

# Prog2, AspectJ bevezetés

## 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](mailto:batfai.norbert@inf.unideb.hu)

Skype: batfai.norbert

Prog2\_6.ppt, v.: 0.0.6, 2011. 12. 08.

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

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

Az óra blogja: <http://progater.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, 2012 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) A nagytestvér beleszótt egy aspektust a csapatomba:

[http://progpater.blog.hu/2011/12/04/a\\_nagytestver\\_beleszott\\_egy\\_aspektust\\_a\\_csapatomba](http://progpater.blog.hu/2011/12/04/a_nagytestver_beleszott_egy_aspektust_a_csapatomba)

# Célok és tartalom

## **Előadás**

- a) Programozási paradigmák
- b) AspectJ nyelvi bevezetés: vágási pont, csatlakozási pont, tanács, átszövő vonatkozás.

## **Labor**

- a) Atan alapú robotfoci:  
Golden Team FC 0.0.4 szöveg bevezetése

## **Laborkártyák**

- a) AspectJ kártyák Golden Team FC 0.0.4 kártyák

## **Otthoni opcionális feladat**

- a) Saját Atan alapú RCSS csapat fejlesztése (esetleg szövése, egységtesztelése)
- b) 100 fölött: GlassFish, Geronimo, Tomcat, Jboss EJB tesztek, fűrtözés stb.

# Minimális gyakorlati cél

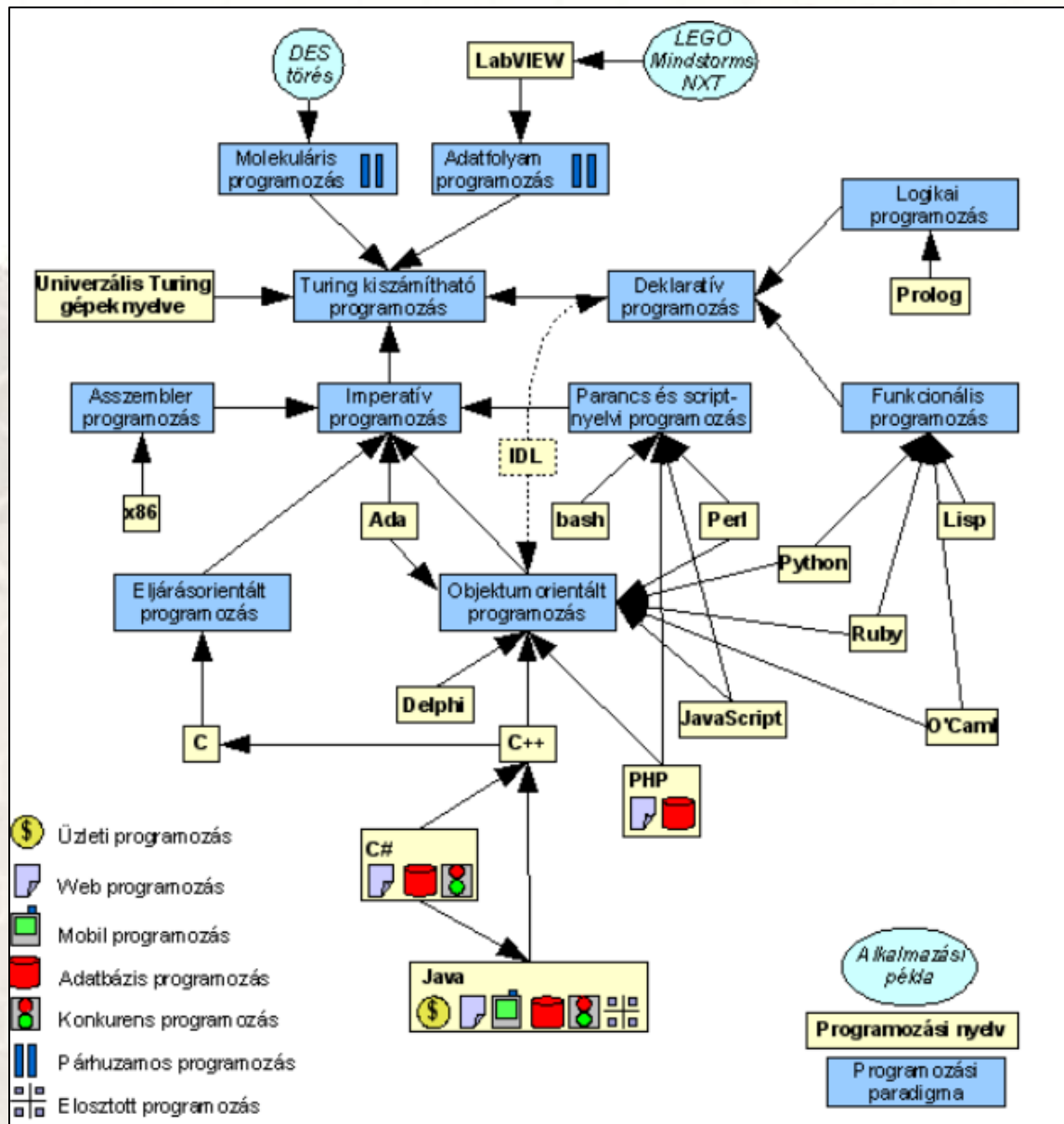
- a) A hallgató el tudja készíteni egy egyszerű szövést, például a saját robotfoci csapatába.



# Minimális elméleti cél

- 1) AOP alapelvek, használati esetek átszövő vonatkozásokra.
- 2) AspectJ nyelv kapcsán: vágási pont, csatlakozási pont, tanács, átszövő vonatkozás.

# Ism.: Java (eredet)



- Imperatív
- Objektum orientált és eljárás orientált paradigma mentén

Aki Javában programoz OO programoz, de az egyszerű típusok, változó fogalom, kifejezések mint C-ben.

# Paradigmák néhány szóban

Imperatív

Struktúrált

Eljárásorientált

OO

Eseményvezérelt (vs. komponens alapú)

Komponens orientált (pl. CORBA)

Szolgáltatás orientált (webszolgok, BPML, BPEL)

Deklaratív



# Problémamegoldás

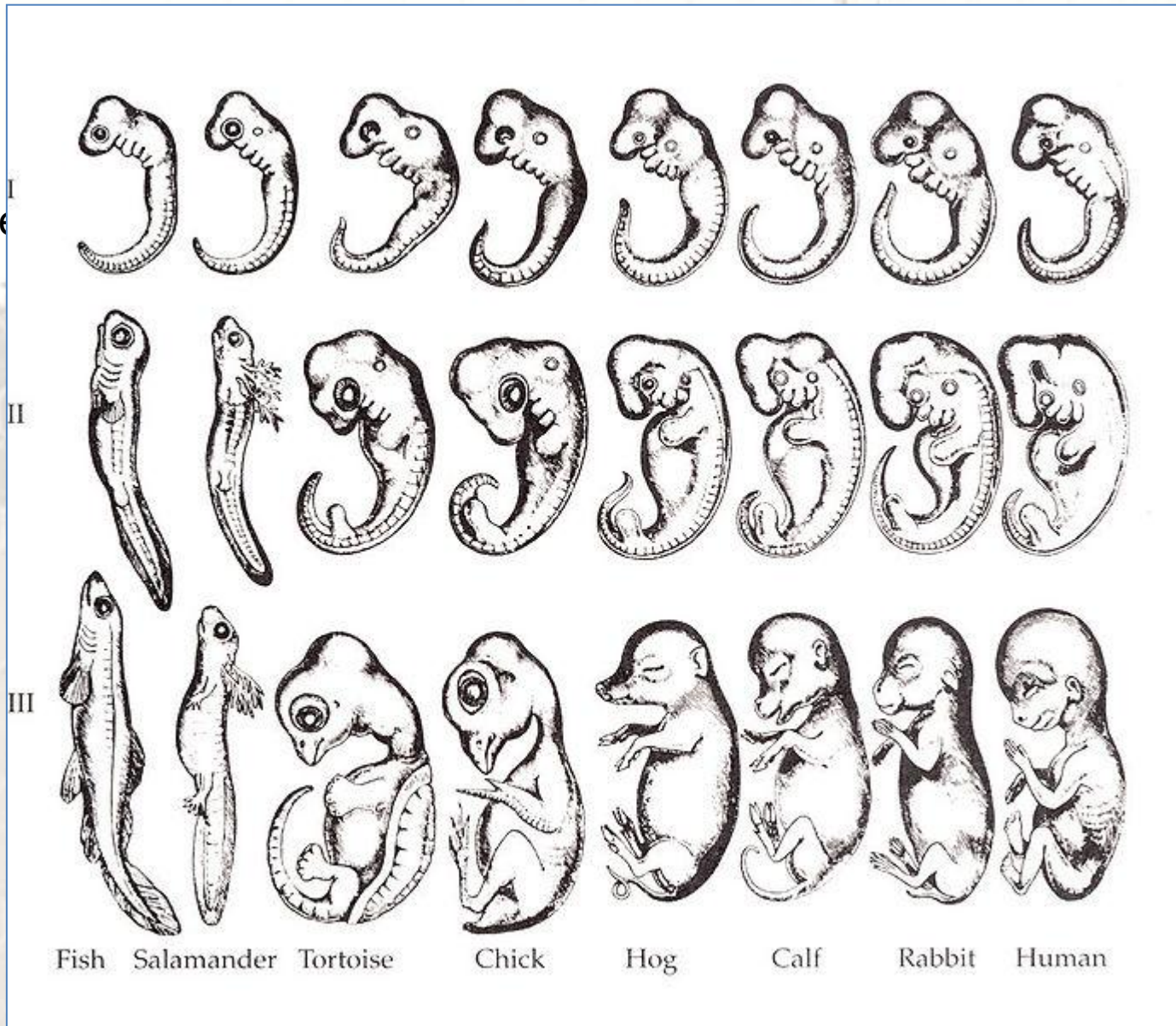
Funkcionális dekompozíció – **MIT?** (függvény)

Adat dekompozíció – **MIVEL?** (objektum)

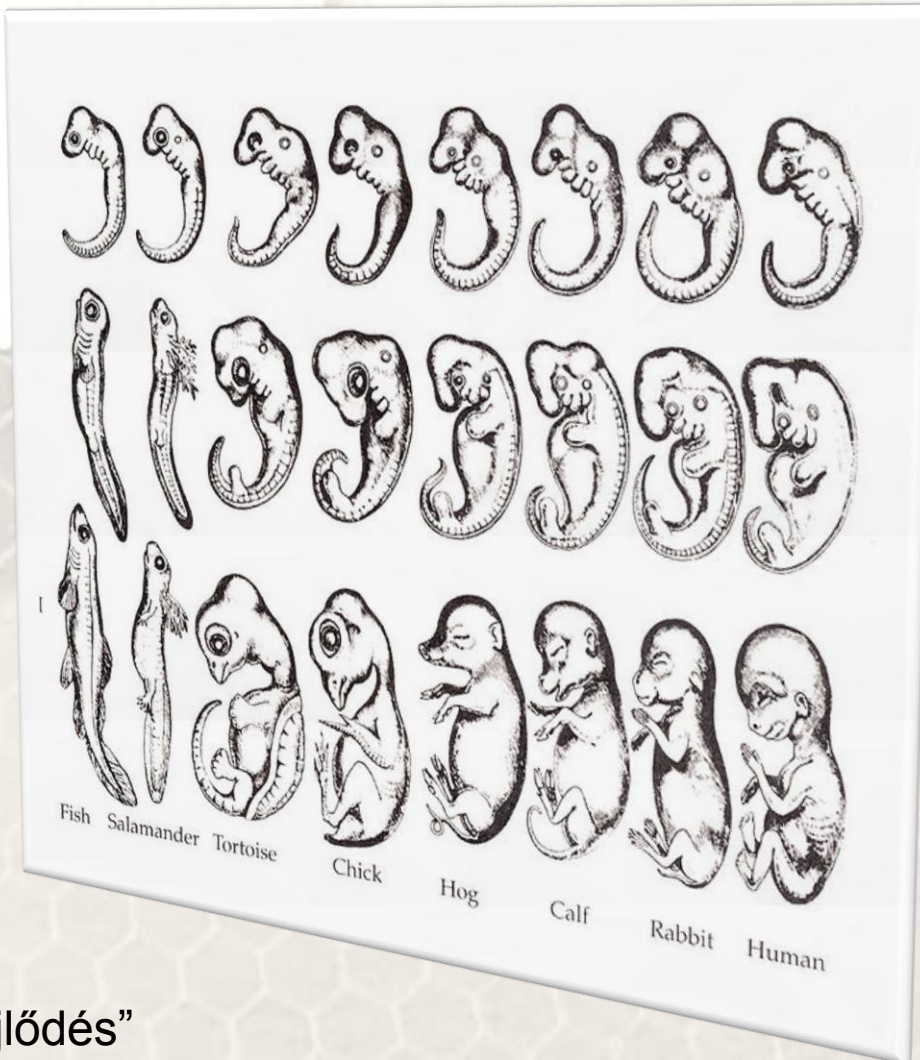
A **MIT** hajlamosabb a változásra, mint a **MIVEL!**

# Hogyan lehet jó programozókat

Az egyedfe



# Hogyan lehet jó programozókat „gyártani”?



„egyedfejlődés” /személyes/

(C-16, C-64, +4)  
Assembly, BASIC

(IBM XT/AT)  
Turbo Pascal  
Mprolog

Pascal  
C

C++  
Java  
AspectJ

gimnázium

egyetem

„önképzés”

„törzsfejlődés”

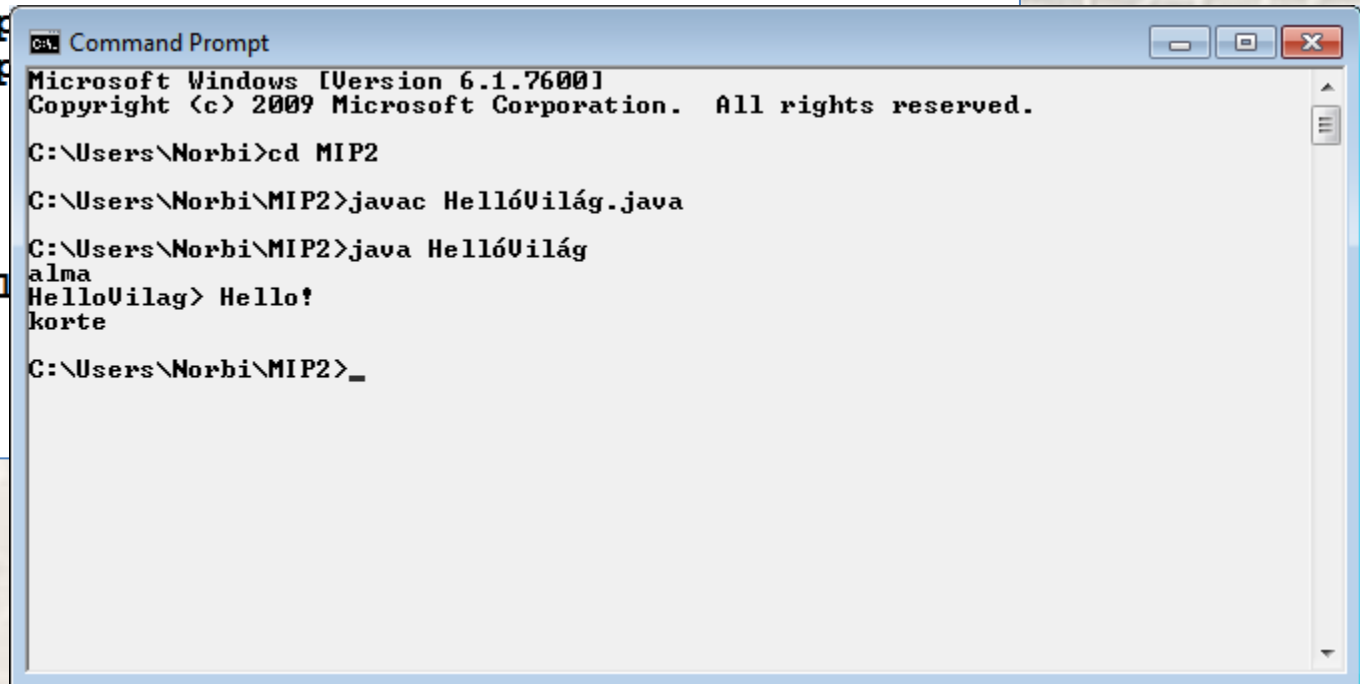
Imperatív - Struktúrált - Eljárásorientált - OO - AOP ...

Assembly, Fortran, C, C++, Java, AspectJ, ...

**OO->AO**

# Házi használatra: „alma-körte tesztelés”

```
public class HellóVilág {  
    public class HellóVilág {  
        public void helló() {  
            // Itt nem akar működni valami... "alma-körte tesztelés":  
            System.out.println("alma");  
            // van alma, körtéig eljut?  
            System.out.p  
            System.out.p  
        }  
        public static  
        .....  
        new HellóVil  
    }  
}
```



```
Command Prompt  
Microsoft Windows [Version 6.1.7600]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
  
C:\Users\Norbi>cd MIP2  
C:\Users\Norbi\MIP2>javac HellóVilág.java  
C:\Users\Norbi\MIP2>java HellóVilág  
alma  
HelloVilag> Hello!  
korte  
C:\Users\Norbi\MIP2>_
```

# „alma-körte tesztelés” AspectJ-vel

```
public class HellóVilág {
```

```
public aspect ElőtteUtána {
```

```
    public pointcut fgvHívás(): call(public void HellóVilág.helló());
```

```
    before(): fgvHívás() {  
        System.out.println("ElotteUtana> Alma");  
    }
```

```
    after(): fgvHívás()  
        System.out.println("HelloVilag> Hello!");  
}
```

Command Prompt

Microsoft Windows [Version 6.1.7600]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Norbi>cd MIP2

C:\Users\Norbi\MIP2>ajc HellóVilág.java ElőtteUtána.aj

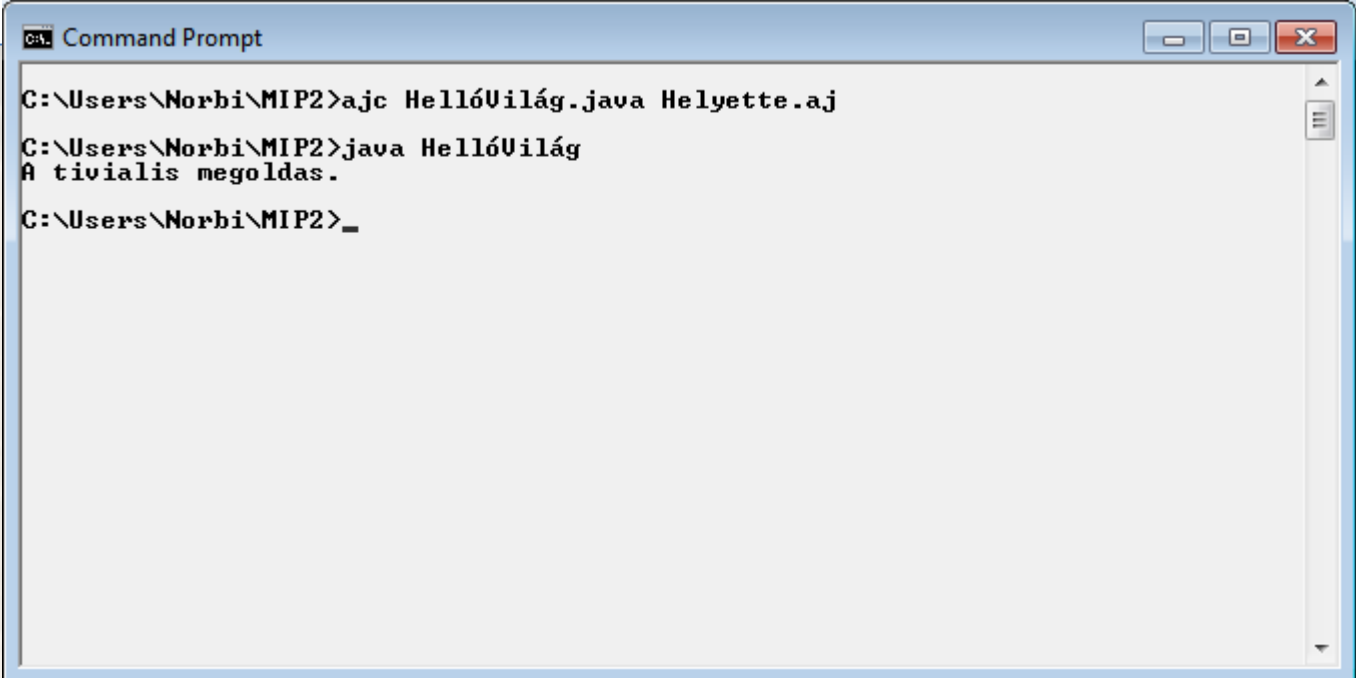
C:\Users\Norbi\MIP2>java HellóVilág

ElotteUtana> Alma  
HelloVilag> Hello!  
ElotteUtana> Korte

C:\Users\Norbi\MIP2>\_

# AspectJ

```
public aspect Helyette {  
  
    public pointcut fgvHívás(): call(public void HellóVilág.helló());  
  
    void around(): fgvHívás() {  
        System.out.println("A tivialis megoldas.");  
    }  
  
}
```



The screenshot shows a Windows Command Prompt window with the following text:

```
CA Command Prompt  
C:\Users\Norbi\MIP2>ajc HellóVilág.java Helyette.aj  
C:\Users\Norbi\MIP2>java HellóVilág  
A tivialis megoldas.  
C:\Users\Norbi\MIP2>_
```

# Az aspektus

- Join point (csatlakozási pont)
- Pointcut (vágási pont)
- Advice (tanács)

```
public class HellóVilág {
```

```
    public void helló() {
```

A csatlakozási pontok az eredeti programban vannak, ezeket az aspektus vágási pontjaival jelöljük ki.

```
    }
```

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

```
        new HellóVilág().helló();
```

```
public aspect ElőtteUtána {
```

```
    public pointcut fgvHívás(): call(public void HellóVilág.helló());
```

```
    before(): fgvHívás() {
```

```
        System.out.println("ElotteUtana> Alma");
```

```
    }
```

```
    after(): fgvHívás() {
```

```
        System.out.println("ElotteUtana> Korte");
```

```
    }
```

```
}
```



# AspectJ

```
public aspect Nagytesó {  
  
    before(): call(* *(..)) && !cflow(adviceexecution()) {  
  
        System.out.println("Nagyteso> " + thisJoinPointStaticPart.getSourceLocation());  
        System.out.println("Nagyteso> " + thisJoinPointStaticPart.getKind());  
        System.out.println("Nagyteso> " + thisJoinPointStaticPart.getSignature());  
  
    }  
  
}
```

Command Prompt

```
C:\Users\Norbi\MIP2>ajc HellóVilág.java Nagytesó.aj  
C:\Users\Norbi\MIP2>java -Dfile.encoding=Cp852 HellóVilág.java  
Nagyteso> HellóVilág.java:13  
Nagyteso> method-call  
Nagyteso> void HellóVilág.helló()  
Nagyteso> HellóVilág.java:5  
Nagyteso> method-call  
Nagyteso> void java.io.PrintStream.println(String)  
alma  
Nagyteso> HellóVilág.java:7  
Nagyteso> method-call  
Nagyteso> void java.io.PrintStream.println(String)  
HelloVilag> Hello!  
Nagyteso> HellóVilág.java:8  
Nagyteso> method-call  
Nagyteso> void java.io.PrintStream.println(String)  
korte  
C:\Users\Norbi\MIP2>_
```

```
1 public class HellóVilág {  
2  
3     public void helló() {  
4         // Itt nem akar működni valami...  
5         System.out.println("alma");  
6         // van alma, körtéig eljut?  
7         System.out.println("HelloVilag> H  
8         System.out.println("korte");  
9     }  
10  
11     public static void main(String[] ar  
12  
13         new HellóVilág().helló();  
14  
15     }  
16 }  
17
```

# AspectJ

```
public aspect NagytésóExec {  
  
    before(): execution(* *(..)) && !cflow(adviceexecution()) {  
  
        System.out.println("Nagyteso> " + thisJoinPointStaticPart.getSourceLocation());  
        System.out.println("Nagyteso> " + thisJoinPointStaticPart.getKind());  
        System.out.println("Nagyteso> " + thisJoinPointStaticPart.getSignature());  
  
    }  
}
```

Command Prompt

```
C:\Users\Norbi\MIP2>ajc HellóVilág.java NagytésóExec.aj  
C:\Users\Norbi\MIP2>java -Dfile.encoding=Cp852 HellóVilág  
Nagyteso> HellóVilág.java:11  
Nagyteso> method-execution  
Nagyteso> void HellóVilág.main(String[])  
Nagyteso> HellóVilág.java:3  
Nagyteso> method-execution  
Nagyteso> void HellóVilág.helló()  
alma  
HelloVilag> Hello!  
korte  
C:\Users\Norbi\MIP2>
```

```
1 public class HellóVilág {  
2  
3     public void helló() {  
4         // Itt nem akar működni valami.  
5         System.out.println("alma");  
6         // van alma, körtéig eljut?  
7         System.out.println("HelloVilag");  
8         System.out.println("korte");  
9     }  
10  
11     public static void main(String[] args) {  
12  
13         new HellóVilág().helló();  
14  
15     }  
16 }  
17
```

# Aspect J



# AspectJ

## ● SZOFTVERFEJLESZTÉS

---

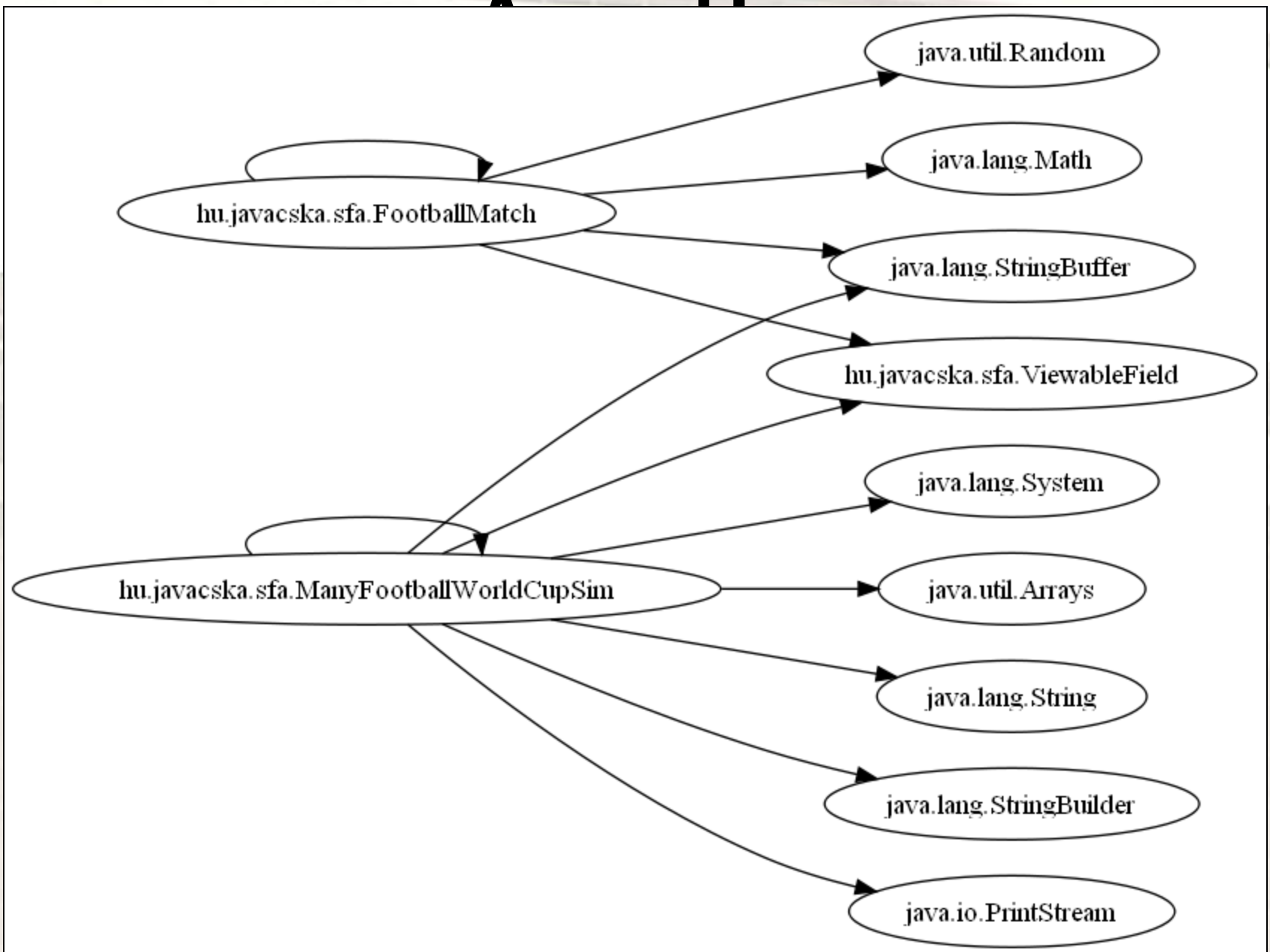
### **Van-e az objektum-orientált programoknak anyanyelve? – avagy egy analitikai szöveg bevezetése**

BÁTFAI NORBERT

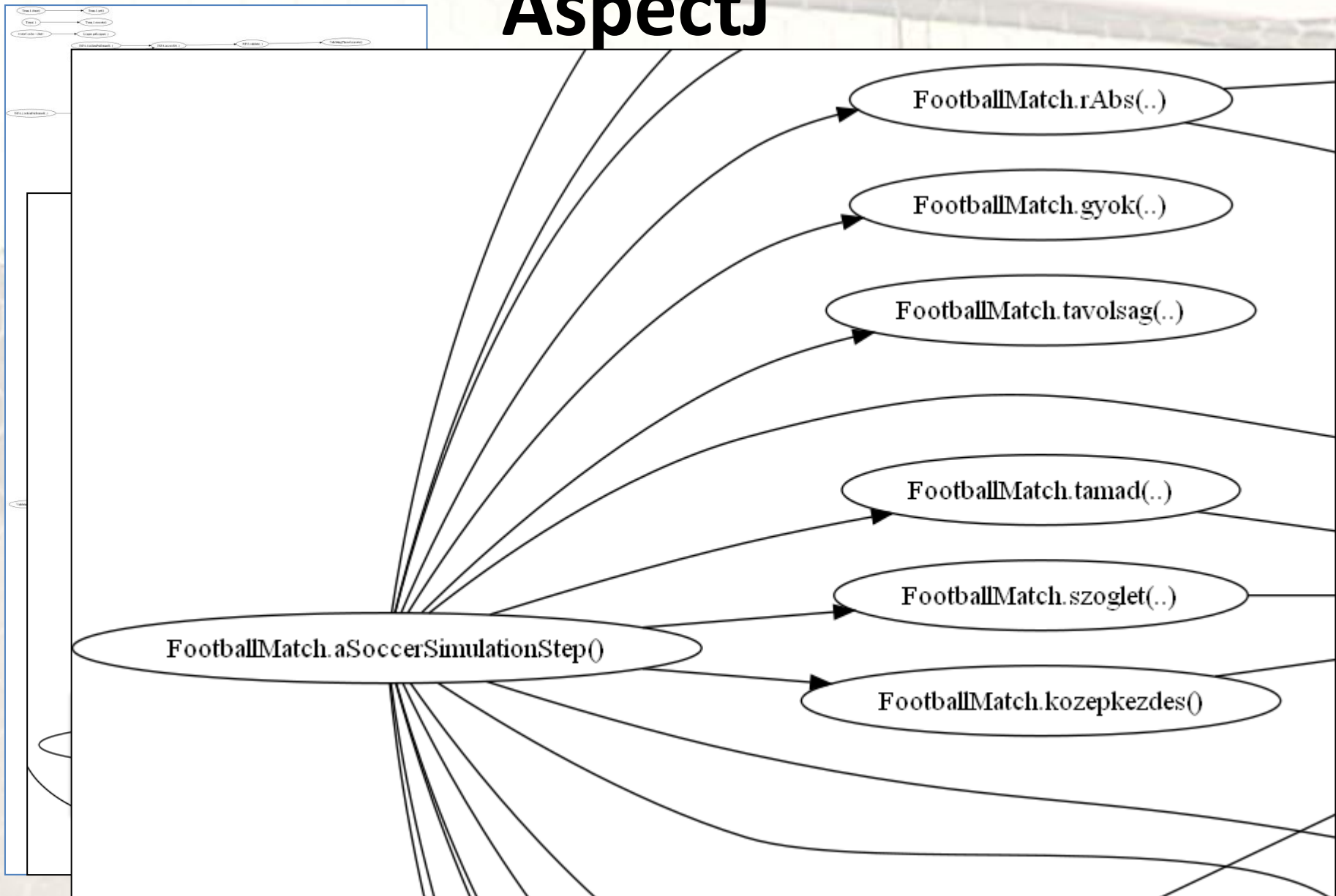
*Debreceni Egyetem, Informatikai Kar, Információ Technológia Tanszék  
batfai.norbert@inf.unideb.hu*

*Kulcsszavak: szoftverfejlesztés, információ-visszakeresés, AOP mérések, AspectJ, Java, PageRank, Zipf, Ant, Maven*

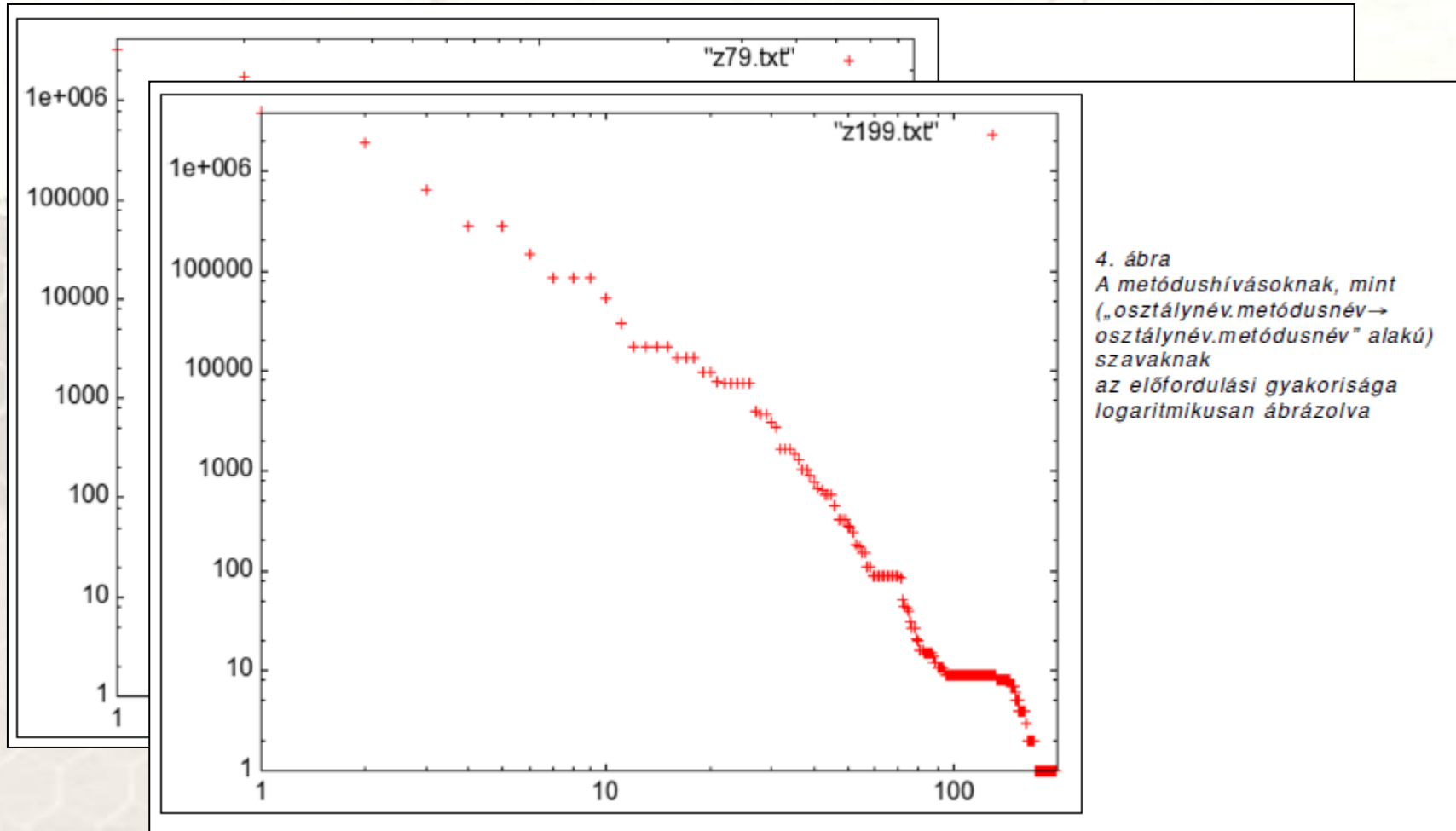
**Rövid írásunkban Java parancssori, Ant vagy Maven forrásprojektekhez vezetünk be olyan AspectJ aspektust, ami analitikát sző a program futásába. Az osztályok és a metódusok halmazára alkalmazza a PageRank algoritmust, illetve az objektumok kommunikációjára heurisztikusan ellenőrzi a Zipf törvény teljesülését.**



# AspectJ



# AspectJ



4. ábra  
A metódushívásoknak, mint („osztálynév.metódusnév→ osztálynév.metódusnév” alakú) szavaknak az előfordulási gyakorisága logaritmikusan ábrázolva

Bátfai N., (2011), *Van-e az objektum orientált programoknak anyanyelve: avagy egy analitikai szöveg bevezetése*, Híradástechnikai Szemle, Híradástechnika, 66. évf. 2. sz., 27-32, 2011.

[http://www.hiradastechnika.hu/data/upload/file/2010/HT2011\\_2\\_komplett.pdf](http://www.hiradastechnika.hu/data/upload/file/2010/HT2011_2_komplett.pdf)

# Gépi tudatosság



Cornell University  
Library

arXiv.org > cs > arXiv:1108.2865

Search or Art

Computer Science > Artificial Intelligence

## Conscious Machines and Consciousness Oriented Programming

Norbert Bátfai

*(Submitted on 14 Aug 2011)*

In this paper, we investigate the following question: how could you write such computer programs that can work like conscious beings? The motivation behind this question is that we want to create such applications that can see the future. The aim of this paper is to provide an overall conceptual framework for this new approach to machine consciousness. So we introduce a new programming paradigm called Consciousness Oriented Programming (COP).

Comments: 25 pages, 8 figures

Subjects: **Artificial Intelligence (cs.AI)**

MSC classes: 68T35

ACM classes: I.2.5; F.1.1

Cite as: [arXiv:1108.2865v1](http://arxiv.org/abs/1108.2865v1) [cs.AI]

### Submission history

From: Norbert Bátfai [[view email](#)]

[v1] Sun, 14 Aug 2011 12:27:39 GMT (669kb,D)

<http://arxiv.org/abs/1108.2865>



# COP

**Definition 1** (Knowing the Future) A computer program is considered to know the future input if it can predict that

**Definition 2** (Knowing the Future State) A computer program is considered to know the future state if it can predict that

**Definition 3** (Conscious Computer Program) A computer program is considered conscious if it

**Definition 4** (Self-Conscious Computer Program) A computer program is considered self-conscious if it

**Definition 5** (Intuitive Computer Program) A computer program is considered intuitive if it

**Example 1** (Walking Across the Zebra Crossing). *This is a trivial example of daily life. Every day the author goes across the zebra crossing shown in Figure 1. Here I am standing (at a safe) 3-4 meters away from the kerb. Then I am going to start to go when the traffic light for cars has changed to yellow, because I know from former personal experience that the traffic light for pedestrians changes to green soon afterwards.*

*This "conscious" behavior represents an advantage for the author over the other pedestrians, because while he are already on the move, others will be waiting for the green signal of the traffic light for pedestrians.*



Figure 1: The author walks across the zebra crossing.

# COP

```
public class Game {  
  
    public static final int FIELD_X = 80;  
    public static final int FIELD_Y = 24;  
    public static final int BALL_LIFESPAN = 1000;  
  
    public static void main(String [] args) {
```

```
        final Ball ball = new  
        final Player playerP =  
        playerQ = new Player(F  
  
        int pointsP = 0, points  
        for (int i = 0; i < BA  
  
            ball.move();
```

```
                playerP.percepti  
            }  
        }.start();
```

```
        new Thread() {  
            public void run() {  
                playerQ.perception(ball.y);  
            }  
        }
```

```
    }.start();
```

```
        if (ball.x == 0 && playerP.y == ball.y) {  
            ++pointsP;  
        }
```

```
        if (ball.x == FIELD_X - 1 && playerQ.y == ball.y) {  
            ++pointsQ;  
        }
```

```
    }
```

```
        System.out.println(pointsP + " vs " + pointsQ);
```

```
    }
```

```
}
```

# COP

The ball can move all four directions with the same probability or, to be more precise, its movement is a random walk.

```
class Ball {  
  
    int x = Game.FIELD_X / 2, y = Game.FIELD_Y / 2;  
  
    void move() {  
        int dx = (int) (Math.random() * 3) - 1;  
        int dy = (int) (Math.random() * 3) - 1;  
  
        if (x + dx < Game.FIELD_X && x + dx >= 0) {  
            x += dx;  
        }  
        if (y + dy < Game.FIELD_Y && y + dy >= 0) {  
            y += dy;  
        }  
    }  
}
```

Listing 2: The sou

The players can only move up and down on the sides of the field. They endeavor to catch the ball when it reaches the sides of the field. Our examples, the players P and Q are aware of the reality via an interface called Sensory.

```
interface Sensory {  
    void perception(int bally);  
}
```

# COP

```
class Player implements Sensory {  
  
    int x = 0, y = Game.FIELD_Y / 2;  
  
    public Player(int x) {  
        this.x = x;  
    }  
  
    public void perception(int ballY) {  
        move(ballY);  
    }  
  
    protected void move(  
        if (y < ballY) {  
            ++y;  
        } else if (y > ba  
            --y;  
        }  
    }  
}
```

Listing 4: The

```
aspect Delay {  
  
    public pointcut perceptionCall():  
        call(public void Player.perception(int));  
  
    before(): perceptionCall() {  
  
        try {  
            Thread.sleep(500);  
        } catch (InterruptedException e){e.printStackTrace();}  
    }  
}
```

Listing 5: The source code for the Delay aspect.

# COP

Table 1: Execution results of the delay aspect (with the variable BALL-LIFESPAN set to 100.000).

Naive example of the <i>living in the past</i>		
Time [ms]	Aver. Scores	Exec. Time [min]
javac	1088	9.7
javac	1156	10.0
no aspect	1162	9.6
no blocking	1036	9.7
0.001	249	11.2
0.01	246	10.0
0.1	239	10.2
0.5	227	9.9

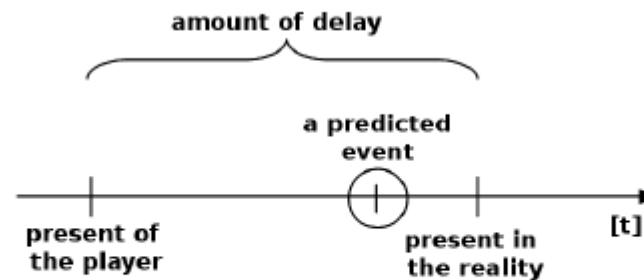


Figure 3: A simple schematic drawing of the well-known conception of "living in the past".

# A nagytstvér beleszótt egy aspektust a csapatomba

```
public aspect NagytesoKick {  
    before(): call(* kick(..)) && !cflow(adviceexecution()) {  
        System.out.println("Nagyteso> " + thisJoinPointStaticPart.getSourceLocation());  
        System.out.println("Nagyteso> " + thisJoinPointStaticPart.getKind());  
        System.out.println("Nagyteso> " + thisJoinPointStaticPart.getSignature());  
    }  
}
```

# A nagytestvér beleszótt egy aspektust a csapatomba

```
<dependency>
  <groupId>org.aspectj</groupId>
  <artifactId>aspectjrt</artifactId>
  <version>1.6.8</version>
</dependency>
```

```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>aspectj-maven-plugin</artifactId>
  <version>1.4</version>
  <configuration>
    <complianceLevel>1.6</complianceLevel>
  </configuration>
  <executions>
    <execution>
      <goals>
        <goal>compile</goal>
        <goal>test-compile</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

# A nagytestvér beleszótt egy aspektust a csapatomba

```
Command Prompt
47836 [Bcsapat Player # 11] INFO hu.fersml.magyarfc.Jatekos - PASSZ/KIS POCC UE
ZETEM
47972 [Bcsapat Player # 11] INFO hu.fersml.magyarfc.Jatekos - EGYIK KAPUT SEM L
ATOM
Nagyteso> Jatekos.java:91
Nagyteso> method-call
Nagyteso> void atan.model.ActionsPlayer.kick(int, double)
47975 [Bcsapat Player # 11] INFO hu.fersml.magyarfc.Jatekos - PASSZ/KIS POCC UE
ZETEM
49031 [Bcsapat Player # 11] INFO hu.fersml.magyarfc.Jatekos - EGYIK KAPUT SEM L
ATOM
Nagyteso> Jatekos.java:91
Nagyteso> method-call
Nagyteso> void atan.model.ActionsPlayer.kick(int, double)
49036 [Bcsapat Player # 11] INFO hu.fersml.magyarfc.Jatekos - PASSZ/KIS POCC UE
ZETEM
49634 [Bcsapat Player # 11] INFO hu.fersml.magyarfc.Jatekos - EGYIK KAPUT SEM L
ATOM
Nagyteso> Jatekos.java:91
Nagyteso> method-call
Nagyteso> void atan.model.ActionsPlayer.kick(int, double)
49638 [Bcsapat Player # 11] INFO hu.fersml.magyarfc.Jatekos - PASSZ/KIS POCC UE
ZETEM
C:\Users\Norbi\MIP2\GoldenTeamFC-0.0.4>_
```



# Hol itt a jobb egérfogó?

Hogyan tudnál válaszolni például az alábbi kérdésekre?

- 1) A csapatod hányszor rúgott bele a labdába?
- 2) Átlagosan mekkora erővel?
- 3) Átlagosan merre?

# Laborkártyák

## Mit szősz kis tanács?

```
public aspect NagytesoKick {  
  
private long szamlalo = 1;  
private long eroSum = 0;  
private double szogSum = 0.0;  
  
    before(int ero, double szog): call(* kick(int, double)) && !cflow(adviceexecution()) && args(ero, szog) {  
  
        System.out.println("Nagyteso> " + thisJoinPointStaticPart.getSourceLocation());  
        System.out.println("Nagyteso> " + thisJoinPointStaticPart.getKind());  
        System.out.println("Nagyteso> " + thisJoinPointStaticPart.getSignature());  
  
        ++szamlalo;  
        eroSum += ero;  
        szogSum += szog;  
  
        System.out.println("Nagyteso> " + (szamlalo-1));  
        System.out.println("Nagyteso> " + (double)eroSum / szamlalo);  
        System.out.println("Nagyteso> " + szogSum / szamlalo);  
  
    }  
}
```

MIRC : bash

File Edit View Bookmarks Settings Help

19.0

73684 [GoldenFC3 Player # 8] INFO hu.fersml.magyarfc.Jatekos - PASSZ/FORGOK

73723 [GoldenFC3 Player # 3] INFO hu.fersml.magyarfc.Jatekos - PASSZ/FORGOK

Nagyteso> Jatekos.java:91

Nagyteso> method-call

Nagyteso> void atan.model.ActionsPlayer.kick(int, double)

Nagyteso> 50

Nagyteso> 20.784313725490197

Nagyteso> 2.7058823529411766

73753 [GoldenFC3 Player # 6] INFO hu.fersml.magyarfc.Jatekos - PASSZ/KIS POCC VEZETEM

73793 [GoldenFC3 Player # 10] INFO hu.fersml.magyarfc.Jatekos - PASSZ/FORGOK

# Otthoni opcionális feladat

Saját Atan alapú RCSS csapat fejlesztése.

<http://www.inf.unideb.hu/~nbatfai/PLB2011osz/tabella.html>

100 fölött: GlassFish, Geronimo, Tomcat, Jboss EJB tesztek, fűrtözés stb.

# Kötelező olvasmány

NYJ

NYJ

NYJ

NYJ

# Ajánlott olvasmány

Russ Miles: AspectJ cookbook

[http://books.google.com/books?id=AKuBIJGI7iUC&printsec=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](http://books.google.com/books?id=AKuBIJGI7iUC&printsec=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)

(LENGYEL LÁSZLÓ, LEVENDOVSKY TIHAMÉR: Aspektus-orientált programozás)

[http://www.hiradastechnika.hu/data/upload/file/2005/2005\\_6/HT\\_0506a-5.pdf](http://www.hiradastechnika.hu/data/upload/file/2005/2005_6/HT_0506a-5.pdf)

(Paller Gábor Javás cikkek)

<http://pallergabor.uw.hu/hu/java-app/>

```
@book{Miles:2004:AC:1096117,  
  author = {Miles, Russell},  
  title = {AspectJ Cookbook},  
  year = {2004},  
  isbn = {0596006543},  
  publisher = {O'Reilly Media, Inc.},  
}
```