

JEGYZŐKÖNYV

=====

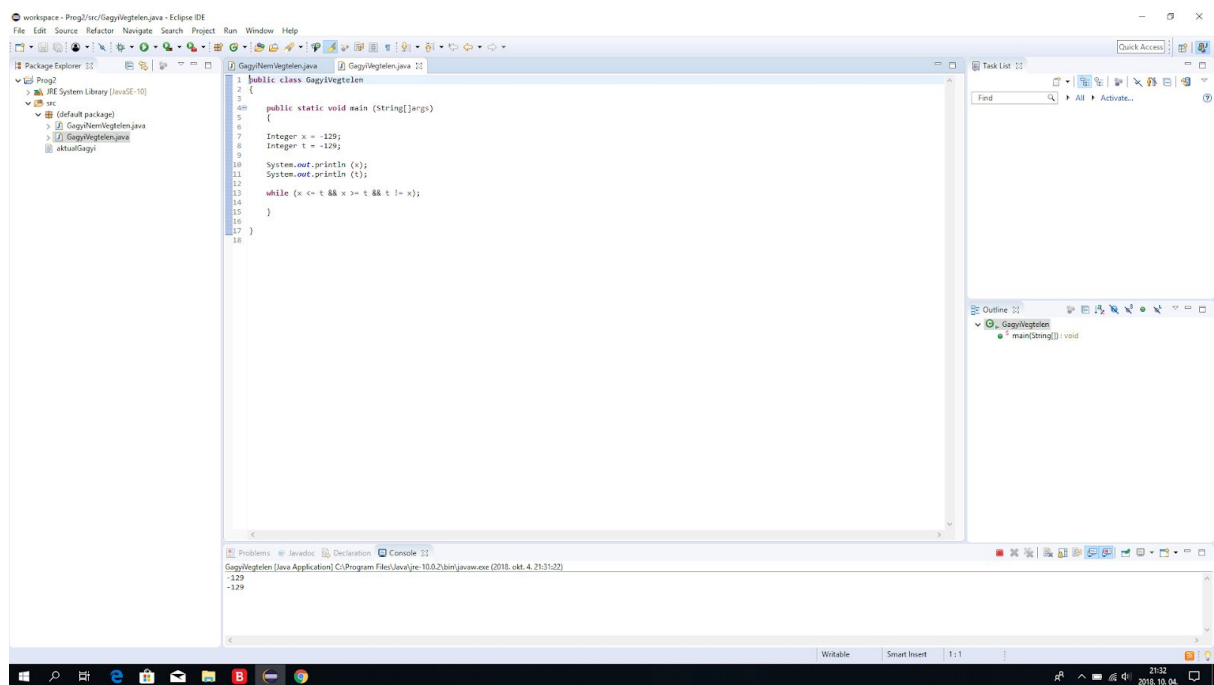
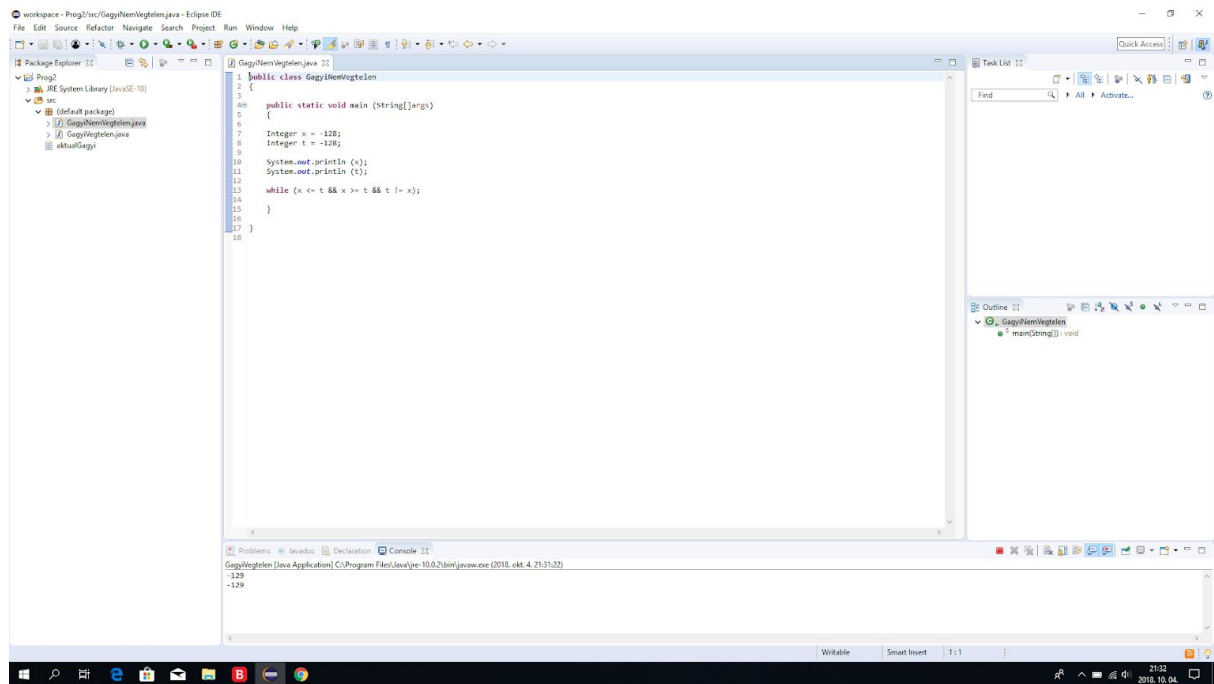
Varga Márton

=====

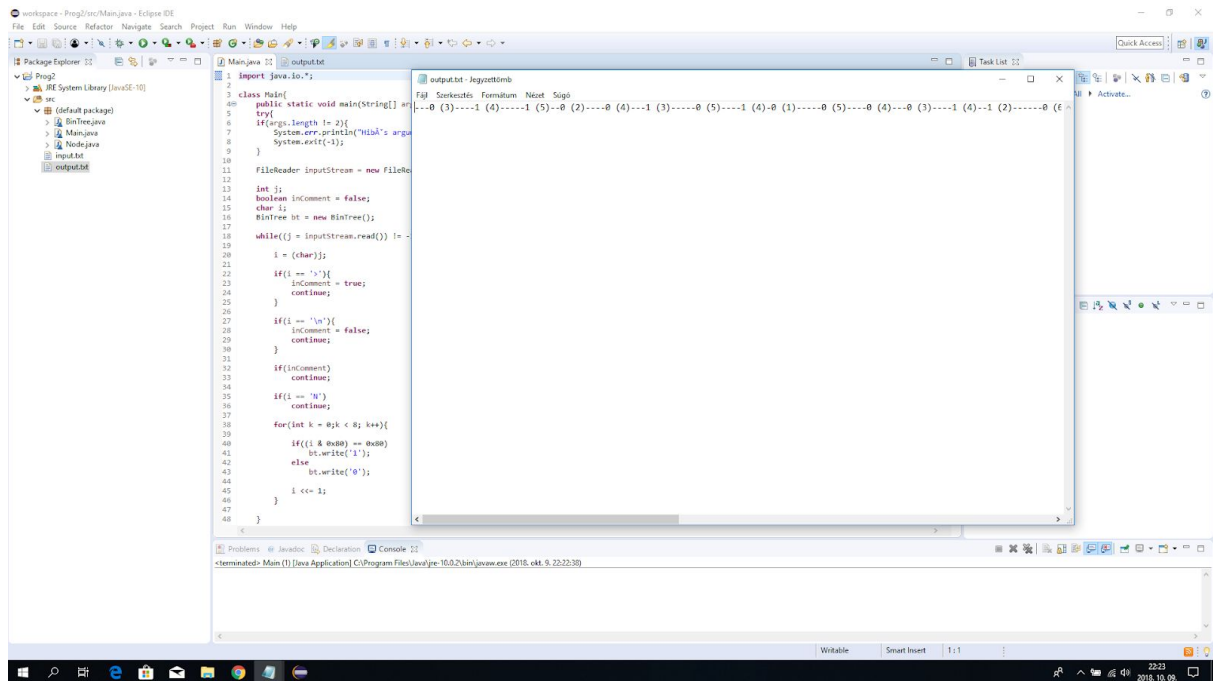
A feladatok fentt vannak a directoryban, hetekre bontva.

1.hét:

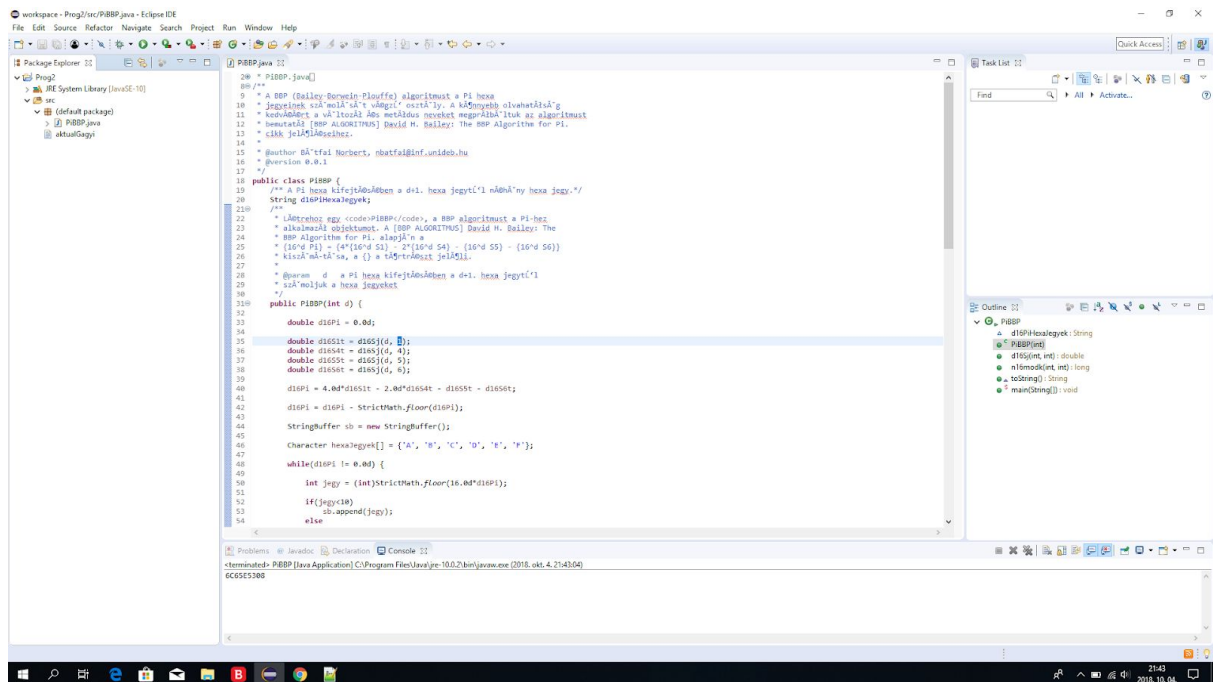
Gagy nem vegtelen es vegtelen:



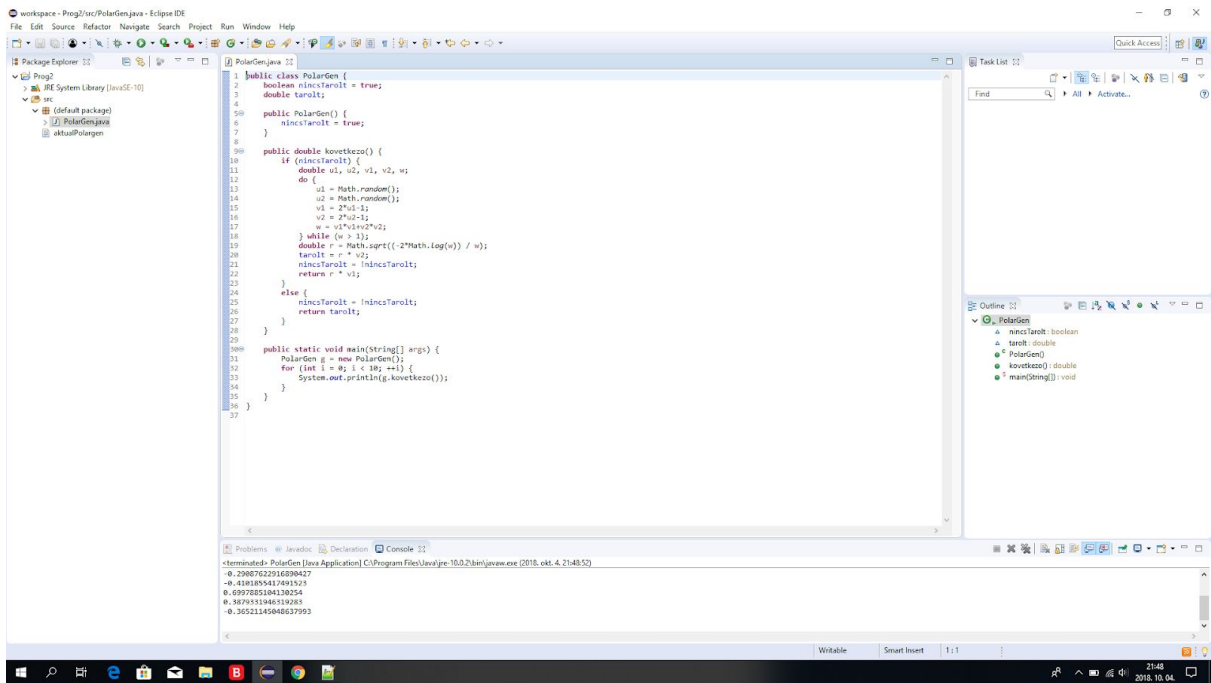
Homokozó:



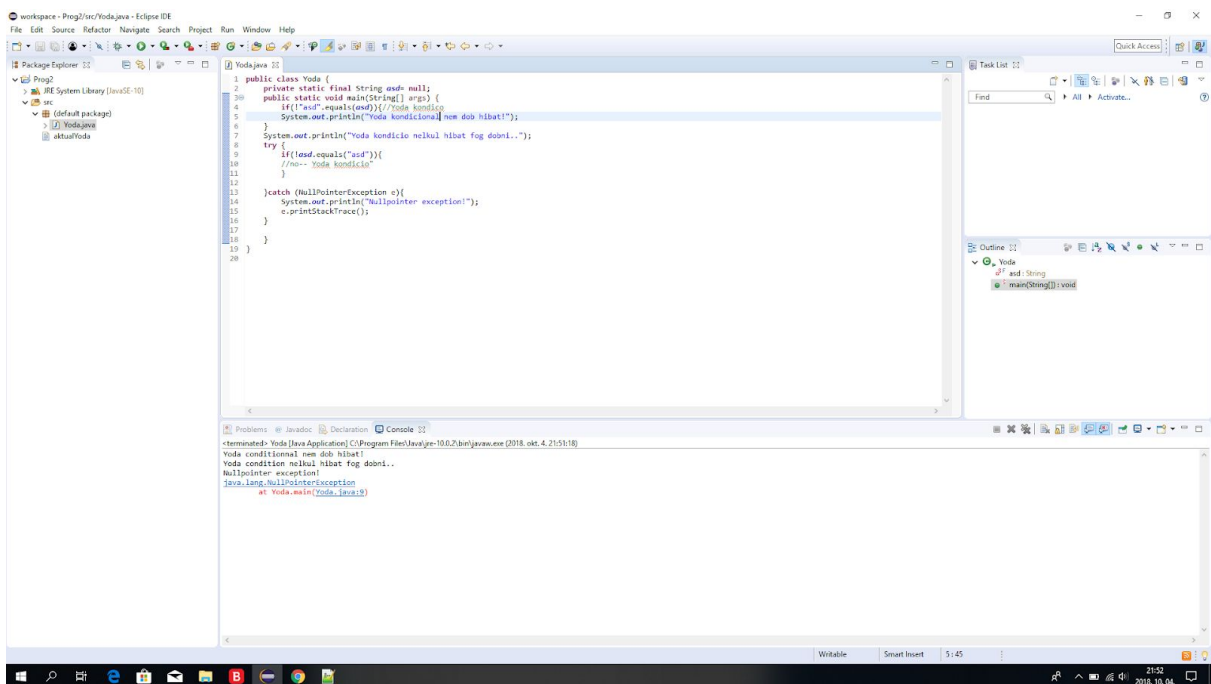
Kódolás from scratch:



OO szemlélet:



Yoda:



2.hét:

Liskov helyettesítés sértése:

```
1 // ez a T az LPS-ben
2 class Hader {
3     //public:
4     void repul();
5 }
6
7 // ez a két osztály alkotja a "P" programot az LPS-ben
8 class Program {
9     public:
10    void fgv ( Hader &hader ) {
11        // hader.repul(); a hader már nem tud repülni
12        // s hiba lesz a leírdmározott típusoknak
13        // repul metódusa, azt a Haderk hader-ra úgysem lehet hívni
14    }
15 }
16
17 // Itt jönnek az LPS-s 5 osztályok
18 class RepuloHader : public Hader {
19     public:
20     virtual void repul() {}
21 }
22
23 class Sas : public RepuloHader {
24 }
25
26 class Pingvin : public Hader // ezt úgy is lehet/kell olvasni, hogy a pingvin tud repülni
27 {}
28
29 int main ( int argc, char **argv )
30 {
31     Program program;
32     Hader hader;
33     program.fgv ( hader );
34     Sas sas;
35     program.fgv ( sas );
36     Pingvin pingvin;
37     program.fgv ( pingvin );
38 }
39
40
41
42
43
44
```

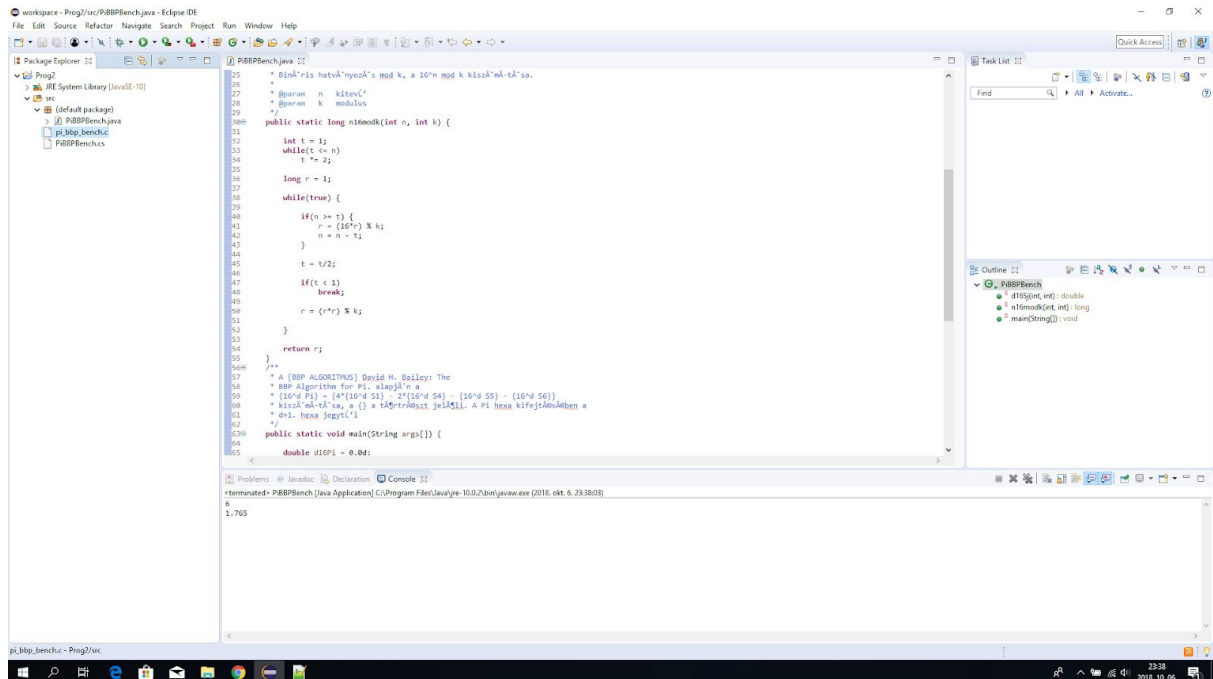
Get URL

Run

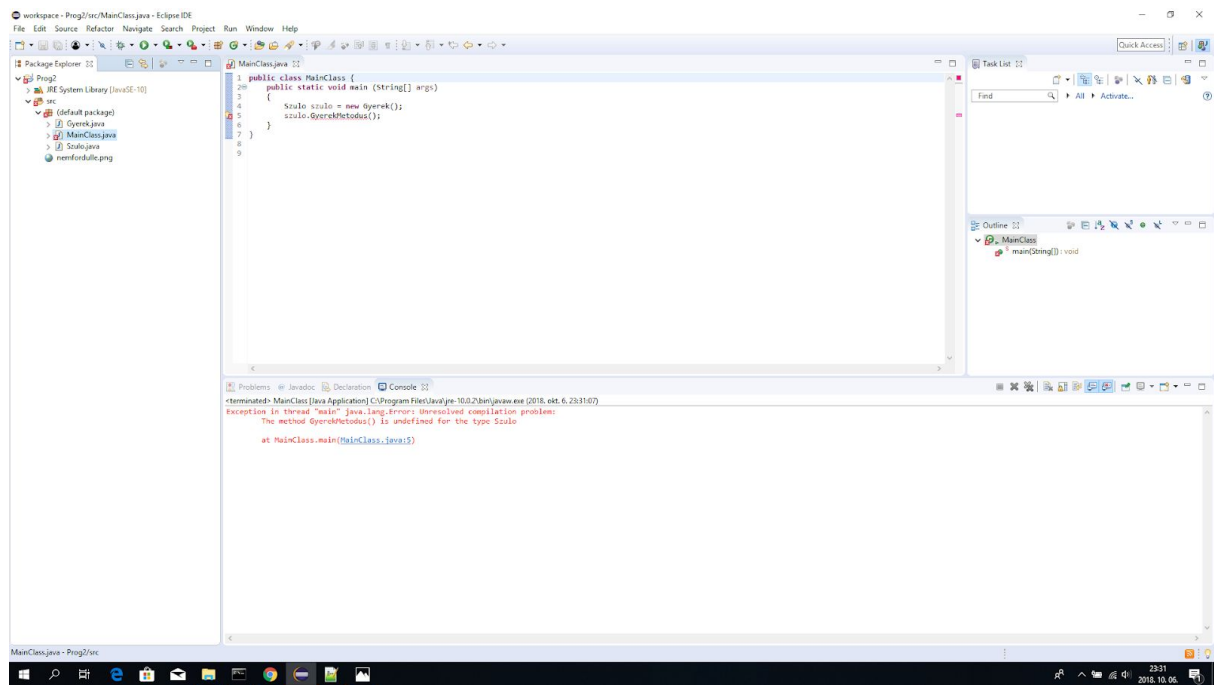
options | compilation | execution

Compilation successful

Anto OO:

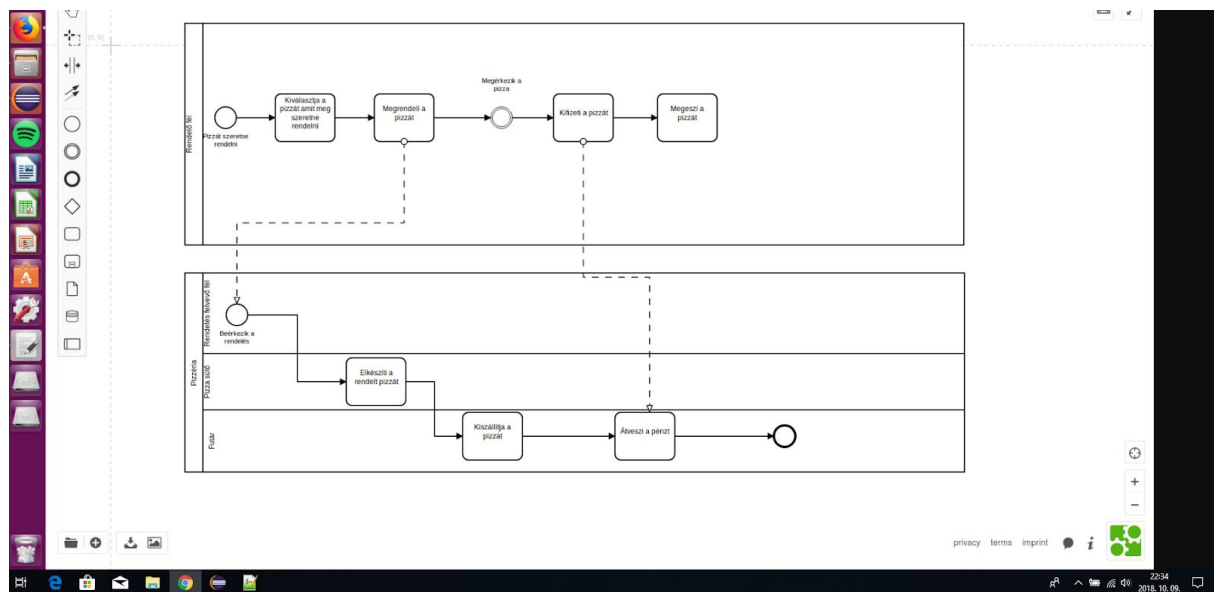


Szülő-gyerek:

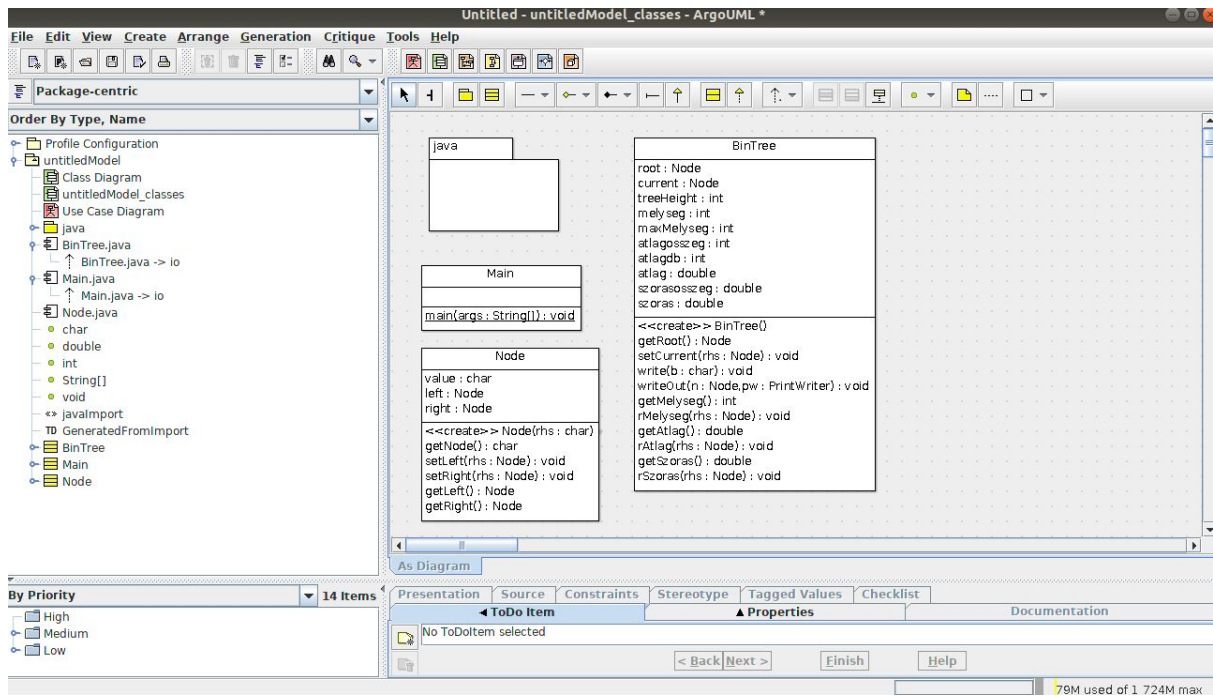


3.hét:

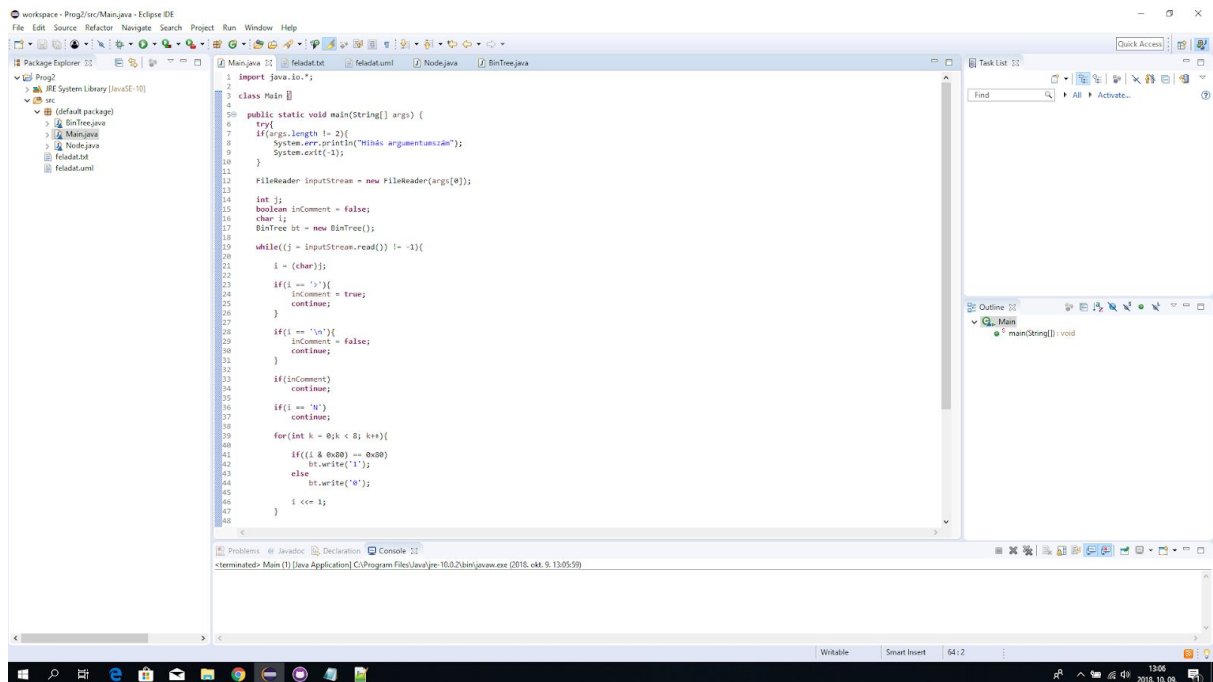
BPMN:



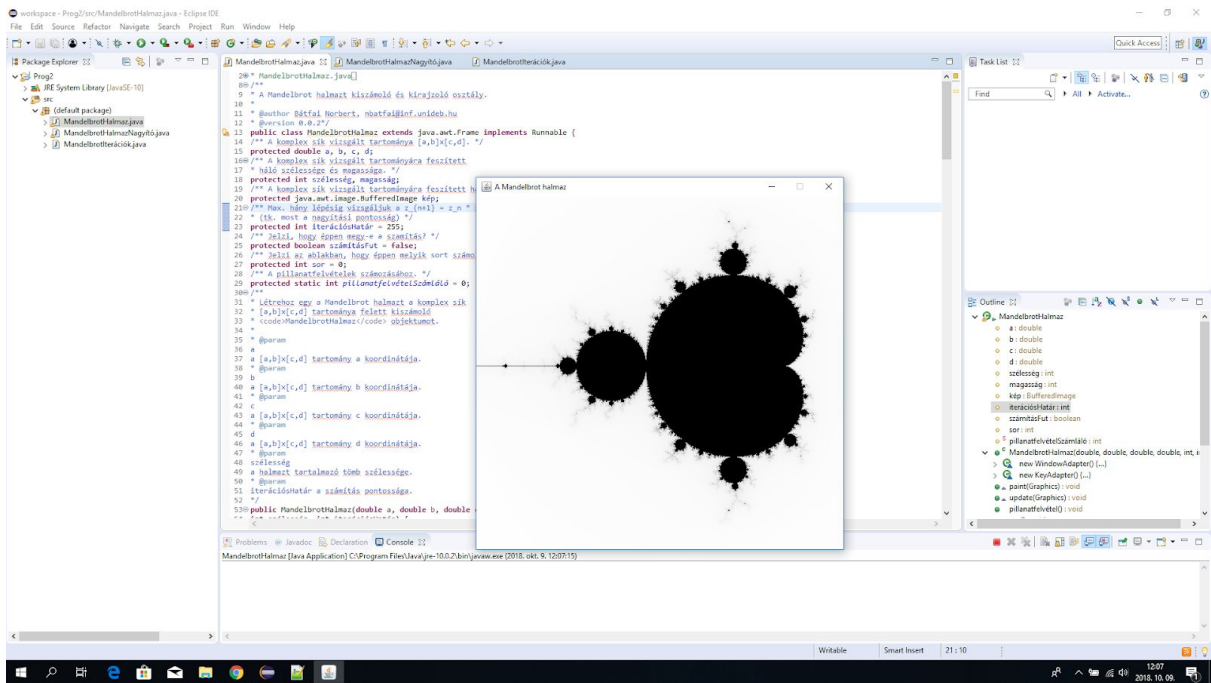
Reverse engineering UML:



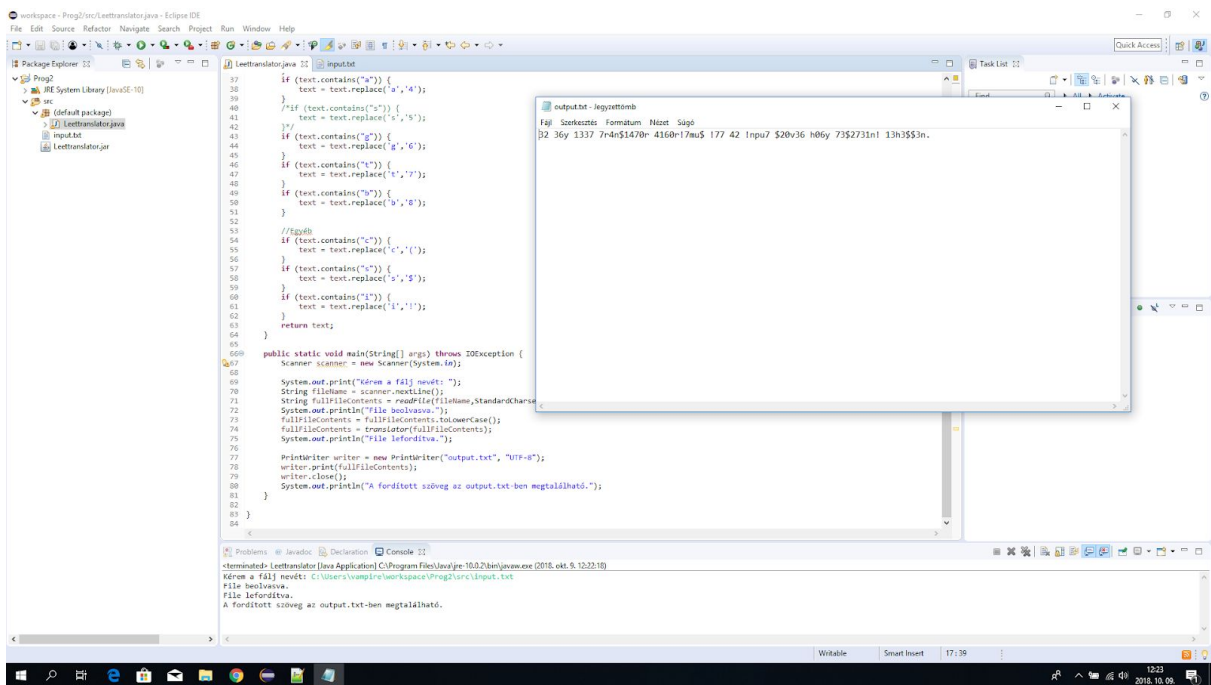
Forward engineering UML:



Encoding:



Leettranslator:



Full Screen:

