docs/api/index.html?javax/swing/SwingWorker.html>

Overview Package **Class** Use Tree Deprecated Index Help

Prev Class Next Class Frames No Frames

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

javax.swing

Class SwingWorker

java.lang.Object
javax.swing

Type Parameters:
T - the result type
V - the type of the computation

All Implemented Interfaces:
Runnable, Future

doInBackground

```
protected abstract T doInBackground() throws Exception
```

Computes a result, or throws an exception if unable to do so.

Note that this method must be invoked from the *SwingWorker* thread.

Note: this method must be invoked from the *SwingWorker* thread.

Returns:
the computed result

Throws:
Exception

done

```
protected void done()
```

process methods

Execute the computation and then call `done()` to indicate that the computation is complete. This method must be invoked from the *SwingWorker* thread.

See Also:
`doInBackground()`

get

```
public final T get() throws InterruptedException, ExecutionException
```

Waits if necessary for the computation to complete, and then retrieves its result.

Note: calling `get` on the *Event Dispatch Thread* blocks *all* events, including repaints, from being processed. When you want the *SwingWorker* to block on the *Event Dispatch Thread* we recommend that you call `get` on a separate thread.

SwingWorker

<http://download.oracle.com/javase/7/docs/api/index.html?javax/swing/SwingWorker.html>

PublicResourceFC\src\main\java\hu\javacska\jsfa\ValidatingThread.java

@Override

```
protected Object doInBackground() {  
    //protected hu.javacska.xml.AvatarFactory doInBackground() {  
  
    hu.javacska.xml.AvatarFactory avatarFactory = null;  
  
    hu.javacska.xml.AvatarLoader avatarLoader = new hu.javacska.xml.AvatarLoader(xml, rnc);  
  
    avatarLoader.validateAvatars();  
    avatarLoader.readAvatars();  
  
    return avatarLoader.getAvatarFactory();  
}
```

@Override

```
protected void done() {  
    try {  
        callBack.callBack(whichTeam, (hu.javacska.xml.AvatarFactory) (get()));  
        //callBack.callBack(whichTeam, get());  
    } catch (java.util.concurrent.ExecutionException ex) {  
    } catch (InterruptedException ex) {  
    }  
}
```

Nyilvános kulcsú kriptográfia

i

j

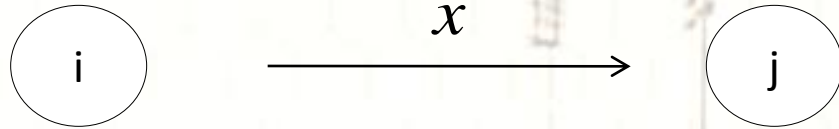
e_i , nyilvános

d_i , titkos

$x \in \{0,1\}^*$,

$f(f(x, e_i), d_i) = f(f(x, d_i), e_i) = x$

Nyilvános kulcsú kriptográfia



$$y = f(f(x, d_i), e_j)$$

$$\begin{aligned} f(f(y, d_j), e_i) &= f(f(f(f(x, d_i), e_j), d_j), e_i) = \\ &= f(f(f(K, e_j), d_j), e_i) = f(K, e_i) = x \end{aligned}$$

Java és a nagy számok

- 1) java.math.BigDecimal, tetszőleges pontosságú valós
- 2) java.math.BigInteger, tetszőleges pontosságú egész

(egyik sem tud túlcsordulni)

Since: JDK1.1
See Also:

BigInteger

```
public BigInteger(int bitLength,  
                 int certainty,  
                 Random rnd)
```

Constructs a randomly generated positive BigInteger that is probably prime, with the specified bitLength.

It is recommended that the [probablePrime](#) method be used in preference to this constructor unless there is a compelling need to specify a certainty.

Parameters:

- bitLength - bitLength of the returned BigInteger.
- certainty - a measure of the uncertainty that the caller is willing to tolerate. The probability that the new BigInteger represents a prime number will exceed $(1 - 1/2^{\text{certainty}})$. The execution time of this constructor is proportional to the value of this parameter.
- rnd - source of random bits used to select candidates to be tested for primality.

Throws:

- [ArithmeticException](#) - bitLength < 2.

See Also:

- [bitLength\(\)](#)

Construc
BigInteger (
Translat
Const
BigInte
public B
Tra
Big
zero
Par
Thu

java.math.BigInteger

Method Summary

int [getLowestSetBit](#)()

getLowestSetBit

public int [getLowestSetBit](#)()

Return bit) [BigInteger](#) [modPow](#)([BigInteger](#) exponent, [BigInteger](#) m)
Returns a [BigInteger](#) whose value is $(\text{this}^{\text{exponent}} \bmod m)$.

Return [BigInteger](#) [remainder](#)([BigInteger](#) val)
Returns a [BigInteger](#) whose value is $(\text{this} \% \text{val})$.

[BigInteger](#) [setBit](#)(int n)
Returns a [BigInteger](#) whose value is equivalent to this [BigInteger](#) with the designated bit set.

[BigInteger](#) [shiftLeft](#)(int n)

byte[] [toArray](#)()
Returns a byte array containing the two's-complement representation of this [BigInteger](#).

[String](#) [toString](#)()
Returns the decimal [String](#) representation of this [BigInteger](#).

[String](#) [toString](#)(int radix)
Returns the [String](#) representation of this [BigInteger](#) in the given radix.

static [BigInteger](#) [valueOf](#)(long val)
Returns a [BigInteger](#) whose value is equal to that of the specified long.

[BigInteger](#) [xor](#)([BigInteger](#) val)
Returns a [BigInteger](#) whose value is $(\text{this} \wedge \text{val})$.

RSA, kulcsgenerálás

```
public class RSA {  
  
    public static void main(String[] args) {  
        java.math.BigInteger p_i =  
            new java.math.BigInteger(meretBitekben, 100, new java.util.Random());  
        System.out.println(p_i);  
        java.math.BigInteger q_i =  
            new java.math.BigInteger(meretBitekben, 100, new java.util.Random());  
        System.out.println(q_i);  
        java.math.BigInteger m_i = p_i.multiply(q_i);  
        System.out.println("m_i");  
        System.out.println(m_i);  
  
        java.math.BigInteger d_i;  
        do {  
            do {  
                d_i = new java.math.BigInteger(meretBitekben, new java.util.Random());  
            } while (d_i.equals(java.math.BigInteger.ONE));  
        } while (!z_i.gcd(d_i).equals(java.math.BigInteger.ONE));  
  
        java.math.BigInteger e_i = d_i.modInverse(z_i);  
        System.out.println("e_i");  
        System.out.println(e_i);  
    }  
}
```

RSA, kulcsgenerálás

```
Output - RSA (run)
> Méret bitekben:
2100
> p_i
13214503534271267300850866132310966230764841400691224877990350143668109199585408826290501236365016749348982783353086621536261216
p_i hexa
e86f23f680954b278a96aa4362d2dal7c9ae2f23be44974544ac6939aea89bc8ad6563abc338fc681737453d7abfalld5ac832bbdb062023079c4bfeae4fe307
q_i
10073889058709146411150801643180238428780756392384569223291650612716093547053820573296232855004594410593818826678666427660987540
m_i
13312144257016866542230278797370944720027450295718729563537116613492128947974742964699042268474836410194022247670215183790303519
z_i
13312144257016866542230278797370944720027450295718729563537116613492128947974742964699042268474836410194022247670215183790303519
d_i
18197938426942526592344015962927854202588372981833403614687089488280723638609560593378980223890796301615432697577959010212886927
e_i
13067605823769374880218209013740123858214361220516841109648803954004728690422915315206808587276714914045616016126269943613143223
BUILD SUCCESSFUL (total time: 22 seconds)
```

```
Output - RSA (run)
> Méret bitekben:
2100
> p_i
138805815511884210090707246696319215160169338354719775762951984607798911084430228900202426079168910499311981171233811266294
p_i hexa
f426668f7d316c3f1f21044f1e50065a125fd423e83d7e7d79c15ae72fe4035dd6b8cc22e7c81ea862de6e07841aa48c6b21bc87d370cda77f888a2b7c
q_i
792803051389089934479058464019562723039286186279444415862864161904927500227114934789602220921529888101859505353555090648284
m_i
11004567408837287418146899622459300403630902600483139249586834151252015857387371556994844550922987163881397412050376361731
z_i
11004567408837287418146899622459300403630902600483139249586834151252015857387371556994844550922987163881397412050376361731
d_i
70172666552896242593389018666741032914529288998984203179480780964150080852319725207122265697848656391178495515765296889879
e_i
80803121689447935124185805538527125144122216803979967539361309447605029512391704727749945953487916962535920977484015305974
BUILD SUCCESSFUL (total time: 8 seconds)
```


RSA, kódolás/dekódolás

```
class Kulcs public class RSAPelda {  
  
    java.math public static void main(String[] args) {  
  
        public Ku        KulcsPar jSzereplo = new KulcsPar();  
        int mer        // i -> j  
        String tisztaSzoveg = "Hello, Vilag!";  
  
        java.ma        // kódol i  
        byte[] buffer = tisztaSzoveg.getBytes();  
        java.ma        java.math.BigInteger[] titkos = new java.math.BigInteger[buffer.length];  
  
        for (int i = 0; i < titkos.length; ++i) {  
            m = p.m        titkos[i] = new java.math.BigInteger(new byte[] {buffer[i]});  
            titkos[i] = titkos[i].modPow(jSzereplo.e, jSzereplo.m);  
        }  
        java.ma  
  
        // -----  
  
        do {  
            // dekódol j  
            do {  
                for (int i = 0; i < titkos.length; ++i) {  
                    d =        titkos[i] = titkos[i].modPow(jSzereplo.d, jSzereplo.m);  
                    } while (        buffer[i] = titkos[i].byteValue();  
                }  
            } while  
  
            System.out.println(new String(buffer));  
  
            e = d.m  
        }  
    }  
}
```

JCA, JCE

Java Cryptography Architecture, Java Cryptography Extension

- 1) Implementáció függetlenség
- 2) Algoritmus függetlenség

JCA

java.security

Csak alap titkosítás
Nincs export korlátozás
Since: 1.1

JCE

javax.crypto

Itt vannak a titkosítók
Export korlátozású
Since: 1.4

JCA meghajtóosztályok

Például: java.security.KeyPair

Constructor Summary	
Constructor and Description	
KeyPair (PublicKey publicKey, PrivateKey privateKey) Constructs a key pair from the given public key and private key.	
Method Summary	
Modifier and Type	Method and Description
PrivateKey	getPrivate () Returns a reference to the private key component of this key pair.
PublicKey	getPublic () Returns a reference to the public key component of this key pair.

```
public abstract class KeyPairGenerator extends KeyPairGeneratorSpi {
```

```
public abstract class KeyPairGeneratorSpi {
```

JCE meghajtóosztályok

Például: `javax.crypto.Cipher`

<code>static Cipher</code>	<code>getInstance(String transformation)</code> Returns a cipher object that implements the specified transformation.
<code>static Cipher</code>	<code>getInstance(String transformation, Provider provider)</code> Returns a cipher object that implements the specified transformation.
<code>static Cipher</code>	<code>getInstance(String transformation, String provider)</code> Returns a cipher object that implements the specified transformation.

Every implementation of the Java platform is required to support the following standard cipher transformations with the key sizes in parentheses:

- AES/CBC/NoPadding (128)
- AES/CBC/PKCS5Padding (128)
- AES/ECB/NoPadding (128)
- AES/ECB/PKCS5Padding (128)
- DES/CBC/NoPadding (56)
- DES/CBC/PKCS5Padding (56)
- DES/ECB/NoPadding (56)
- DES/ECB/PKCS5Padding (56)
- DESede/CBC/NoPadding (168)
- DESede/CBC/PKCS5Padding (168)
- DESede/ECB/NoPadding (168)
- DESede/ECB/PKCS5Padding (168)
- RSA/ECB/PKCS1Padding (1024, 2048)
- RSA/ECB/OAEPWithSHA-1AndMGF1Padding (1024, 2048)
- RSA/ECB/OAEPWithSHA-256AndMGF1Padding (1024, 2048)

Bouncy Castle

1 cég

2 egyetemi tanszék

539 személy

<http://www.bouncycastle.org/contributors.html>

RSA, Bouncy Castle

```
public class BCRSA {  
  
    public static void main(String[] args) {  
  
        java.security.Security.addProvider(  
            new org.bouncycastle.jce.provider.BouncyCastleProvider());  
  
        javax.crypto.Cipher cipher = null;  
  
        try {  
            cipher = javax.crypto.Cipher.getInstance("RSA/None/NoPadding", "BC");  
            System.out.println(cipher.getProvider());  
        } catch (java.security.NoSuchAlgorithmException ex) {  
            java.util.logging.Logger.getLogger(  
                BCRSA.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  
        } catch (java.security.NoSuchProviderException ex) {  
            java.util.logging.Logger.getLogger(  
                BCRSA.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  
        } catch (javax.crypto.NoSuchPaddingException ex) {  
            java.util.logging.Logger.getLogger(  
                BCRSA.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  
        }  
    }  
}
```

Output - BCRSA (run)

```
run:  
SunJCE version 1.6  
BC version 1.46  
BUILD SUCCESSFUL (total time: 1 second)
```

RSA, kódolás/dekódolás, Bouncy Castle

```
public class BCRSAPelda {

    public static void main(String[] args) {
        java.security.Security.addProvider(
            new org.bouncycastle.jce.provider.BouncyCastleProvider());

        javax.crypto.Cipher cipher = null;

        try {

            cipher = javax.c

            // i -> j
            String tisztaSzoveg = "Hello, Vilag!";

            java.security.Ke
                java.sec
            keyPairGenerator

            // kódol i
            cipher.init(javax.crypto.Cipher.ENCRYPT_MODE, eKey);

            byte[] titkos = cipher.doFinal(tisztaSzoveg.getBytes());

            java.security.Ke
            java.security.Ke
            java.security.Ke

            // -----

            // dekódol j
            cipher.init(javax.crypto.Cipher.DECRYPT_MODE, dKey);

            System.out.println(new String(cipher.doFinal(titikos)));

        }
    }
}
```



A Java fauna távolabbról

Tigrisek, musztángok, delfinek...

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-java-080904-2>

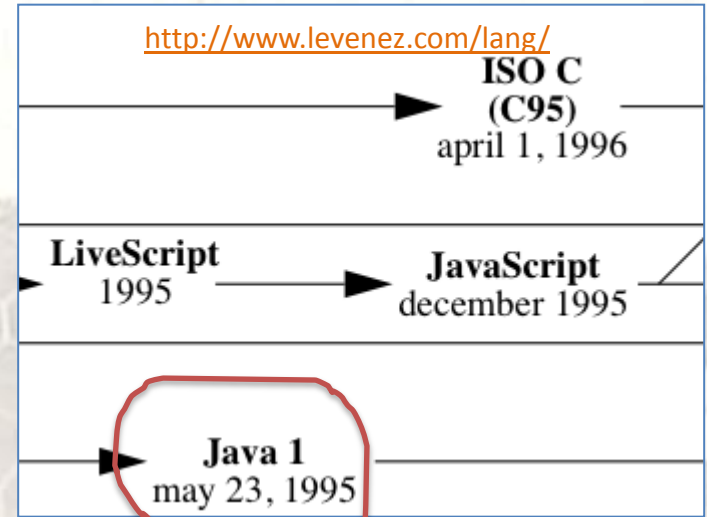
Java 1

JDK 1.0, Oak

JSDK (Java Software Development Kit)

Java 1.0.2

- 1) AWT, hordozható: Windows, Solaris
- 2) Appletek
- 3) Konténer alapú eseménykezelés, pl. `action()`, `mouseDown()` a GUI objektum egyben eseménykezelő is
- 4) ...



Tigrisek, musztángok, delfinek...

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-java-080904-2>

Java 1.1

JDK 1.1

Java 1.1

- 1) Eseménykezelés átírása **listener alapúra**, a GUI elválik az alkalmazás logikájától, minden esemény külön osztályban absztrahálva
- 2) Belső osztályok
- 3) Objektum szerializáció
- 4) RMI (Remote Method Invocation)
- 5) ...

Tigrisek, musztángok, delfinek...

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-java-080904-2>

Java 1.2

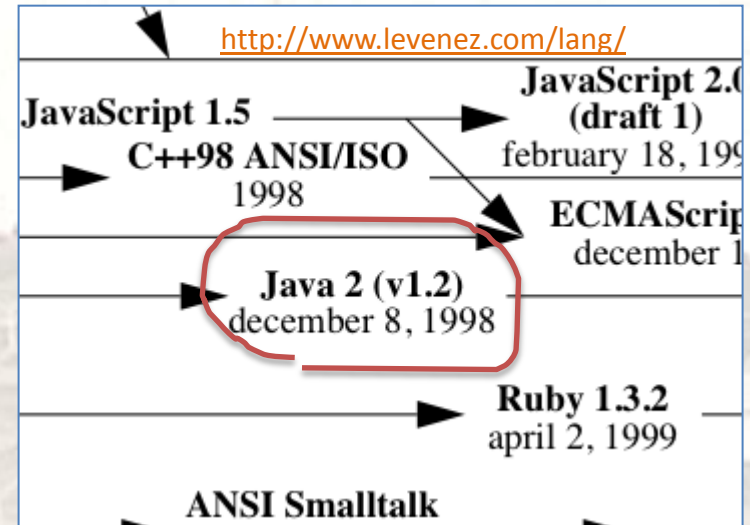
J2SE (Java 2 Platform, Standard Edition)

J2SE 1.2

Playground

J2SDK 1.2

- 1) Swing
- 2) Java IDL
- 3) JIT (Just In Time) fordítás
- 4) A strictfp kulcsszó
- 5) JFC (Java Collections Framework)
- 6) ...



J2SE, (Java 2 Standard Edition)
J2EE, (Java 2 Enterprise Edition)
J2ME, (Java 2 Micro Edition)

Tigrisek, musztángok, delfinek...

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-java-080904-2>

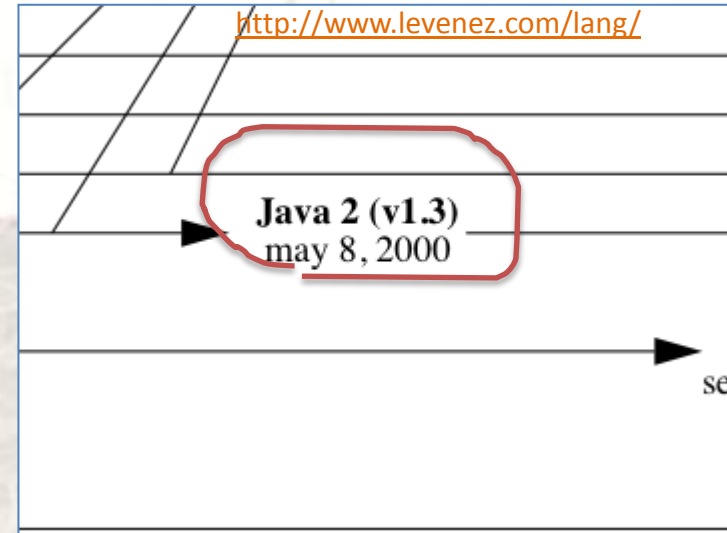
Java 1.3

J2SE 1.3

Kestrel

J2SDK 1.3

- 1) HotSpot
- 2) Java Sound API
- 3) Java 2D
- 4) ...



Tigrisek, musztángok, delfinek...

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-java-080904-2>

Java 1.4

J2SE 1.4

Merlin

- 1) XML feldolgozás, JAXP
- 2) NIO API
- 3) Image IO API
- 4) JCE

http://www.levenez.com/lang/
Java 2 (v1.4) early access february 6, 2002
Ruby 1.6.5 mber 19, 2001

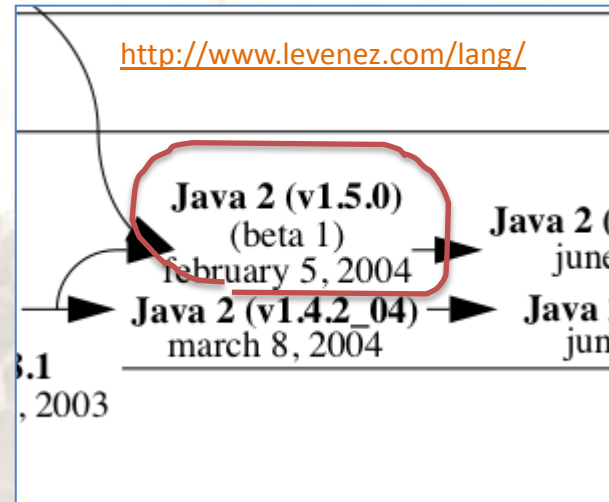
Tigrisek, musztángok, delfinek...

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-java-080904-2>

Java 1.5

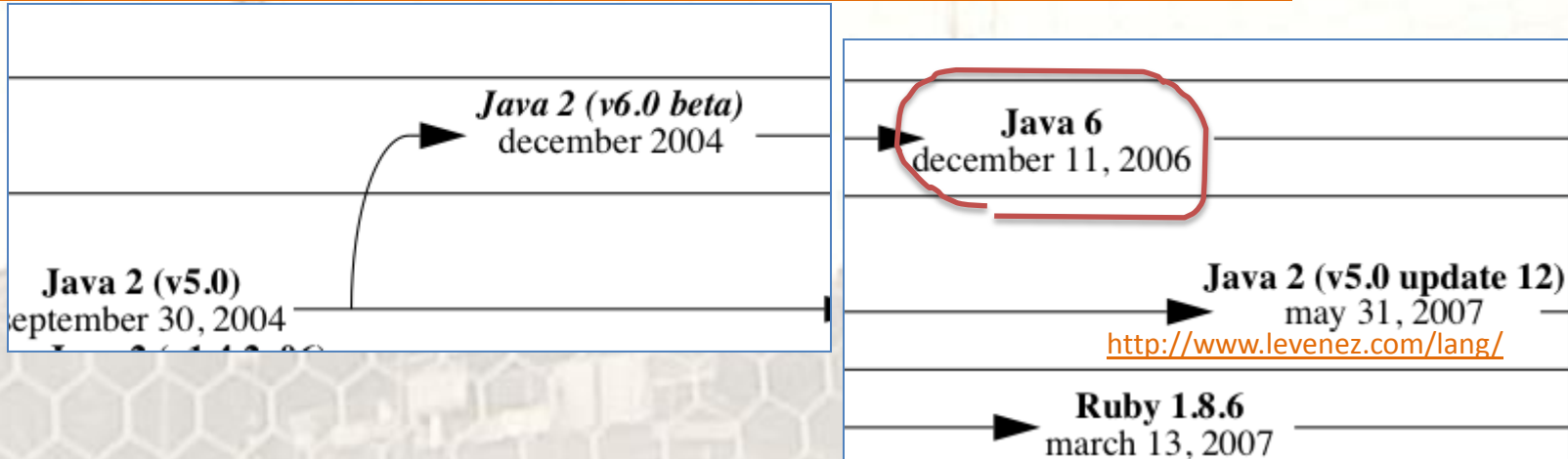
J2SE 5.0

- 1) megjelent a generikus
- 2) egy újfajta for ciklus: az iteráló ciklus
- 3) immár a primitív Java típusok automatikusan csomagolódnak be és vissza csomagoló osztályaikba
- 4) megjelent a felsorolásos típus
- 5) lehetőség nyílt változó paraméterszámú függvények írására
- 6) és statikus tagok olyan importjára, ami elhagyhatóvá teszi a tagra vonatkozó osztálynév minősítést.



Tigrisek, musztángok, delfinek...

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-java-080904-2>



Java 6

Java SE 6

- 1) SwingWorker
- 2) GUI/értékesítési terület, indítóképernyő
- 3) ...

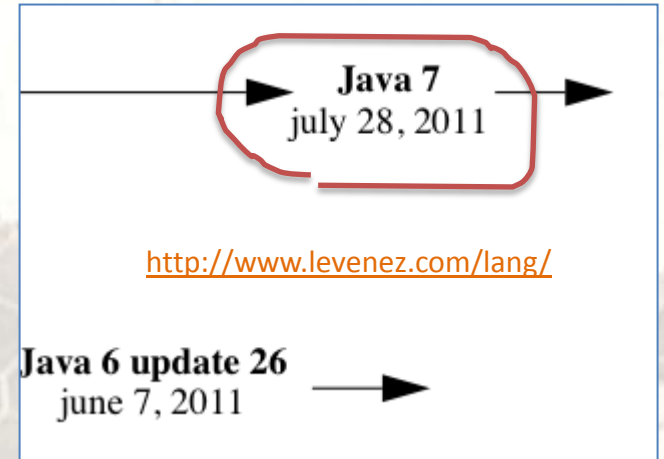
Tigrisek, musztángok, **delfinek**...

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-java-080904-2>

Java 7

Java SE 7

- 1) NIO 2
- 2) ...





A Java fauna közelebbről

Tigrisek, musztángok, delfinek...

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-java-080904-2>

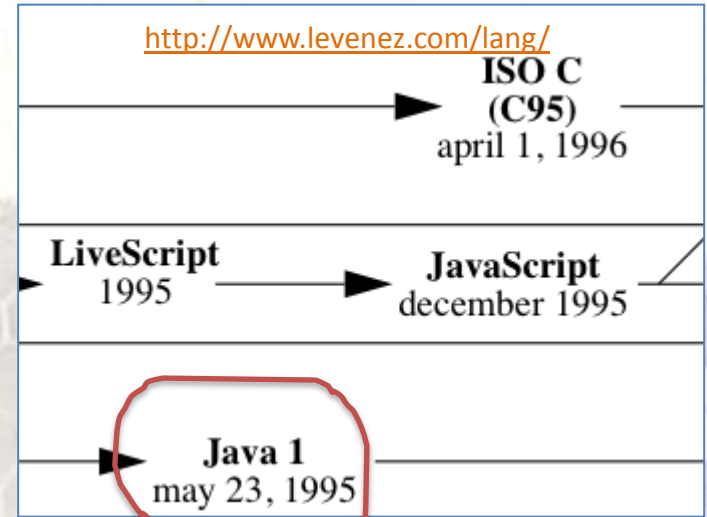
Java 1

JDK 1.0, Oak

JSDK (Java Software Development Kit)

Java 1.0.2

- 1) AWT, hordozható: Windows, Solaris
- 2) Appletek
- 3) Konténer alapú eseménykezelés, pl. `action()`, `mouseDown()` a GUI objektum egyben eseménykezelő is
- 4) ...



Tigrisek, musztángok, delfinek...

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-java-080904-2>

Java 1.1

JDK 1.1

Java 1.1

- 1) Eseménykezelés átírása **listener alapúra**, a GUI elválik az alkalmazás logikájától, minden esemény külön osztályban absztrahálva
- 2) Belső osztályok
- 3) Objektum szerializáció
- 4) RMI (Remote Method Invocation)
- 5) ...

Eseménykezelés

```
public class TeljesKepernyo extends java.awt.Frame {
...
    public TeljesKepernyo(
        ...
        addMouseListener(new java.awt.event.MouseAdapter() {

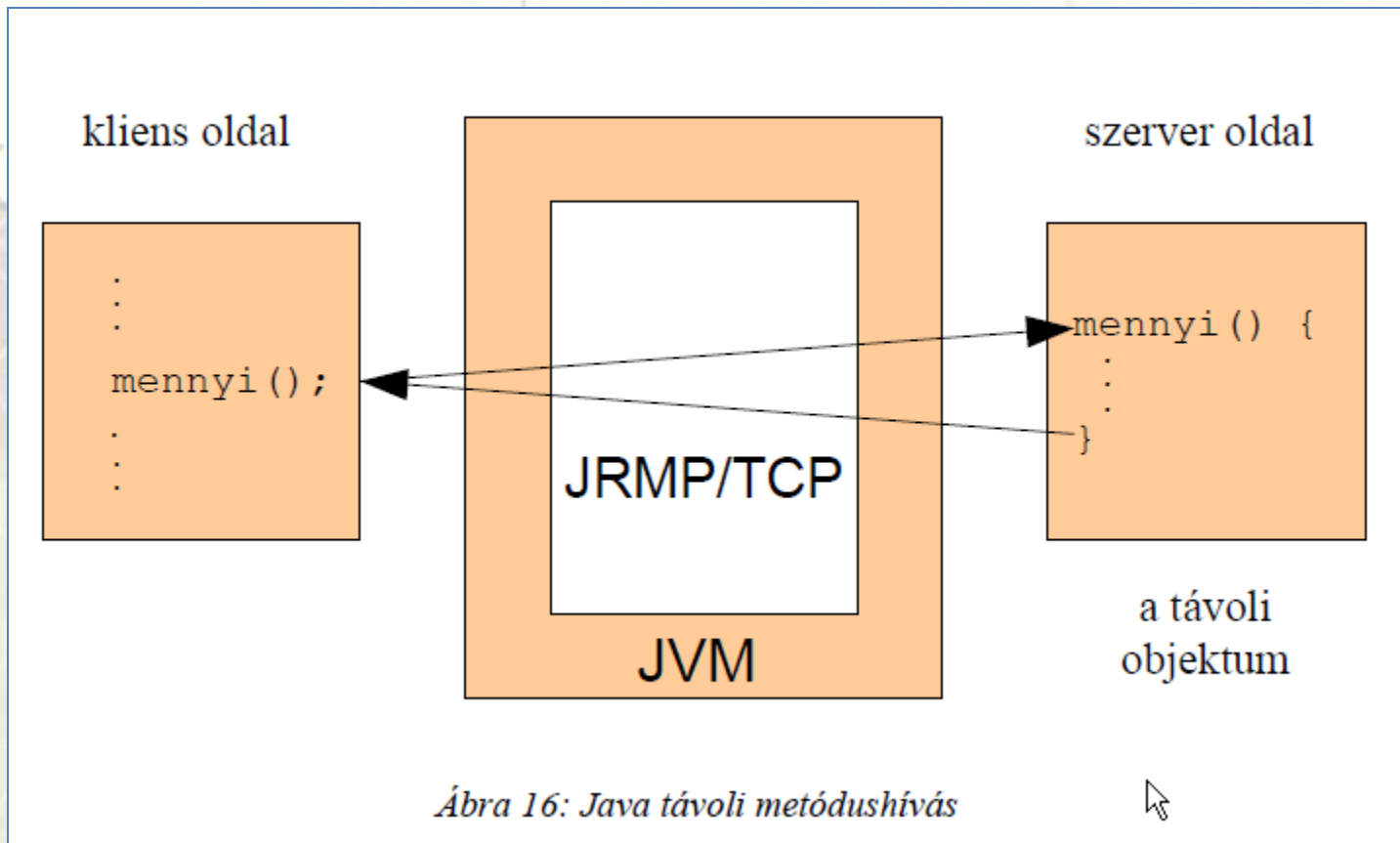
            public void mousePressed(java.awt.event.MouseEvent e) {
                setVisible(false);
                TeljesKepernyo.this.graphicsDevice.setFullScreenWindow(null);
                System.exit(0);
            }
        })
    }
}

package java.awt.event;
jdk1.1.8\src\java\awt\event\MouseMotionAdapter.java

/**
 * The adapter which receives mouse motion events.
 * The methods in this class are empty; this class is provided as a
 * convenience for easily creating listeners by extending this class
 * and overriding only the methods of interest.
 *
 * @version 1.6 12/10/01
 * @author Amy Fowler
 */
public abstract class MouseMotionAdapter implements MouseMotionListener {
    public void mouseDragged(MouseEvent e) {}
    public void mouseMoved(MouseEvent e) {}
}
}
```

RMI

PP 161



RMI

```
C:\ Parancssor  
C:\EA\rmi>javac Ido.java IdoImpl.java IdoSzerver.java IdoKliens.java  
C:\EA\rmi>rmic IdoImpl  
C:\EA\rmi>_
```

```
C:\ Parancssor - rmiregistry  
C:\EA>rmiregistry  
-
```

```
C:\ Parancssor - java -Djava.rmi.server.codebase=file:///c:/EA/rmi/ IdoSzerver  
C:\EA\rmi>javac Ido.java IdoImpl.java IdoSzerver.java IdoKliens.java  
C:\EA\rmi>rmic IdoImpl  
C:\EA\rmi>java -Djava.rmi.server.codebase=file:///c:/EA/rmi/ IdoSzerver  
89.133.251.39  
-
```

```
- □ ×
```

RMI

```
public interface Ido extends java.rmi.Remote {  
    public String mennyi() throws java.rmi.RemoteException;  
}
```

```
public class IdoImpl implements Ido {  
    public String mennyi() {  
        try {  
            System.out.println(java.rmi.server.RemoteServer.getClientHost());  
        } catch (java.rmi.server.ServerNotActiveException e) {  
            e.printStackTrace();  
        }  
        return new java.util.Date().toString();  
    }  
}
```

RMI

```
public class IdoSzerver {  
  
    public IdoSzerver() {  
        try {  
            IdoImpl idoImpl = new IdoImpl();  
            Ido ido = (Ido) java.rmi.server.UnicastRemoteObject.exportObject(idoImpl);  
            java.rmi.Naming.rebind("MennyiAzIdo", ido);  
        } catch (java.rmi.RemoteException re) {  
            re.printStackTrace();  
        }  
    }  
  
    public static void main(String args[]) {  
        new IdoSzerver();  
    }  
}
```

Parancssor - java -Djava.rmi.server.codebase=file:///c:/EA/rmi/ IdoSzerver


C:\EA\rmi>javac Ido.java IdoImpl.java IdoSzerver.java IdoKliens.java

C:\EA\rmi>rmic IdoImpl

C:\EA\rmi>java -Djava.rmi.server.codebase=file:///c:/EA/rmi/ IdoSzerver
89.133.251.39

RMI

```
public class IdoKliens {  
  
    public static void main(String args[]) {  
        try {  
            Ido ido = (Ido) java.rmi.Naming.lookup("MennyiAzIdo");  
            System.out.println(ido.mennyi());  
        } catch (java.net.MalformedURLException mu) {  
            mu.printStackTrace();  
        } catch (java.rmi.NotBoundException be) {  
            be.printStackTrace();  
        } catch (java.rmi.RemoteException re) {  
            re.printStackTrace();  
        }  
    }  
}
```



The screenshot shows a Windows command prompt window with the title bar 'Parancssor'. The command prompt displays the following text:

```
C:\EA\rmi>java IdoKliens  
Sun Oct 09 16:26:54 CEST 2011  
C:\EA\rmi>
```

Tigrisek, musztángok, delfinek...

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-java-080904-2>

Java 1.2

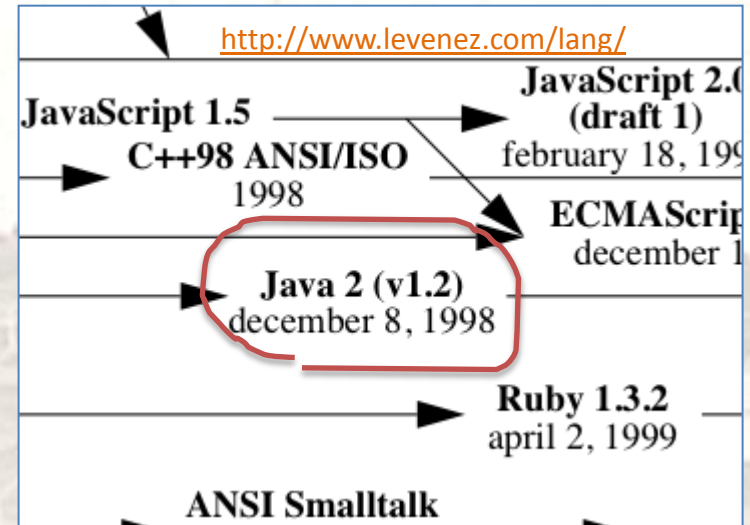
J2SE (Java 2 Platform, Standard Edition)

J2SE 1.2

Playground

J2SDK 1.2

- 1) Swing
- 2) Java IDL
- 3) JIT (Just In Time) fordítás
- 4) A strictfp kulcsszó
- 5) JFC (Java Collections Framework)
- 6) ...



J2SE, (Java 2 Standard Edition)
J2EE, (Java 2 Enterprise Edition)
J2ME, (Java 2 Micro Edition)

Kollekciók



Kollekciók



Kollekciók



Tigrisek, musztángok, delfinek...

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-java-080904-2>

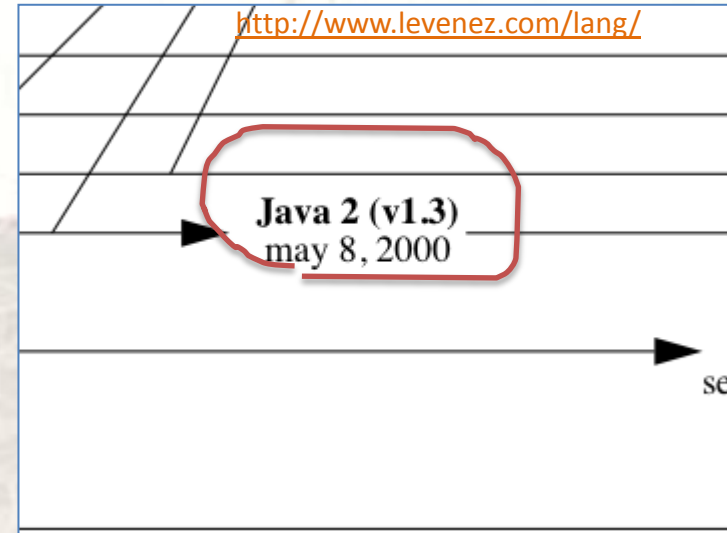
Java 1.3

J2SE 1.3

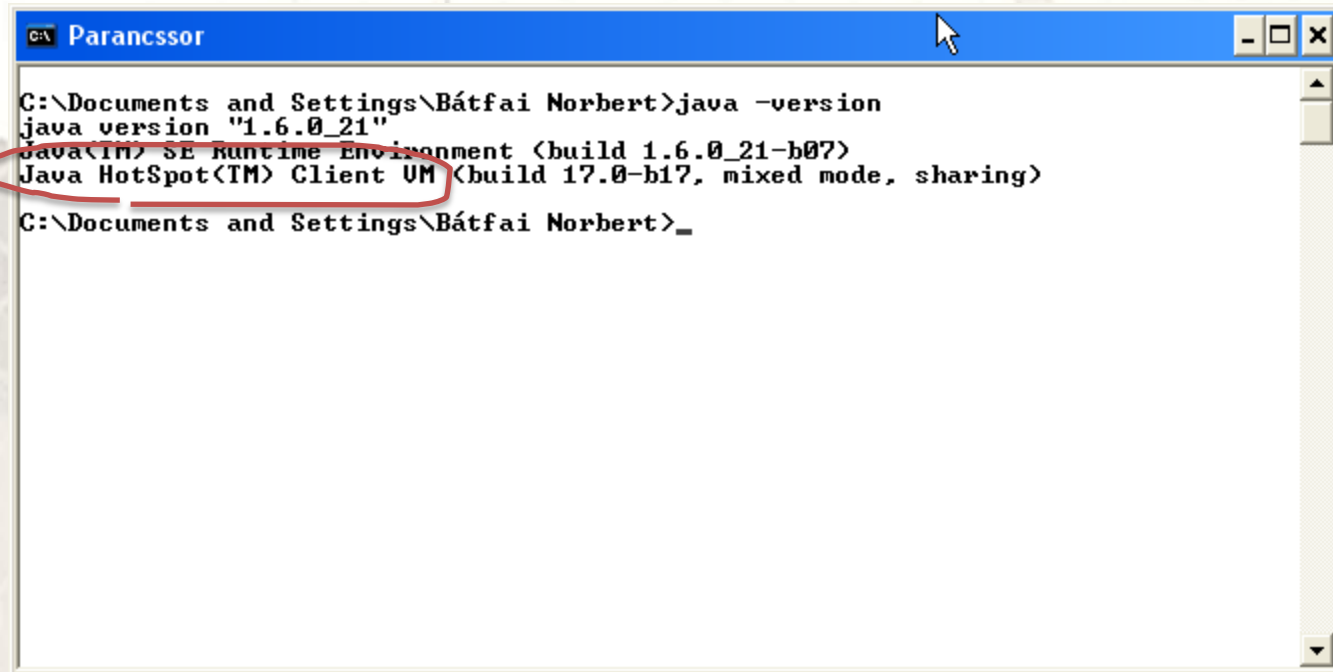
Kestrel

J2SDK 1.3

- 1) HotSpot
- 2) Java Sound API
- 3) Java 2D
- 4) ...

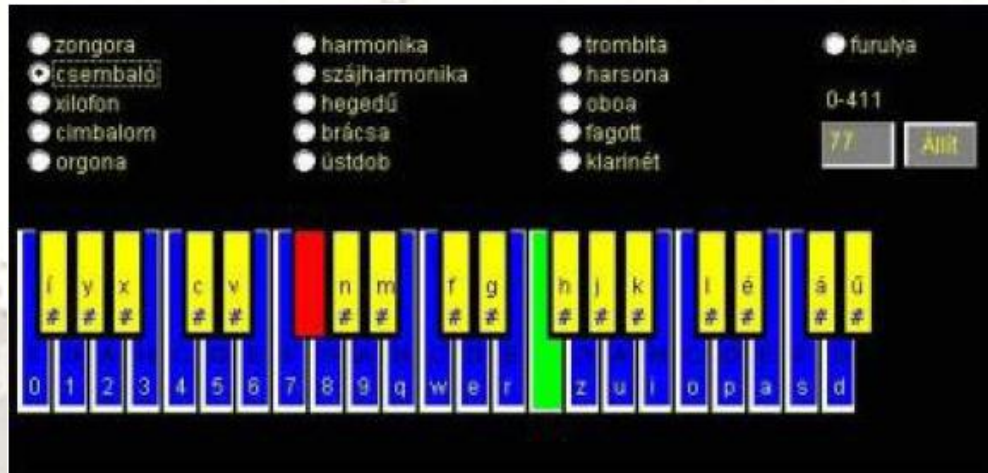


HotSpot



```
C:\> Parancssor
C:\Documents and Settings\Bátfai Norbert>java -version
java version "1.6.0_21"
Java(TM) SE Runtime Environment (build 1.6.0_21-b07)
Java HotSpot(TM) Client VM (build 17.0-b17, mixed mode, sharing)
C:\Documents and Settings\Bátfai Norbert>_
```

Java Sound



<http://javacska.lib.unideb.hu/seged/szakkor-Midi.pdf>



<http://sourceforge.net/projects/javacska/>

Tigrisek, musztángok, delfinek...

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-java-080904-2>

Java 1.4

J2SE 1.4

Merlin

- 1) XML feldolgozás, JAXP
- 2) NIO API
- 3) Image IO API
- 4) JCE

http://www.levenez.com/lang/
Java 2 (v1.4) early access february 6, 2002
Ruby 1.6.5 mber 19, 2001

NIO



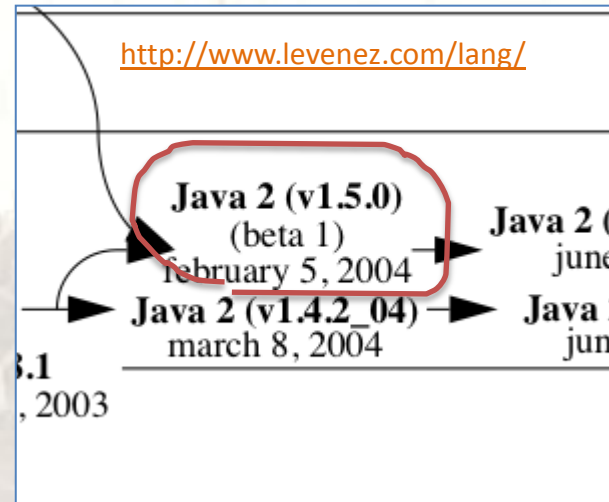
Tigrisek, musztángok, delfinek...

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-java-080904-2>

Java 1.5

J2SE 5.0

- 1) megjelent a generikus
- 2) egy újfajta for ciklus: az iteráló ciklus
- 3) immár a primitív Java típusok automatikusan csomagolódnak be és vissza csomagoló osztályaikba
- 4) megjelent a felsorolásos típus
- 5) lehetőség nyílt változó paraméterszámú függvények írására
- 6) és statikus tagok olyan importjára, ami elhagyhatóvá teszi a tagra vonatkozó osztálynév minősítést.



Generikus

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-1-tigris-080904>

```
protected Kincs [] kincsek;  
protected Szörny [] szörnyek;  
  
... ->  
  
protected java.util.List<Kincs> kincsek;  
protected java.util.List<Szörny> szörnyek;
```

```
public void nyomtat1(java.util.List<?> lista) {  
    for(Object objektum: lista)  
        System.out.println(objektum);  
}  
public void nyomtat2(java.util.List<? extends Szereplő> lista) {  
    for(Szereplő szereplő: lista)  
        System.out.println(szereplő);  
}  
public <E> void nyomtat3(java.util.List<E> lista) {  
    for(E e: lista)  
        System.out.println(e);  
}
```

javattanitok.labirintus.GenerikusLabirintus

Az iteráló ciklus

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-1-tigris-080904>

```
for(Szörny szörny: szörnyek)
    for(Kincs kincs: kincsek)
        if(kincs.megtalált(szörny))
            szörny.megtaláltam(kincs);
```

```
boolean vanKincs(int sor, int oszlop) {

    boolean van = false;

    for(int i=0; i<kincsek.length; ++i)
        if(sor == kincsek[i].sor()
            && oszlop == kincsek[i].oszlop()
            && !kincsek[i].megtalálva()) {
            van = true;
            break;
        }

    return van;
}
```

```
boolean vanKincs(int sor, int oszlop) {

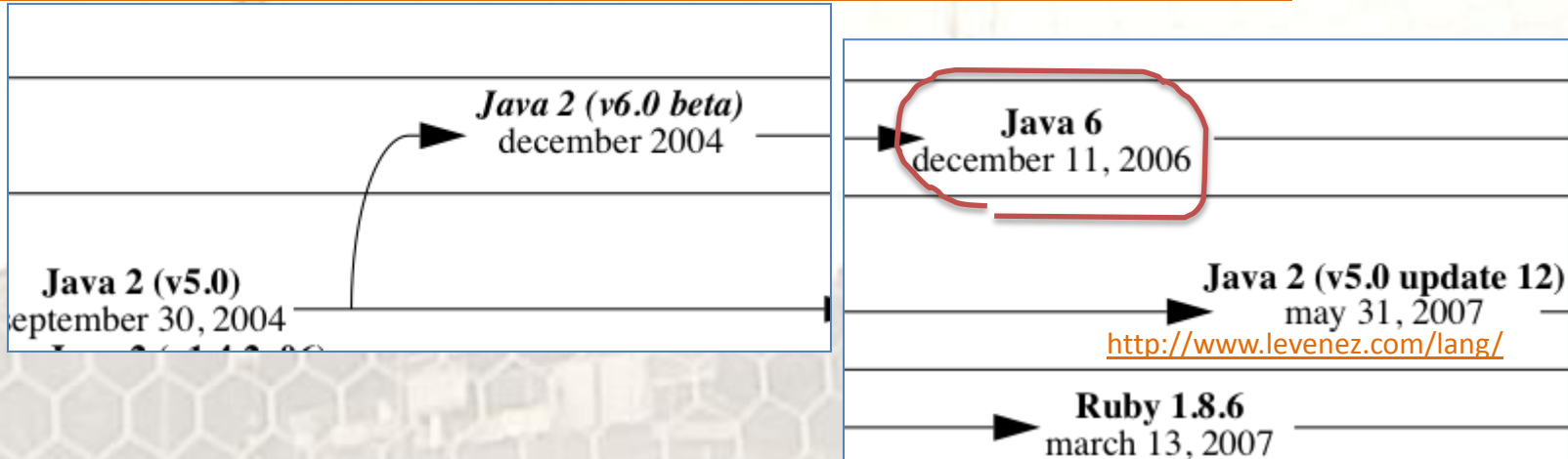
    boolean van = false;

    for(Kincs kincs: kincsek)
        if(sor == kincs.sor()
            && oszlop == kincs.oszlop()
            && !kincs.megtalálva()) {
            van = true;
            break;
        }

    return van;
}
```

Tigrisek, musztángok, delfinek...

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-java-080904-2>



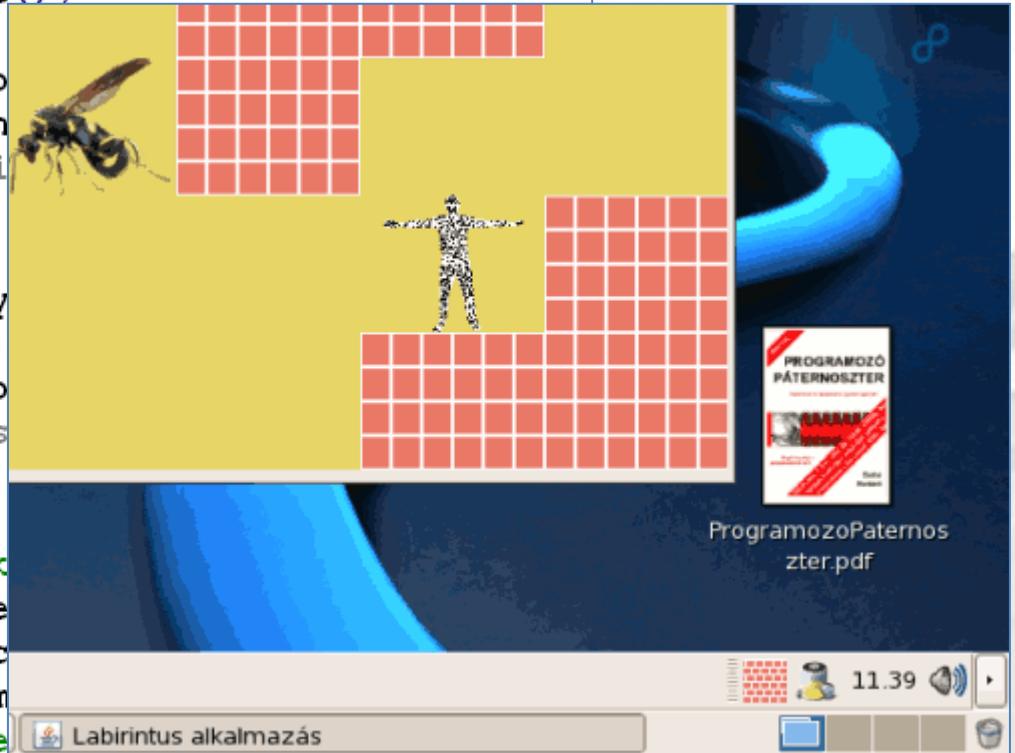
Java 6

Java SE 6

- 1) SwingWorker
- 2) GUI/értékesítési terület, indítóképernyő
- 3) ...

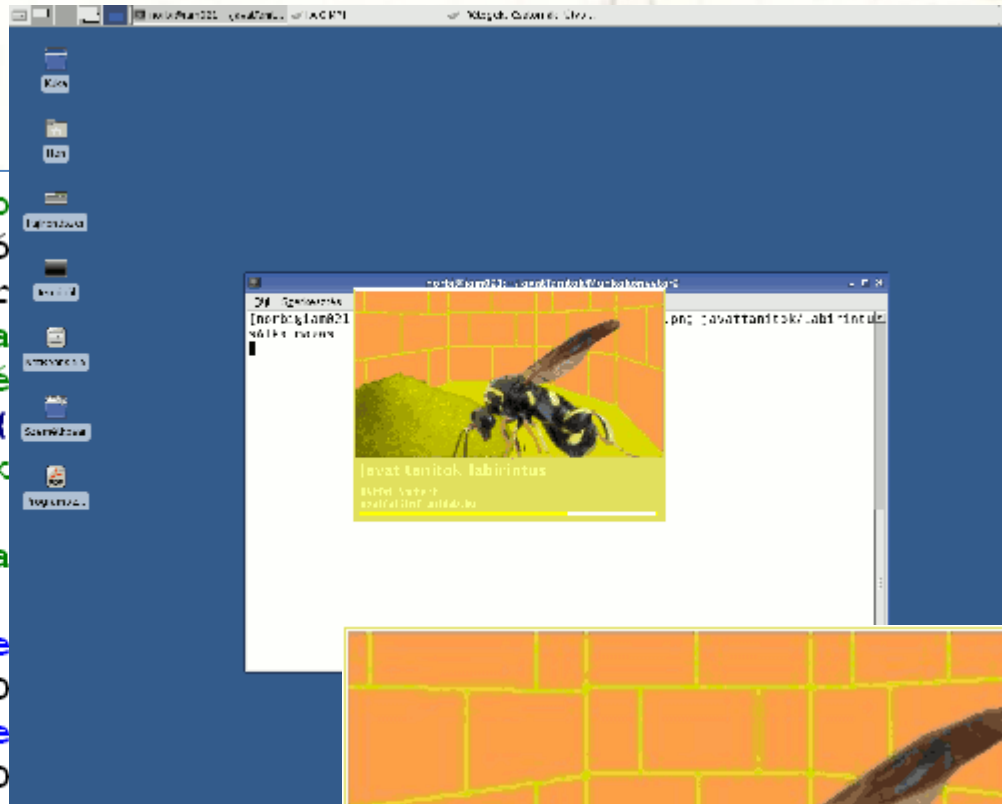
GUI/értesítési terület,

```
// Ha a rendszerben támogatott az értesítési terület
if (java.awt.SystemTray.isSupported()) {
    // akkor egy kis 16x16 pixel méretű
    java.awt.Image ikonKép = new javax.swing.ImageIcon
        ("ikon.png").getImage();
    // értesítő ikont
    java.awt.TrayIcon értesítőIko
        new java.awt.TrayIcon
            "Javat tanítok labiri
    // most mi is kiteszünk
    try {
        java.awt.SystemTray.getSy
            add(értesítőIkon)
    } catch (java.awt.AWTEExceptio
        System.out.println("Értes
            + e);
    }
    // amire ha duplán kattintunk
    értesítőIkon.addActionListener
        new java.awt.event.Ac
    public void actionPerformed
        // például szorult he
        // hőst, visszadobjuk a kiindulási pozíciójába
        hős.sor(9);
        hős.oszlop(0);
        repaint();
    }
}
});
}
```



GUI/indítóképernyő

```
// Hozzáférünk az indító kép  
java.awt.SplashScreen indító  
    = java.awt.SplashScr  
// Ha sikerül, azaz, ha java  
// formában indítottuk és mé  
if(indítóKépernyő != null) {  
    // akkor rajzolni fogunk  
    java.awt.Graphics2D g =  
    // némi szöveget, amit a  
    // ráírhattunk volna  
    java.awt.Font font1 = ne  
    java.awt.Font font2 = ne  
    java.awt.Font font3 = ne  
    g.setFont(font1);  
    // világosabb sárgával  
    g.setColor(new java.awt.Color(250, 255,  
    g.drawString("Javat tanítok labirintus"
```



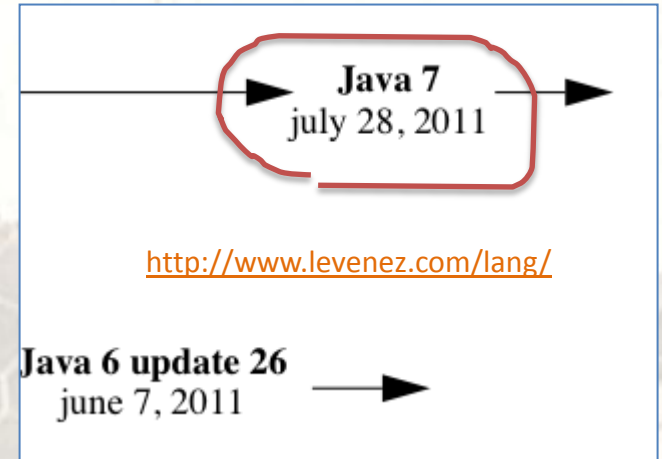
Tigrisek, musztángok, **delfinek**...

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-java-080904-2>

Java 7

Java SE 7

- 1) NIO 2
- 2) ...



Labor – a bedobás

```
package hu.fersml.aranyfc;

public class Jatekos extends hu.fersml.aranyfc.JatekosAdapter {

    ...

    protected boolean bedobas = false;
    protected boolean szoglet = false;
    protected boolean szabad = false;
    protected boolean kirugas = false;
```

```
@Override
public void preInfo() {

    super.preInfo();
    latomAFocit = false;
    latomASzelet = false;
    latomASajatKaput = false;
    latomAMasikKaput = false;

}
```

Labor – a bedobás

```
@Override
public void postInfo() {
    if (bedobas) {
        bedobashoz();
    } else if (szabad) {
        szabadhoz();
    } else if (szoglet) {
        szoglethez();
    }
}
```

```
protected void jatekban() {
    bedobas = false;
    szoglet = false;
    szabad = false;
    kirugas = false;
}
```

```
@Override
public void infoHearPlayMode(atan.model.enums.PlayMode playMode) {
    ...

    case KICK_IN_OWN:
        bedobas = true;
        break;
    case PLAY_ON:
        jatekban();
        break;
}
```

Labor – a bedobás

```
protected void bedobashoz() {  
  
    if (latomAFocit) {  
  
        if (distanceFoci < 0.9) {  
            bedobas();  
        } else {  
            getPlayer().turn(directionFoci);  
            getPlayer().dash(100);  
        }  
  
    } else {  
  
        getPlayer().turnNeck(25);  
    }  
}
```

```
protected void bedobas() {  
  
    double bedobasSzoge = -180.0 + directionFoci;  
    if (directionFoci < 0.0) {  
        bedobasSzoge = 180.0 + directionFoci;  
    }  
  
    logger.info("BEDOBAS "  
                + getPlayer().getNumber()  
                + " foci tavolsaga = " + distanceFoci  
                + " iranya = " + directionFoci  
                + " bedobas iranya = " + bedobasSzoge);  
  
    getPlayer().kick(45, bedobasSzoge);  
    getPlayer().turn(bedobasSzoge);  
}
```

Laborkártyák

Magyarázd meg az alábbi szituációt!

```
>javac javattanitok\LabirintusVilág.java
```

```
>java javattanitok.LabirintusVilág labirintus.txt
```

```
javattanitok.labirintus.RosszLabirintusKivétel:
```

```
Nincs meg a fájl: java.io.FileNotFoundException: labirintus.txt
```

```
(The system cannot find the file specified)
```

```
at javattanitok.labirintus.Labirintus.<init>(Labirintus.java:234)
```

```
232         } catch (java.io.FileNotFoundException e1) {  
233  
234             throw new RosszLabirintusKivétel("Nincs meg a fájl: " + e1);
```

```
at javattanitok.LabirintusVilág.<init>(LabirintusVilág.java:55)
```

```
55     labirintus = new Labirintus(labirintusFájlNév);
```

```
at javattanitok.LabirintusVilág.main(LabirintusVilág.java:149)
```

```
149         new LabirintusVilág(args[0]);
```

```
try {  
    new LabirintusVilág(args[0]);  
} catch (RosszLabirintusKivétel rosszLabirintusKivétel) {  
  
    //System.out.println(rosszLabirintusKivétel);  
    rosszLabirintusKivétel.printStackTrace();
```

Laborkártyák

Lefordul a „LabirintusVilág” az alábbi módosítással? Ismertesd a szituációt!

```
232     } catch (java.io.FileNotFoundException e1) {
233
234         throw new RosszLabirintusKivétel("Nincs meg a fájl: " + e1);
235
236     } catch (java.io.IOException e2) {
237
238         throw new RosszLabirintusKivétel("IO kivétel történt: "+e2);
239         /*
240     } catch (java.util.NoSuchElementException e3) {
241
242         throw new RosszLabirintusKivétel("Nem jó a labirintus szerkezete: "
243             +e3);
244         */
245     } finally {
246
247         if (szovegesCsatorna != null) {
248
249             try{
250                 szovegesCsatorna.close();
251             } catch (Exception e) {}
252
253         }
```

Laborkártyák

Lefordul a „LabirintusVilág” az alábbi módosítással? Ismertesd a szituációt!

```
232     } catch(java.io.FileNotFoundException e1) {
233
234         throw new RosszLabirintusKivétel("Nincs meg a fájl: " + e1);
235         /*
236     } catch(java.io.IOException e2) {
237
238         throw new RosszLabirintusKivétel("IO kivétel történt: "+e2);
239         */
240     } catch(java.util.NoSuchElementException e3) {
241
242         throw new RosszLabirintusKivétel("Nem jó a labirintus szerkezete: "
243             +e3);
244
245     } finally {
246
247         if(szövegesCsatorna != null) {
248
249             try{
250                 szövegesCsatorna.close();
251             } catch(Exception e) {}
252
253         }
254
```

Laborkártyák

Lefordul a „LabirintusVilág” az alábbi módosítással? Ismertesd a szituációt!

```
231     /*
232     } catch(java.io.FileNotFoundException e1) {
233
234         throw new RosszLabirintusKivétel("Nincs meg a fájl: " + e1);
235     */
236     } catch(java.io.IOException e2) {
237
238         throw new RosszLabirintusKivétel("IO kivétel történt: "+e2);
239
240     } catch(java.util.NoSuchElementException e3) {
241
242         throw new RosszLabirintusKivétel("Nem jó a labirintus szerkezete: "
243             +e3);
244
245     } finally {
246
247         if(szövegesCsatorna != null) {
248
249             try{
250                 szövegesCsatorna.close();
251             } catch(Exception e) {}
252         }
```


Laborkártyák

Lefordul a „LabirintusVilág” az alábbi módosítással? Ismertesd a szituációt!

```
231     /*
232     } catch(java.io.FileNotFoundException e1) {
233
234         throw new RosszLabirintusKivétel("Nincs meg a fájl: " + e1);
235
236     } catch(java.io.IOException e2) {
237
238         throw new RosszLabirintusKivétel("IO kivétel történt: "+e2);
239     /*
240     } catch(java.util.NoSuchElementException e3) {
241
242         throw new RosszLabirintusKivétel("Nem jó a labirintus szerkezete: "
243             +e3);
244
245     } finally {
246
247         if(szovegesCsatorna != null) {
248
249             try{
250                 szovegesCsatorna.close();
251             } catch(Exception e) {}
252
253         }
```

Laborkártyák

Mi történik, ha a <https://sourceforge.net/projects/footballerml/> ValidatingThread osztályában elvégezzük az alábbi 3 módosítást (régi kommentetve, új alatta)?

```
package hu.javacska.jsfa;  
  
public class ValidatingThread extends javax.swing.SwingWorker {  
//public class ValidatingThread extends javax.swing.SwingWorker<hu.javacska.xml.AvatarFactory, Object> {
```

```
@Override  
protected Object doInBackground() {  
//protected hu.javacska.xml.AvatarFactory doInBackground() {
```

hu Most csak kipróbálni kell (hogy szokja a szemetek), a helyes választ majd a 4. prezi után
várom itt, ezek közül:

- 1) Nem fordul le, mert nincs generikus csak az 1.6-os Javában
- 2) Nem fordul le, mert a SwingWorker nem része a standard Java API-nak
- 3) Az 1.6 óta a SwingWorker része a standard Java API-nak, de az osztály megváltozott:
construct() és finished()
- 4) Nem fordul le más okból
- 5) Ugyanúgy lefordul
- 6) Ugyanúgy lefordul, de deprecated , mert a SwingWorker construct() és finished() módszerei a használandóak
(<http://java.sun.com/products/jfc/tsc/articles/threads/threads2.html>)
- 7) Már az eredeti sem fordulhatott le, mert annotációk nem használhatóak a Java forrásállományokban, csak XML-ben

Laborkártyák- A kliens-szerver modell, most a kivételkezelés alkalmazása

```
public class KapuSzkenner {  
  
    public static void main(String[] args) {  
        for (int i = 0; i < 1024; ++i) {  
            try {  
                java.net.Socket socket = new java.net.Socket("www.inf.unideb.hu", i);  
                System.out.println(i + " figyel");  
                socket.close();  
            } catch (Exception e) {  
                System.out.println(i + " nem figyel");  
            }  
        }  
    }  
}
```

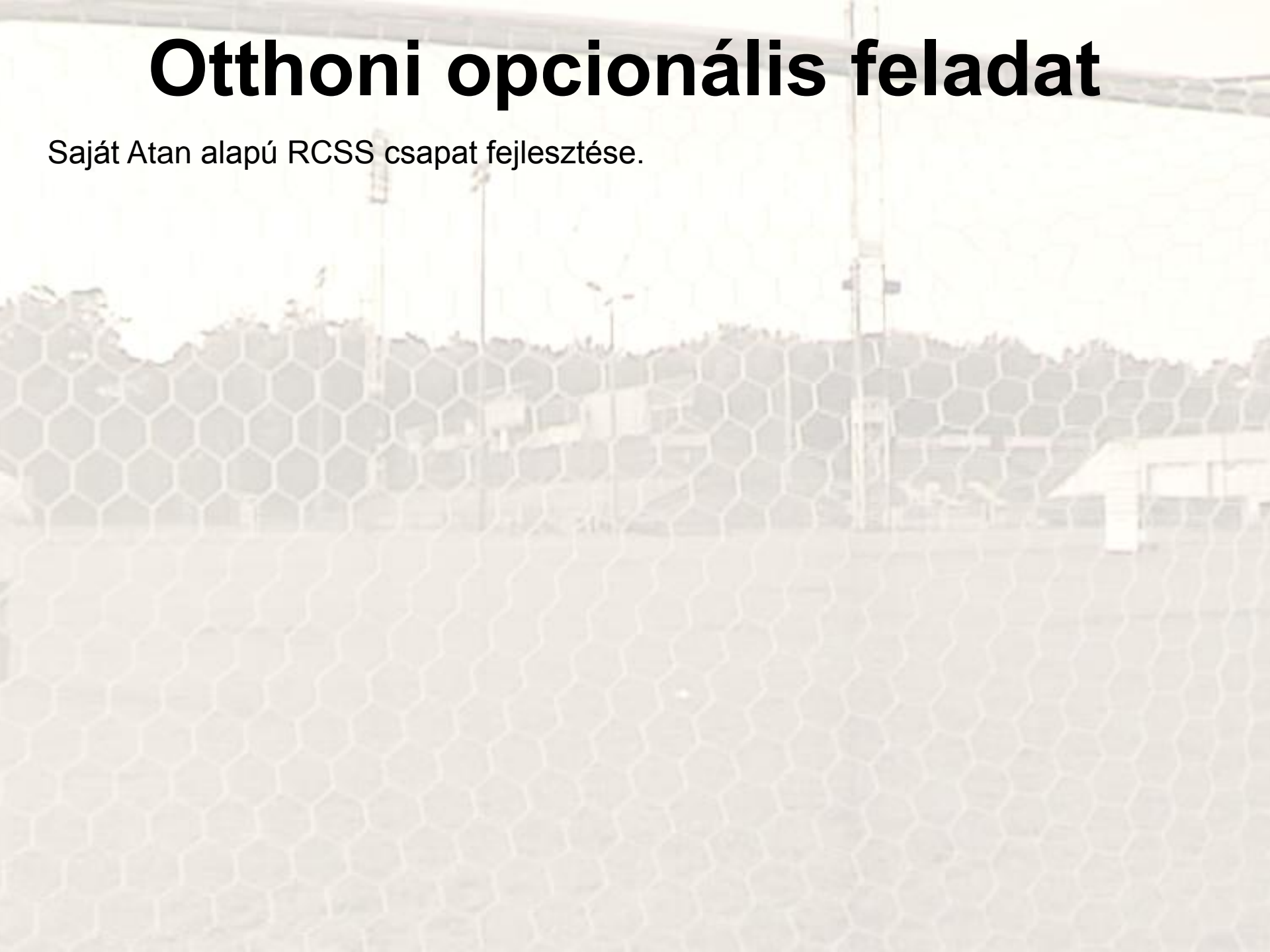
Mit csinál és hogy működik ez a példa? Interneten ne próbálgasd! Csak lokálhoszton vagy a laboron, mert sok rendszergazda ezt már támadásnak veszi!

Output - KapuSzkenner (run)

```
18 nem figyel  
19 nem figyel  
20 nem figyel  
21 figyel  
22 figyel  
23 nem figyel  
24 nem figyel  
25 nem figyel  
26 nem figyel
```

Otthoni opcionális feladat

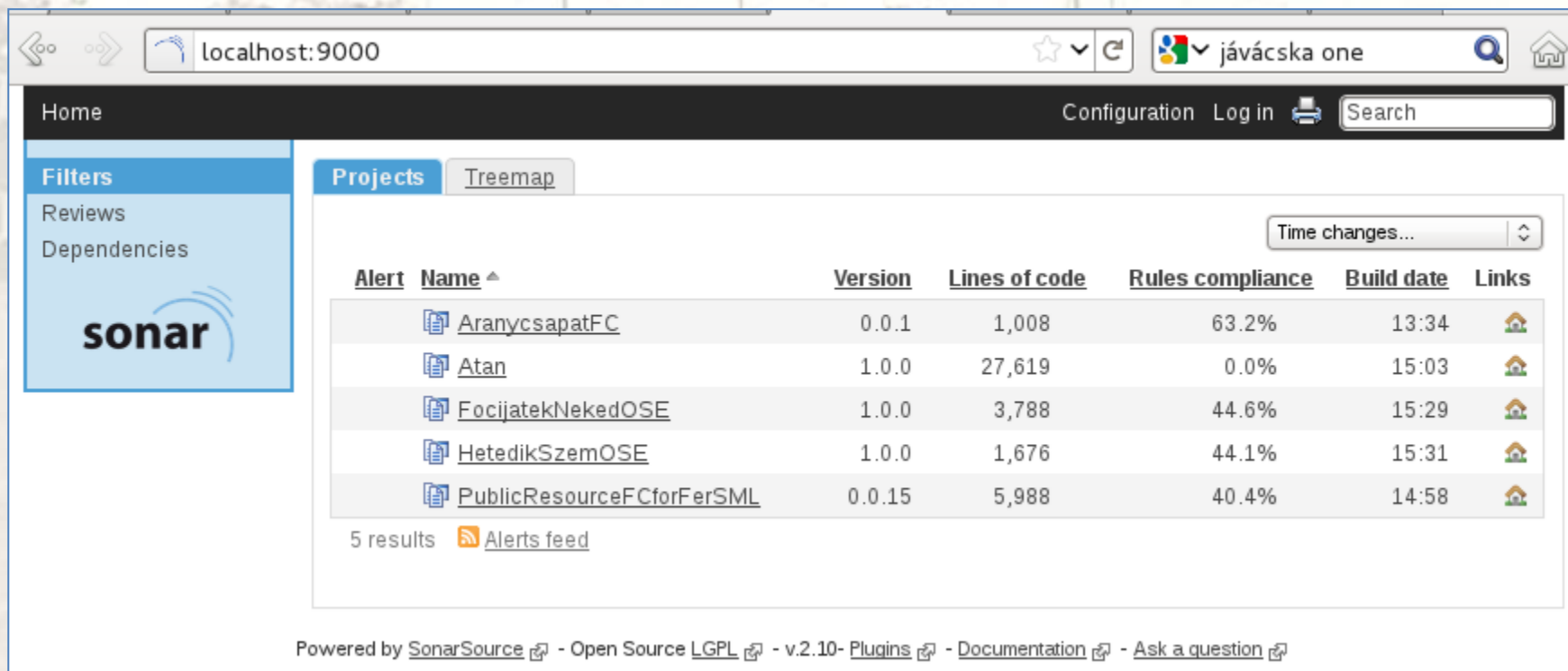
Saját Atan alapú RCSS csapat fejlesztése.













Otthoni opcionális feladat


Kísérletezés a Sonar és Xradar Maven bővítményekkel:

- 1) <http://mojo.codehaus.org/sonar-maven-plugin/>
- 2) <http://xradar.sourceforge.net/>



The screenshot shows the SonarQube web interface running on localhost:9000. The browser's address bar shows 'localhost:9000' and the user is logged in as 'jávácška one'. The interface includes a navigation menu on the left with 'Home', 'Filters', 'Reviews', and 'Dependencies'. The main content area displays a table of projects under the 'Projects' tab. The table has columns for 'Alert', 'Name', 'Version', 'Lines of code', 'Rules compliance', 'Build date', and 'Links'. There are 5 results listed, each with a project icon and a home link. Below the table, there is a '5 results' summary and an 'Alerts feed' link. At the bottom, there is a footer with links to 'SonarSource', 'Open Source', 'v.2.10- Plugins', 'Documentation', and 'Ask a question'.

Alert	Name ^	Version	Lines of code	Rules compliance	Build date	Links
	 AranycsapatFC	0.0.1	1,008	63.2%	13:34	
	 Atan	1.0.0	27,619	0.0%	15:03	
	 FocijatekNekedOSE	1.0.0	3,788	44.6%	15:29	
	 HetedikSzemOSE	1.0.0	1,676	44.1%	15:31	
	 PublicResourceFCforFerSML	0.0.15	5,988	40.4%	14:58	

5 results  [Alerts feed](#)

Powered by [SonarSource](#) - Open Source [LGPL](#) - v.2.10- [Plugins](#) - [Documentation](#) - [Ask a question](#)

Violations Drilldown

- Time Machine
- Clouds
- Design
- Hotspots
- Libraries



Severity

Blocker	0
Critical	236
Major	4,889
Minor	12,406
Info	174

Rule

Inner Assignment	801
Method Name	800
Parameter Assignment	462
If Else Stmt's Must Use Braces	388
Boolean Expression Complexity	216
Visibility Modifier	160

atan.parser.player	330
atan.parser.coach	288
atan.parser.trainer	179
atan.model	2
atan.parser	2

ByteBuffer	2
------------	---

Path: Major [Clear](#) > Inner Assignment [Clear](#) > atan.model [Clear](#) >

atan.model.ByteBuffer

[Coverage](#) |
 [Dependencies](#) |
 [Duplications](#) |
 [LCOM4](#) |
 [Source](#) |
 [Violations](#) |
 [Raw](#) |
 [New window](#)

2 violations |
 Blocker: 0 |
 Critical: 0 |
 Major: 2 |
 Minor: 0 |
 Info: 0

Full source |
 Time changes... |
 Inner Assignment (2)

```

40 public void setString(String str) throws IOException {
41     Reader r = new StringReader(str);
42     int c;
43     reset();
44     while (((c = r.read()) != 0) && (c != -1)) {

```

Inner Assignment | 7 minutes

Inner assignments should be avoided.

Kísérlet

- 1) h
- 2) h

Components

Violations Drilldown

- Time Machine
- Clouds
- Design
- Hotspots
- Libraries



Severity	
Blocker	0
Critical	9
Major	292
Minor	2,645
Info	86

Rule	
Cyclomatic Complexity	15
Ncss Method Count	15
Local Variable Name	12
Hidden Field	8
Avoid Print Stack Trace	5
Parameter Assignment	4

hu.javacska.sfa	14
hu.javacska.xml	1

FootballMatch	6
SFApplet	5
ManyFootballWorldCupSim	1
FootballWorldCup	1
AllFormationsSim	1
AvatarLoader	1

Path: Major [Clear](#) » Cyclomatic Complexity [Clear](#) »

hu.javacska.sfa.FootballMatch

Coverage Dependencies Duplications LCOM4 Source **Violations** Raw | New window

1,237 violations Blocker: 0 Critical: 0 Major: 80 Minor: 1,157 Info: 0

Full source | Time changes... | Cyclomatic Complexity (6)

```
816 public int felallas() {
817     return seFelallas;
818 }
819
820 protected void felallashozEmberfogas() {
821
822     // az első 4 eset leörösa (felcserölve is)
823     if (tervezettFelallas[0] == 0) { // 3x3x3
```

Cyclomatic Complexity | 17 minutes

Cyclomatic Complexity is 41 (max allowed is 10).

Kötelező olvasmány

Kivételkezelés

NYJ I/75-88

NYJ I/60-73

Párhuzamosság

NYJ I/472-509

NYJ I/219-259

Gyűjtemények

NYJ I/286-302

NYJ I/163-175