

Varga Mátyás

## INBPM0315-17 Magas szintű programozási nyelvek 2

### 1.hét

#### **OO szemlélet :**

Egységbezárás, adatrejtés, öröklés:

Java:

[https://github.com/megvagyhadnagy01/MagasProg2/tree/master/1\\_gyak/OO%20szeml%C3%A9let/polarJava](https://github.com/megvagyhadnagy01/MagasProg2/tree/master/1_gyak/OO%20szeml%C3%A9let/polarJava)

C++:

[https://github.com/megvagyhadnagy01/MagasProg2/tree/master/1\\_gyak/OO%20szeml%C3%A9let/polargenC%2B%2B](https://github.com/megvagyhadnagy01/MagasProg2/tree/master/1_gyak/OO%20szeml%C3%A9let/polargenC%2B%2B)

---

#### **Homokozó:**

A feladat megoldásához tutorként Bátfai Norbert tanár úr forráskódját vettem igénybe. igyekeztem a feladathoz hűen rámutatni a pointerekre illetve a referenciák. A feladatot értelmeztem és sikeresen fordítottam, futtattam.

Source Code:

[https://github.com/megvagyhadnagy01/MagasProg2/tree/master/1\\_gyak/Homokozo](https://github.com/megvagyhadnagy01/MagasProg2/tree/master/1_gyak/Homokozo)

---

#### **“Gagy”**

A Gagy feladat megoldásánál rámutat a különbségre hogy a program hogy reagál arra ha -128-nál kisebb számot adunk meg

és a while ciklus végtelen lesz. JDK forrásai szintjén:

vegyük a -129 esetet, ami fagy, azaz végtelen ciklus, ekkor

i-ben a -129 van

```
public static Integer valueOf(int i) {
```

```
    amivel ez az if
```

```
    if (i >= IntegerCache.low && i <= IntegerCache.high)
```

```
        nem teljesül (ellenőrizd az Integer.java forrásban!)
```

```
        return IntegerCache.cache[i + (-IntegerCache.low)];
```

```
        tehát ez fog lefutni mindkét esetben, amikor a Gagy progiban csináljuk a két Integer
```

```
        return new Integer(i);
```

```
    }
```

A válasz innen már trivi: a két integernél két új objektum jön létre, máshol a tárban más "mem.címmel" persze,

így a hasonlításuk hamis lesz, hiszen különböző objectek, különböző "címmel".

A másik esetben (a -128 esetén) pedig ugyanarra a számra ugyanazt az objectet adja,

hiszen csak kiveszi a kész konzerv integer a poolból, így nyilván ez a két cím meg == lesz.

Source code:

[https://github.com/megvagyhadnagy01/MagasProg2/tree/master/1\\_gyak/Gagyi](https://github.com/megvagyhadnagy01/MagasProg2/tree/master/1_gyak/Gagyi)

Futtatás:

[https://github.com/megvagyhadnagy01/MagasProg2/blob/master/1\\_gyak/Gagyi/GagyiNemVegtelen.PNG](https://github.com/megvagyhadnagy01/MagasProg2/blob/master/1_gyak/Gagyi/GagyiNemVegtelen.PNG)

----->

### **Yoda:**

A Yoda feladatnál egy saját kódot írtam ami sikeresen fordult és fut.  
Igyekeztem megfogni a lényegét a feladatnál try illetve catch függvények és hasonló képpen NullPointerException-nal.

Source code:

[https://github.com/megvagyhadnagy01/MagasProg2/tree/master/1\\_gyak/Yoda](https://github.com/megvagyhadnagy01/MagasProg2/tree/master/1_gyak/Yoda)

Futtatás:

[https://github.com/megvagyhadnagy01/MagasProg2/blob/master/1\\_gyak/Yoda/Yoda.PNG](https://github.com/megvagyhadnagy01/MagasProg2/blob/master/1_gyak/Yoda/Yoda.PNG)

----->

### **Kódolás from scratch :**

A feladat megoldásához tutorként Bátfai Norbert tanár úr forráskódját vettem igénybe.  
Illetve a megadott angol nyelvű forrást igyekeztem értelmezni és az alapján is felfogni a feladatot  
A feladatot értelmeztem és sikeresen fordítottam, futtattam.

Source code:

[https://github.com/megvagyhadnagy01/MagasProg2/tree/master/1\\_gyak/PiBBP](https://github.com/megvagyhadnagy01/MagasProg2/tree/master/1_gyak/PiBBP)

Futtatás:

[https://github.com/megvagyhadnagy01/MagasProg2/blob/master/1\\_gyak/PiBBP/PiBBP.PNG](https://github.com/megvagyhadnagy01/MagasProg2/blob/master/1_gyak/PiBBP/PiBBP.PNG)

## 2.hét

### **Anti OO:**

Az algoritmust tutor segítségével sikerült megoldani. Sikeresen értelmeztem és futtatam a programot.

Source code:

[https://github.com/megvagyhadnagy01/MagasProg2/tree/master/2\\_gyak/Anti%20OO](https://github.com/megvagyhadnagy01/MagasProg2/tree/master/2_gyak/Anti%20OO)

Futtatás:

[https://github.com/megvagyhadnagy01/MagasProg2/blob/master/2\\_gyak/Anti%20OO/Screenshot%20from%202018-10-09%2011-03-24.png](https://github.com/megvagyhadnagy01/MagasProg2/blob/master/2_gyak/Anti%20OO/Screenshot%20from%202018-10-09%2011-03-24.png)

---

### **Ciklomatikus komplexitás:**

Még folyamatban.

---

### **Szülő-gyerek:**

Minden objektum referencia. Mindig dinamikus a kötés. De ezzel nem küldhetjük a Gyerek által hozott új üzeneteket. Csak akkor van dinamikus kötés, ha a viselkedés virtuálisra (virtual kulcsszó az ősből) van deklarálva. Ugyanúgy igaz, hogy ősz osztály referencián vagy pointeren keresztül, csak az ősz üzenetei küldhetőek.

Source code:

[https://github.com/megvagyhadnagy01/MagasProg2/tree/master/2\\_gyak/SzuloGyerek](https://github.com/megvagyhadnagy01/MagasProg2/tree/master/2_gyak/SzuloGyerek)

Futtatás:

[https://github.com/megvagyhadnagy01/MagasProg2/blob/master/2\\_gyak/SzuloGyerek/Screenshot%20from%202018-10-09%2016-41-16.png](https://github.com/megvagyhadnagy01/MagasProg2/blob/master/2_gyak/SzuloGyerek/Screenshot%20from%202018-10-09%2016-41-16.png)

---

### **Liskov:**

Az előadás foliája alapján igyekeztem megírni a kódot, melyben (Szülő\*szülő = new Gyerek();) öröklődés következtében melyben az ősökkel működő függvény működjön a gyermekkel is.

Source code:

[https://github.com/megvagyhadnagy01/MagasProg2/tree/master/2\\_gyak/liskov](https://github.com/megvagyhadnagy01/MagasProg2/tree/master/2_gyak/liskov)

Futtatás:

[https://github.com/megvagyhadnagy01/MagasProg2/blob/master/2\\_gyak/liskov/Screenshot%20from%202018-10-09%2011-13-12.png](https://github.com/megvagyhadnagy01/MagasProg2/blob/master/2_gyak/liskov/Screenshot%20from%202018-10-09%2011-13-12.png)

---

### 3.hét

#### **BPMN:**

A feladatot a bpmn.io oldalon hajtottam végre ahol egy pizzéria működéséről készítettem diagramot.

Source code:

[https://github.com/megvagyhadnagy01/MagasProg2/tree/master/3\\_gyak/BPMN](https://github.com/megvagyhadnagy01/MagasProg2/tree/master/3_gyak/BPMN)

Futtatás:

[https://github.com/megvagyhadnagy01/MagasProg2/blob/master/3\\_gyak/BPMN/diagram.svg](https://github.com/megvagyhadnagy01/MagasProg2/blob/master/3_gyak/BPMN/diagram.svg)

---

#### **Forward engineering UML osztálydiagram :**

Source code:

[https://github.com/megvagyhadnagy01/MagasProg2/tree/master/3\\_gyak/Foward](https://github.com/megvagyhadnagy01/MagasProg2/tree/master/3_gyak/Foward)

Futtatás:

[https://github.com/megvagyhadnagy01/MagasProg2/blob/master/3\\_gyak/Foward/Forwarduml.png](https://github.com/megvagyhadnagy01/MagasProg2/blob/master/3_gyak/Foward/Forwarduml.png)

---

#### **Reverse engineering UML osztálydiagram**

Source code:

[https://github.com/megvagyhadnagy01/MagasProg2/tree/master/3\\_gyak/Reverse%20engineering%20UML](https://github.com/megvagyhadnagy01/MagasProg2/tree/master/3_gyak/Reverse%20engineering%20UML)

Futtatás:

[https://github.com/megvagyhadnagy01/MagasProg2/blob/master/3\\_gyak/Reverse%20engineering%20UML/eredm%C3%A9ny.png](https://github.com/megvagyhadnagy01/MagasProg2/blob/master/3_gyak/Reverse%20engineering%20UML/eredm%C3%A9ny.png)

---

#### **“Egy esettan” :**

Igyekeztem értelmezni a megadott feladatokat és meglévő kódokat. Lefuttattam.

A BME-s C++ tankönyv 14. fejezetét (427-444 elmélet, 445-469 az esettan) dolgoztam fel.

Source code:

[https://github.com/megvagyhadnagy01/MagasProg2/tree/master/3\\_gyak/ProductManagement](https://github.com/megvagyhadnagy01/MagasProg2/tree/master/3_gyak/ProductManagement)

Futtatás:

[https://github.com/megvagyhadnagy01/MagasProg2/blob/master/3\\_gyak/ProductManagement/Screenshot%20from%202018-10-09%2011-45-36.png](https://github.com/megvagyhadnagy01/MagasProg2/blob/master/3_gyak/ProductManagement/Screenshot%20from%202018-10-09%2011-45-36.png)

---

#### 4.hét

##### **Encoding :**

A netbeans alap karakterkódolás Utf-8 a file ok pedig olyan karakterkódolást tartalmaznak mai nincs beállítva ezért át kell állítani windows-1252-re.Sikeresen lefordult a program.

Source code:

<https://github.com/megvagyhadnagy01/MagasProg2/tree/master/4.gyakorlat/Encoding>

Futtatás:

<https://github.com/megvagyhadnagy01/MagasProg2/blob/master/4.gyakorlat/Encoding/MandelBrot.PNG>

---

##### **Full screen:**

A feladat lényege java grafikus képességét szemlélteti.Frame függvény használatával karöltve.

Source code:

<https://github.com/megvagyhadnagy01/MagasProg2/tree/master/4.gyakorlat/Fullscreen>

Futtatás:<https://github.com/megvagyhadnagy01/MagasProg2/blob/master/4.gyakorlat/Fullscreen/Fullscreen.PNG>

---

##### **OOCWC lexer:**

A carlexer.Il fájl elején rövidítések vannak definiálva, amelyek értékei vagy konkrét karaktorsorozat ként, vagy reguláris kifejezésként vannak megadva.A lexer ezeket a rövidítéseket használja fel mintaként, ha pedig a megadott minták valamelyikével egyezést talál, akkor annak a szövegét megfelelő formátumúvá alakítja át,és ezt memóriacímen tárolja.

Source code:

<https://github.com/megvagyhadnagy01/MagasProg2/tree/master/4.gyakorlat/Leettranslator>

Futtatás:

<https://github.com/megvagyhadnagy01/MagasProg2/blob/master/4.gyakorlat/Leettranslator/Leettranslator.PNG>

---