

Eclipse telepítése otthon

Elérhető minden operációs rendszerre.

Windowson Java JDK!!!(nem jre) kell hozzá.

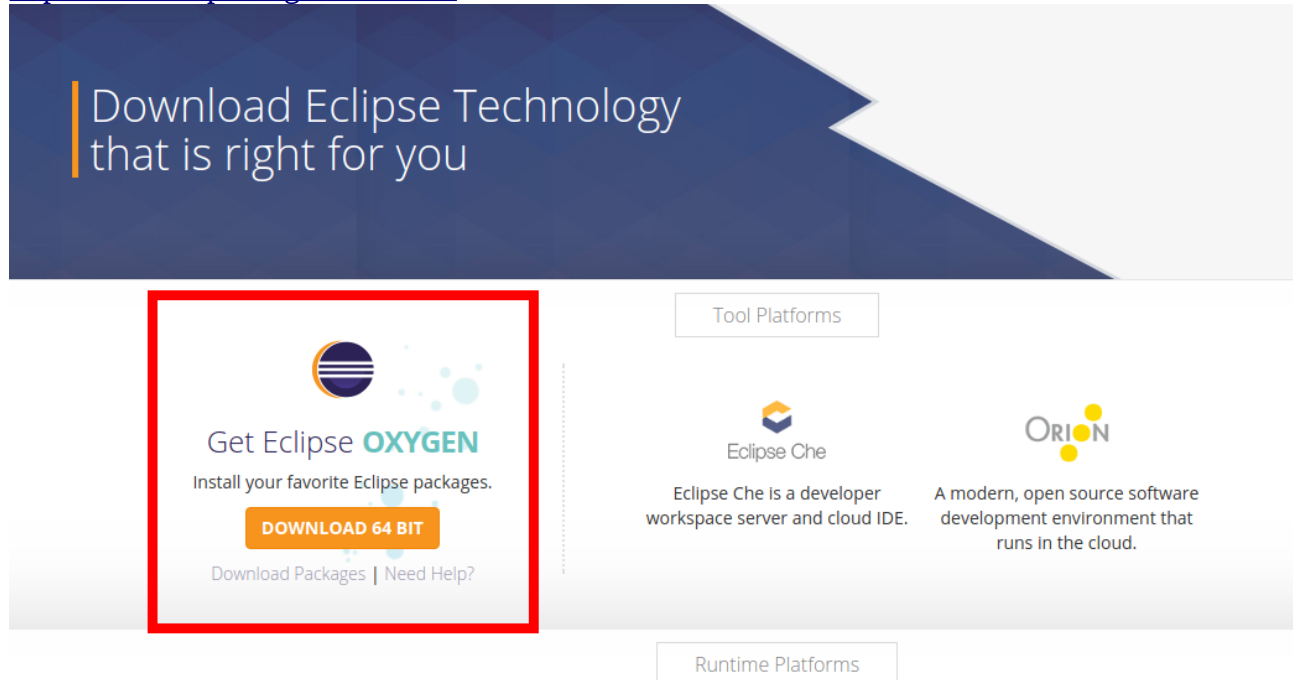
Linuxon vagy openjdk vagy hivatalos jdk(tutorial

<https://community.linuxmint.com/tutorial/view/1372>)

Java nélkül nem települ

Letöltés:

<http://www.eclipse.org/downloads/>



Eclipse EE-t célszerű telepíteni(a telepítő dobja fel, hogy melyiket szeretnénk).

Next next agree install szintű telepítője van.

Default workspace-be fogja tenni a programokat, nem érdemes ssd-re tenni.

Linuxon ikont a menübe nekünk kell készíteni, DE-nként eltérő

Git alapok

Konzolos

Nincs minden fejlesztőkörnyezetbe normálisan integrálva a git, érdemes megtanulni a konzolos parancsokat.

apt-gettel települ linuxra

Windowsra innen telepíthető: <https://git-scm.com/downloads>

Repository letöltése (akár saját, akár másé):

git clone url (pl git clone <http://github.com/ttspeaker88/oopJava.git>)

letölti a terminál aktuális mappájában és létrehozza az oopJava mappát

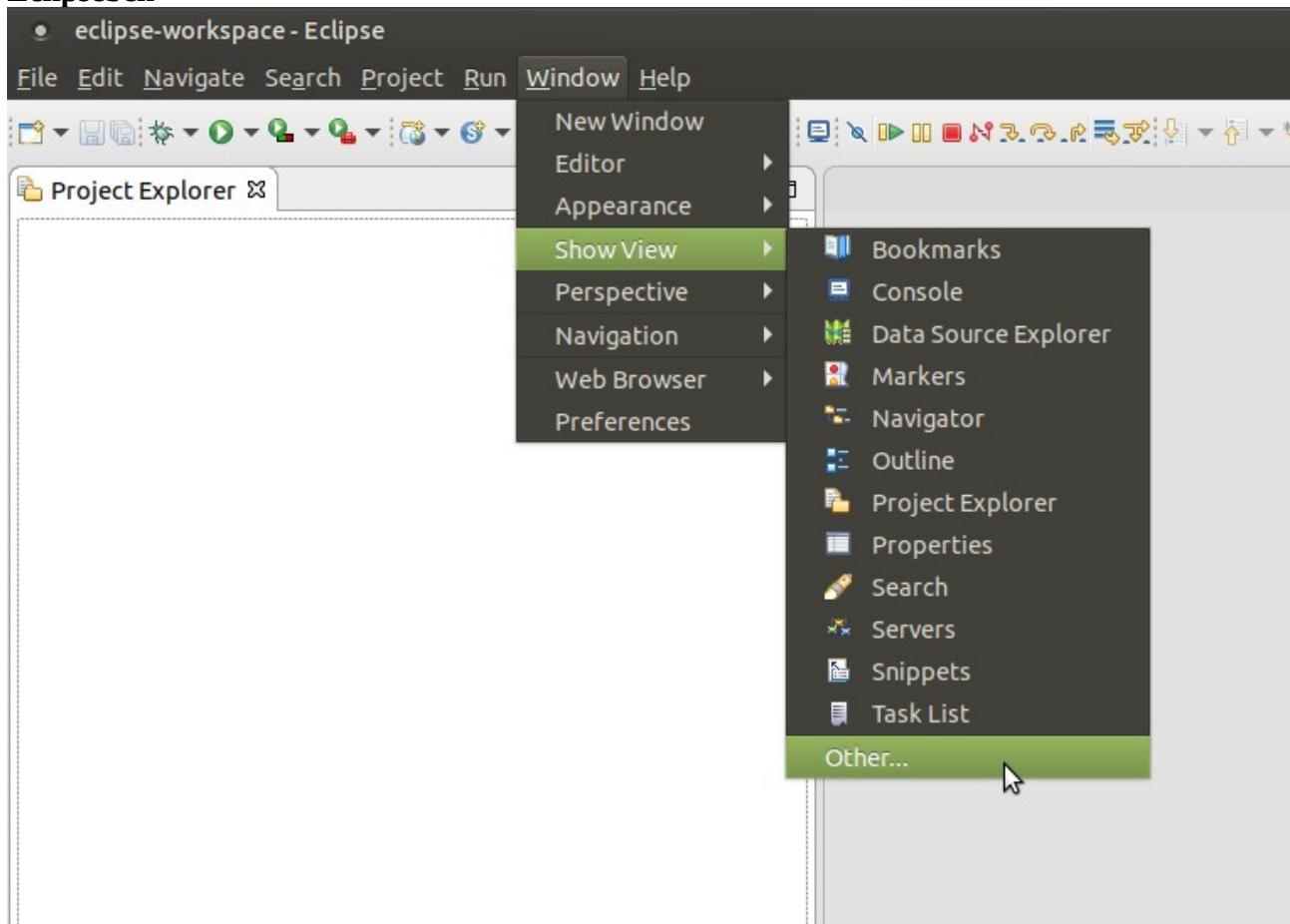
saját repoba feltöltés

git add . (vagy git add fájlnev, a “.” hozzáad mindent abban a mappában)

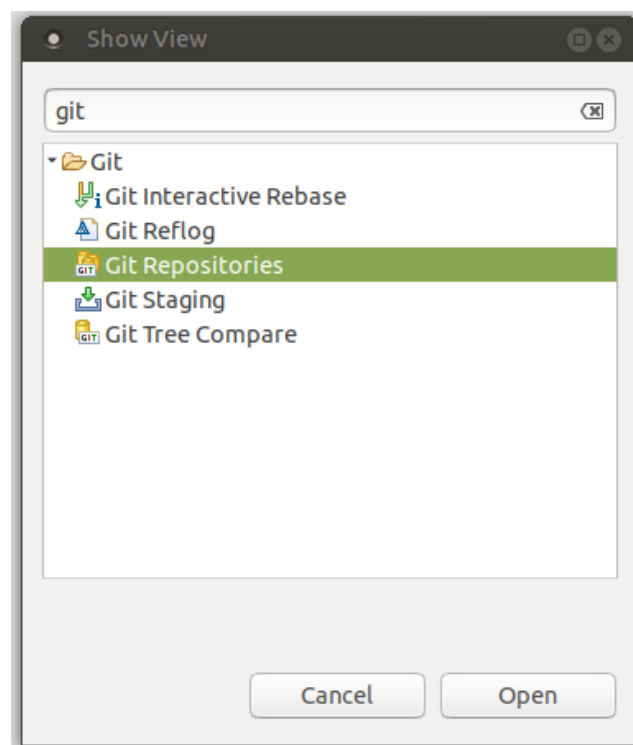
git commit -m “üzenet” (készít egy commitot, üzenet néven. Az üzenet azt tartalmazza hogy mi változott)

git push (majd meg kell adni a github-os felhasználónév/jelszó párost(kérni fogja))

Eclipseben



Window/Show View/ Otherssearch/git → Git Repositories

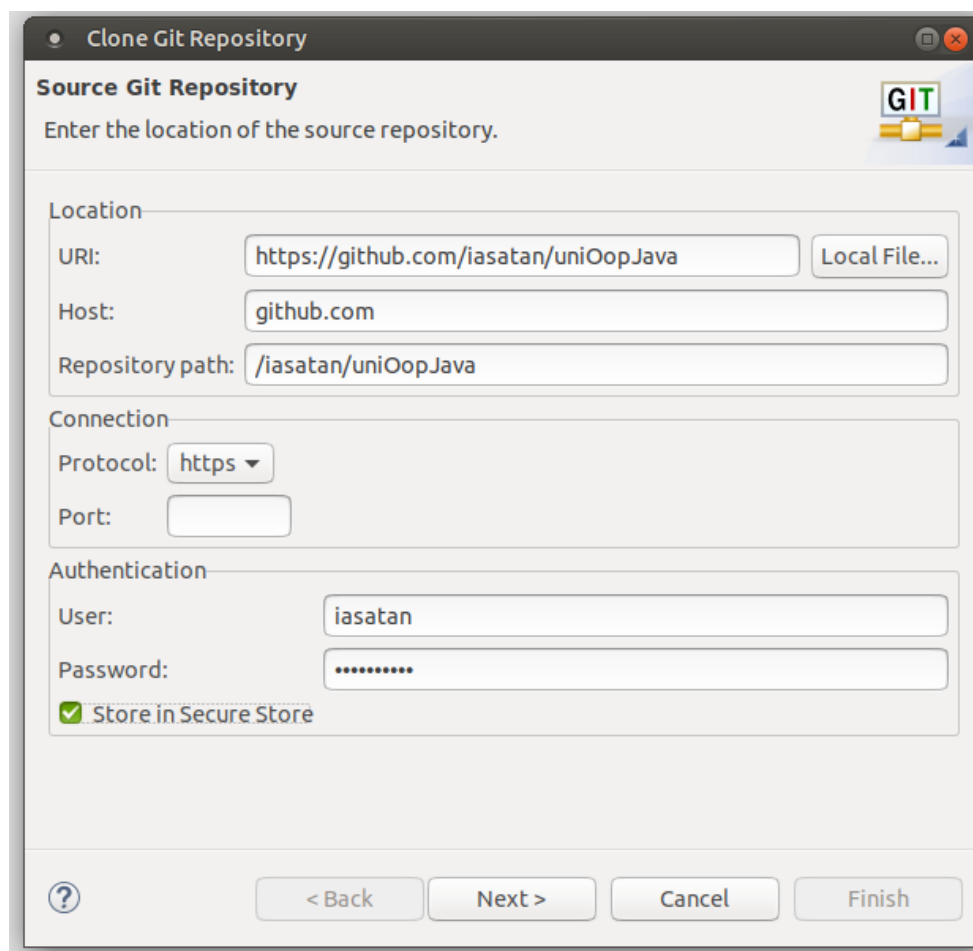


Lent megjelent egy új fül

Clone a Git repository-val lehet letölteni githubról repokat.

Fenti minta alapján kell kitölteni

Jelszót érdemes lehet eltárolni, de nem kötelező



next

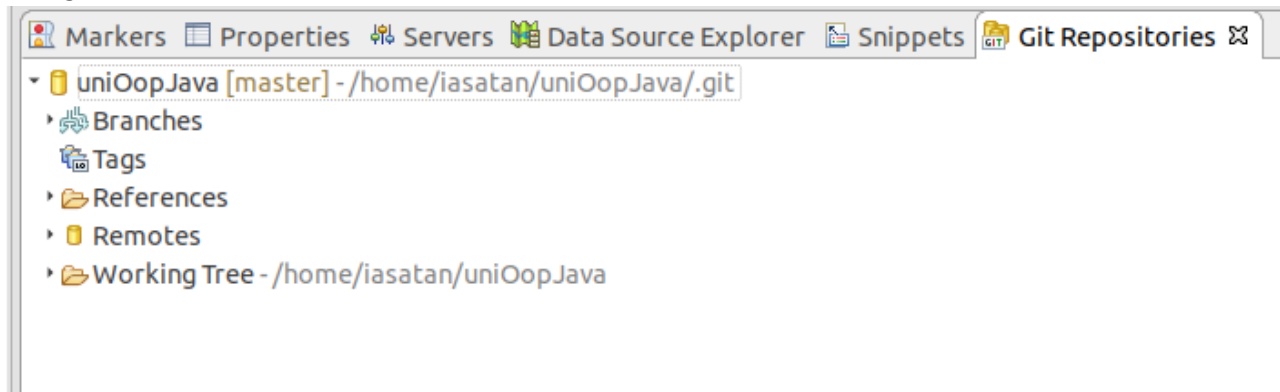
Kiválasztjuk a master-t(alapból ki van)

next

Megadjuk, hogy hova töltsse le a repository-t a fájlrendszeren

Érdemes lehet az eclipse-es workspace mappát választani, hogy egy helyen legyen minden

Finnish

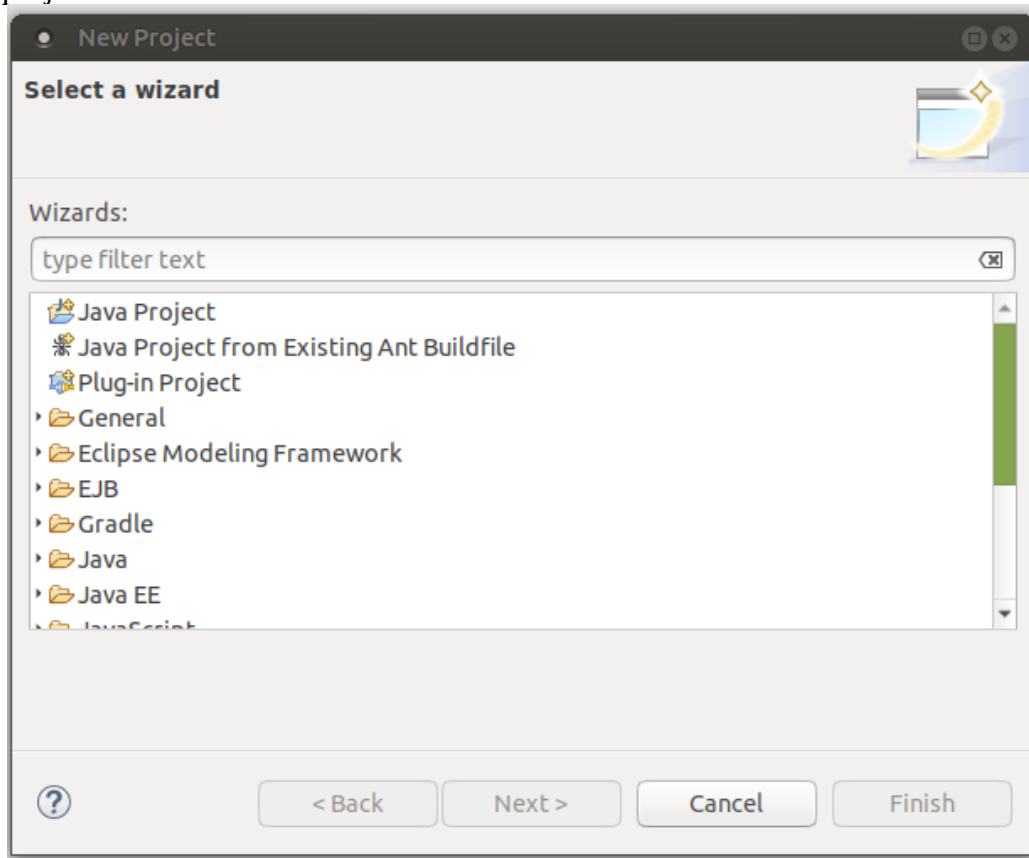


Megjelent a repo lent, innen lehet majd grafikus interfacel feltölteni a készülő programokat.

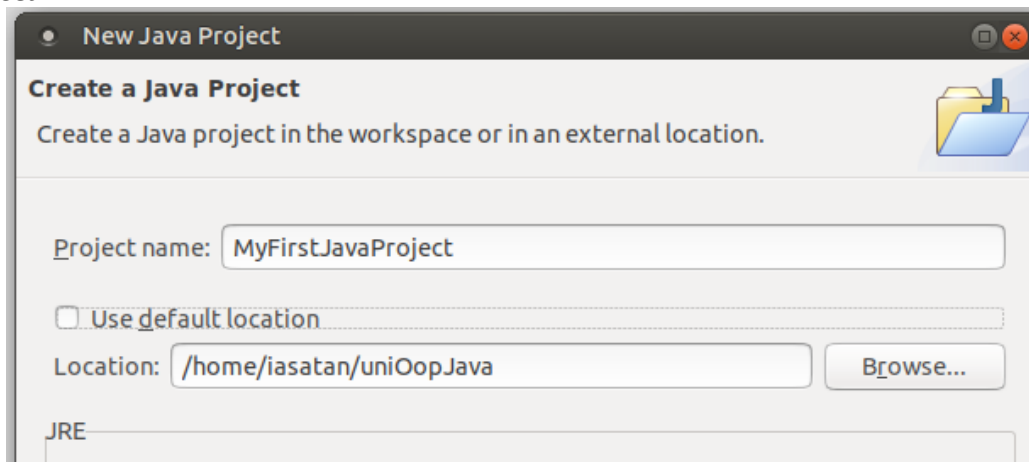
Új java projekt

Mivel a repo kezdetben üres, ezért létre kell hozni egyet

File/new/project

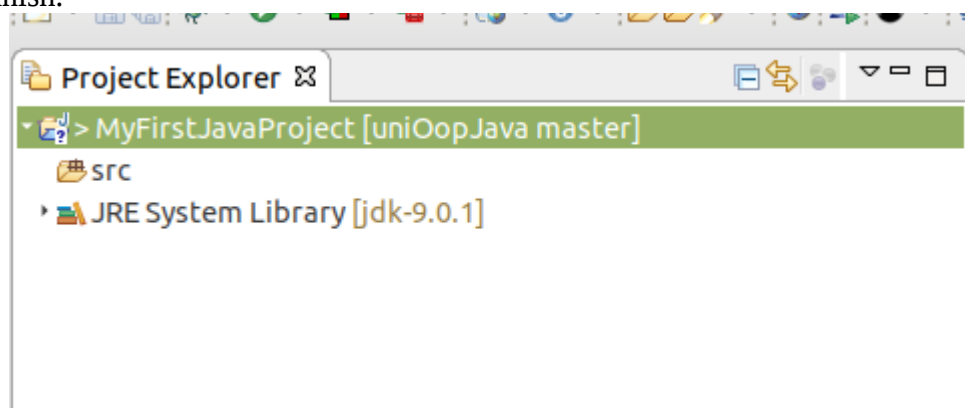


Java project



Elnevezzük a projektet, és megadjuk a git-es repot helynek.

Next next finish.



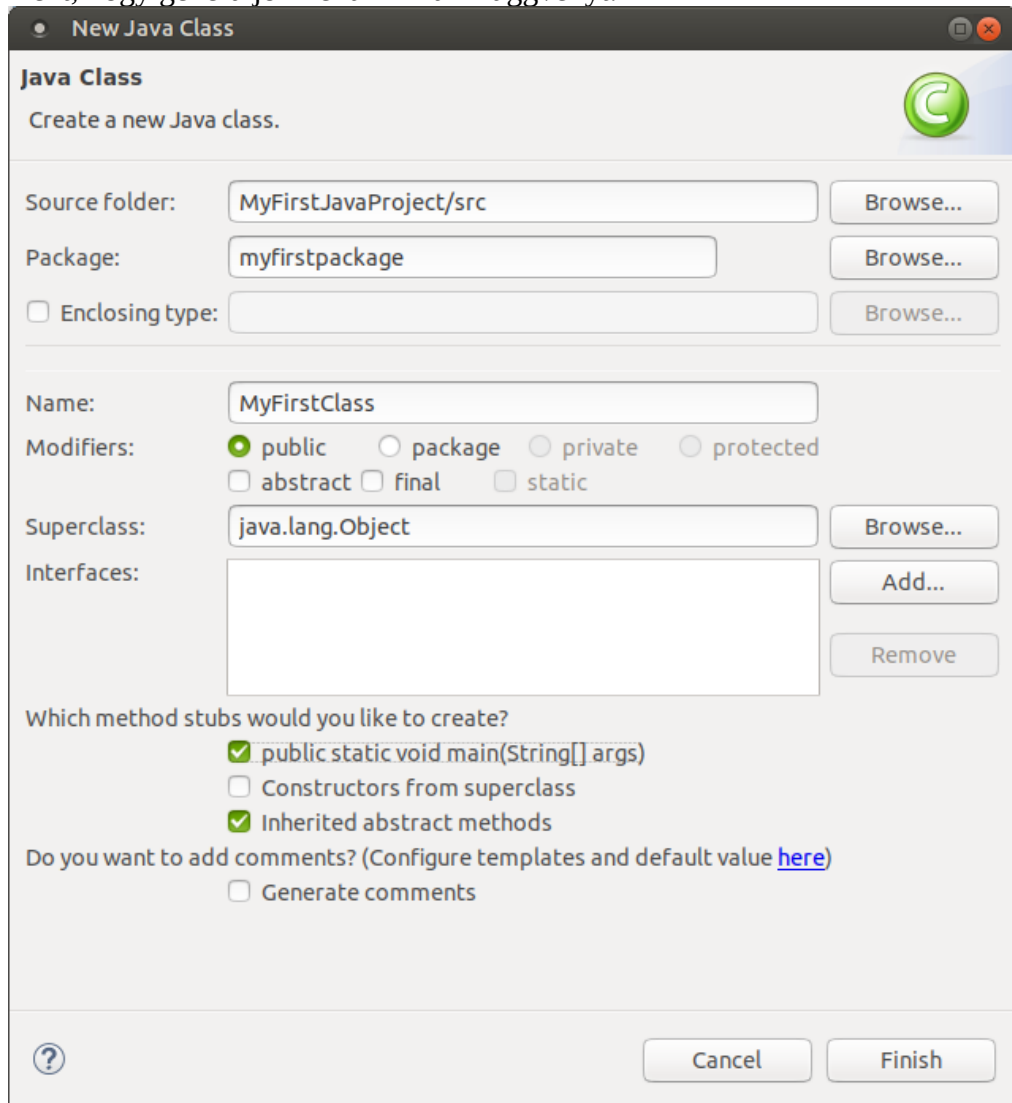
Megjelent az első Java projectünk bal oldalt.

A project még üres, szükség van egy package-re(csomag), és egy Osztályra src, jobbgomb, new class.

Megadjuk az osztály csomagját. Csupa kisbetűvel

Megadjuk az osztály nevét. Minden szó nagybetűvel kezdődik és egybe írjuk.

Bepipáljuk lent, hogy generáljon nekünk Main függvényt.



Finish.

Megvan az első osztályunk:

```
MyFirstClass.java ✕
1  package myfirstpackage;
2
3  public class MyFirstClass {
4
5      public static void main(String[] args) {
6          // TODO Auto-generated method stub
7      }
8  }
9
10 }
11
```

Készítsünk belőle egy Hello Világot.

```
MyFirstClass.java
1 package myfirstpackage;
2
3 public class MyFirstClass {
4
5     public static void main(String[] args) {
6         String message = "Hello World";
7         System.out.println(message);
8     }
9 }
10
11 }
12
```

A *String* a változó típusa(szöveg).

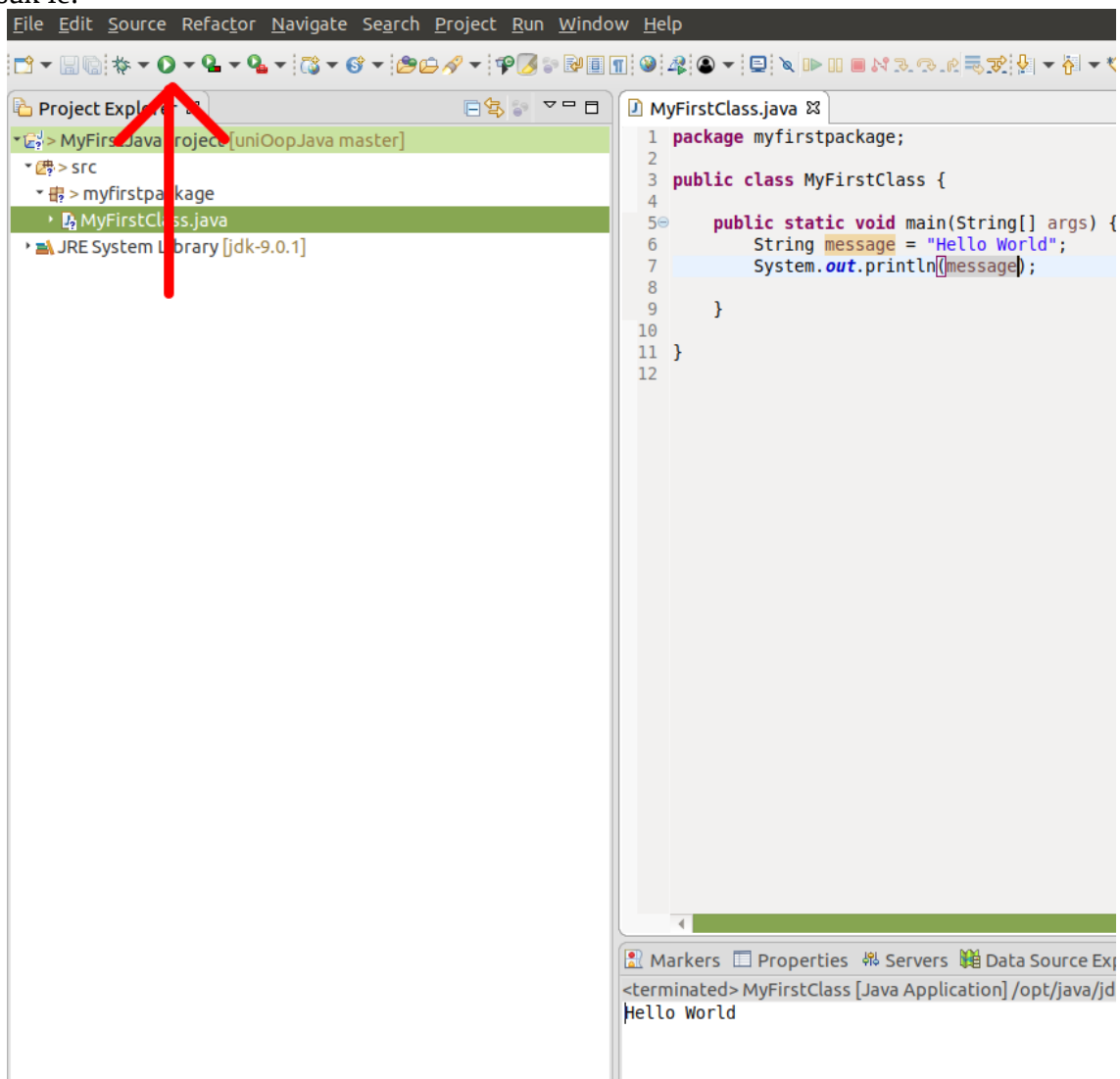
A *message* a változó neve amivel hivatkozni lehet rá.

A "Hello World" pedig a kezdőértéke.

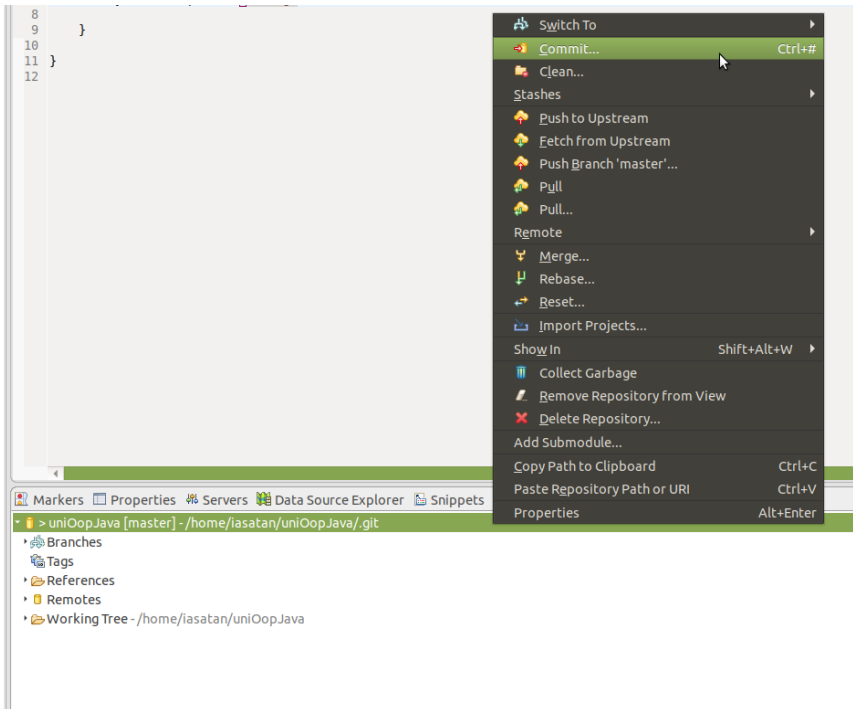
Ezzel létrehozuk az első Objektumunkat.

System.out.println(szöveg); segítségével lehet kiírni valamit a konzolra.

Futtasuk le:

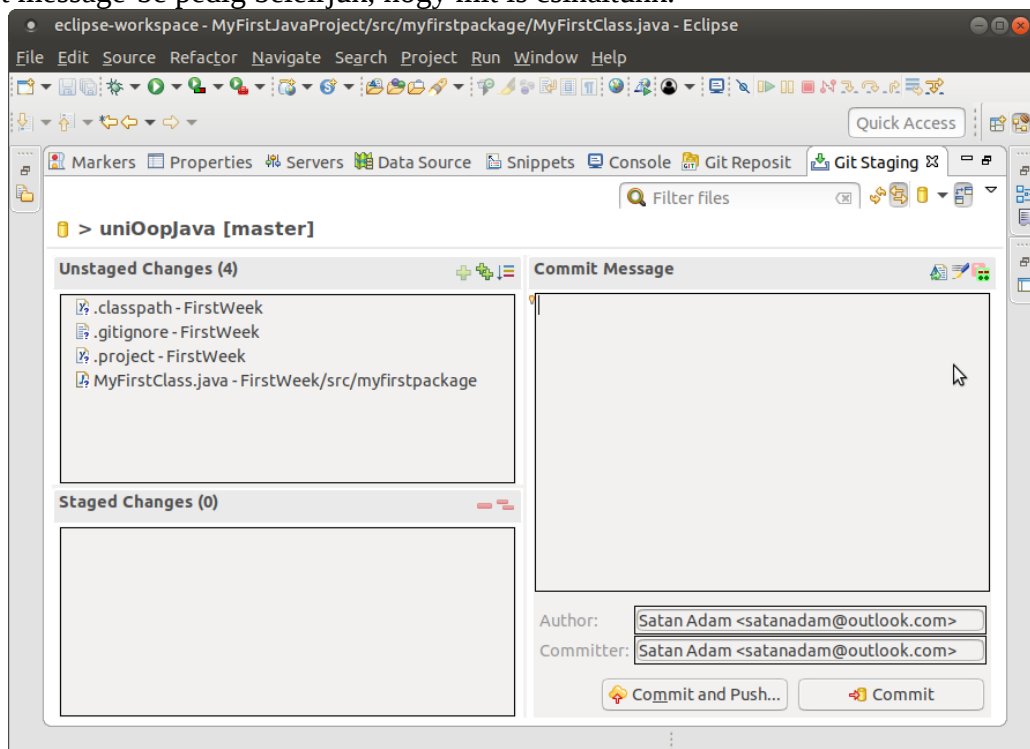


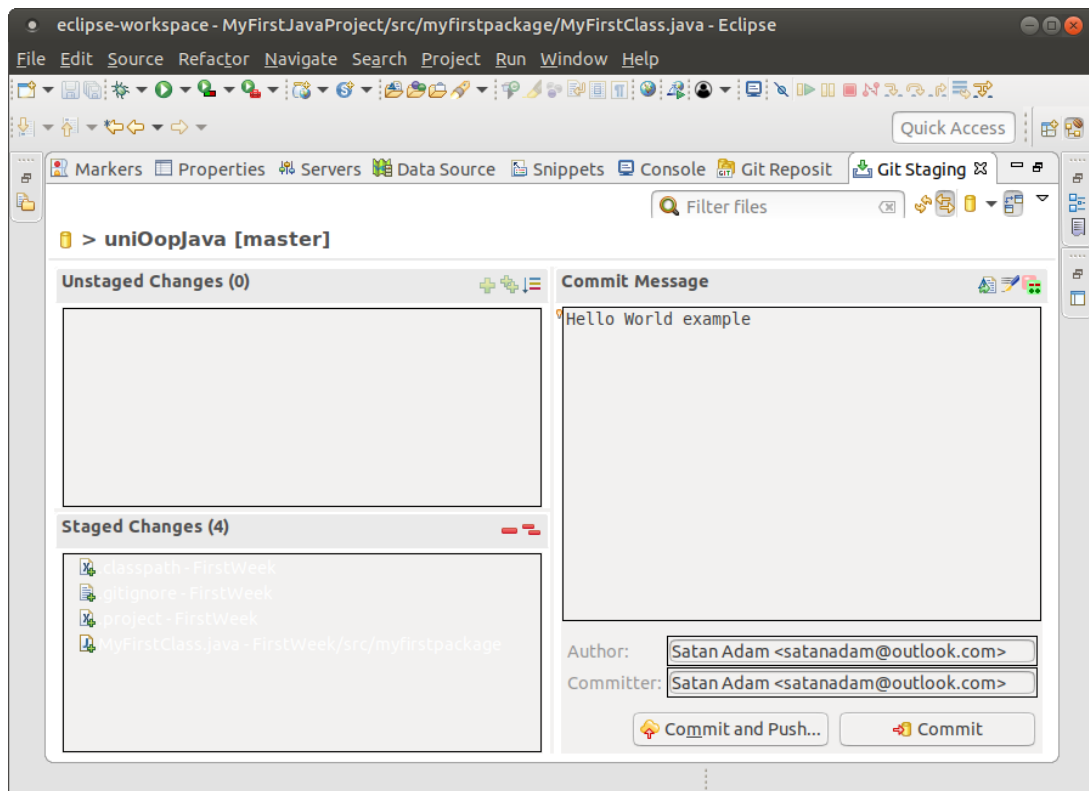
Kírtá hogy Hello World, a programunk elkészült.
Töltsük fel github-ra.



Megjelent egy új fül lent Git Staging néven
Bal oldalon fent láthatóak azok a fájlok amik még nincsenek feltöltve gitre, vagy nem a legfrissebb változat van fent.

Kijelöljük mindet(ctrl+a), és drag&drop-al áthuzzuk ball lentre
A commit message-be pedig beleírjuk, hogy mit is csináltunk.

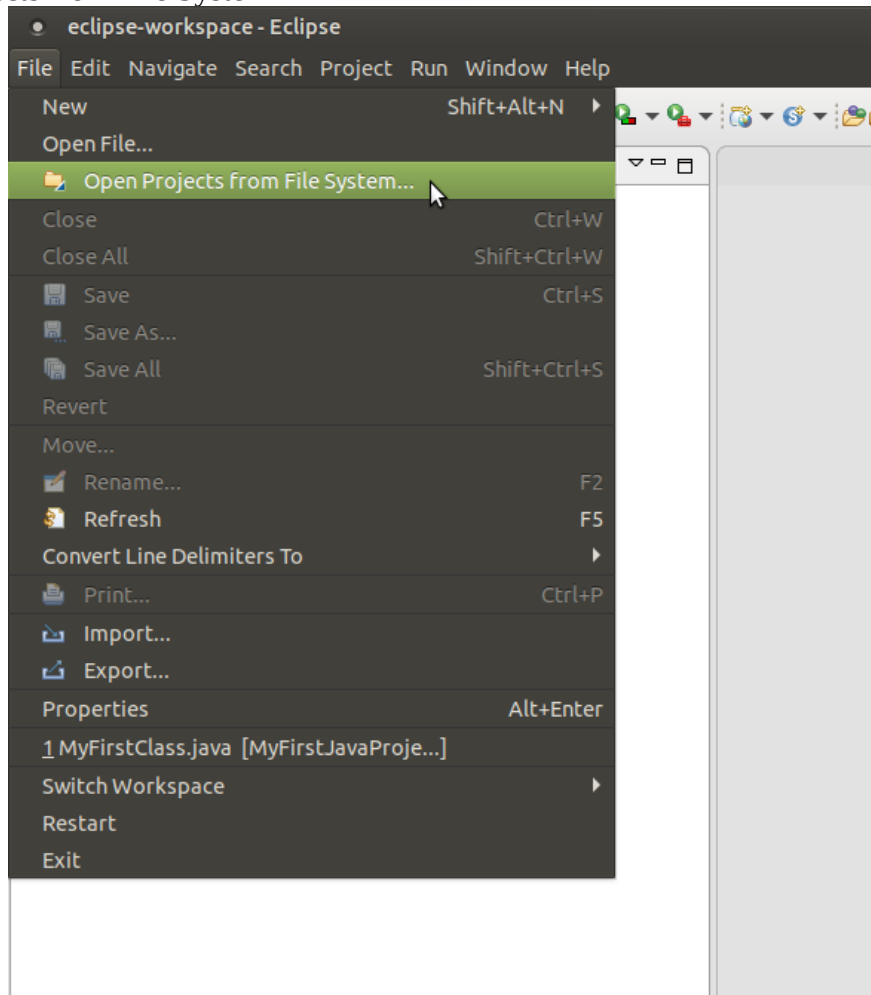




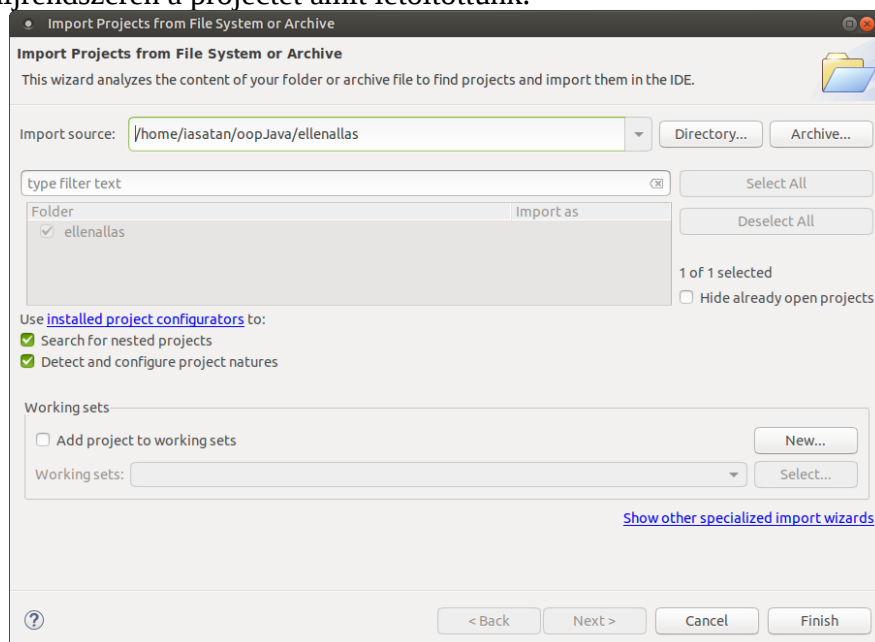
Commit and Push-al feltölti egyből githubra, simán Committal elmenti, de még nem tölti fel. Egyszerre több commitot is fel lehet tölteni

Létező project megnyitása

File/Open Projects from File System



Kikeressük a fájlrendszeren a projectet amit letöltöttünk.



Finish

Javadoc generálása

```
/**
 * main metodus
 * @param args
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                Foablak frame = new Foablak();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}
```

A kódban találhatóak kommentek, ezekből lehet generálni Javadoc-ot.

Saját komment írása:

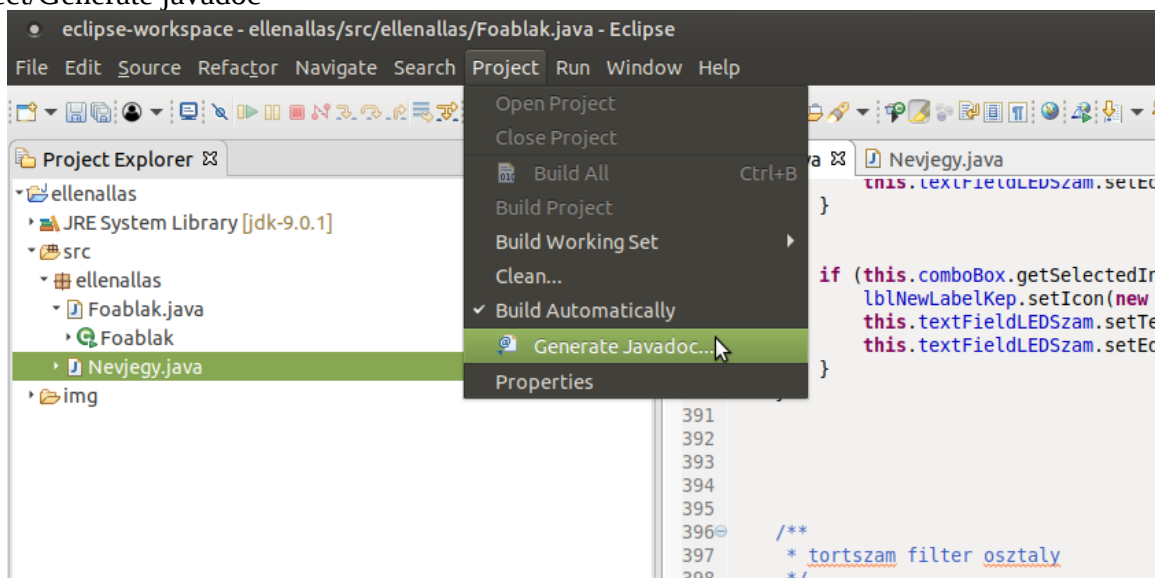
Egysoros komment: //

többsoros komment: /* több sor */

javadoc komment: /**
 * leírás
 * @param paraméterek
 * @return visszatérési érték
 */

Javadoc generálása kommentekből:

Project/Generate javadoc



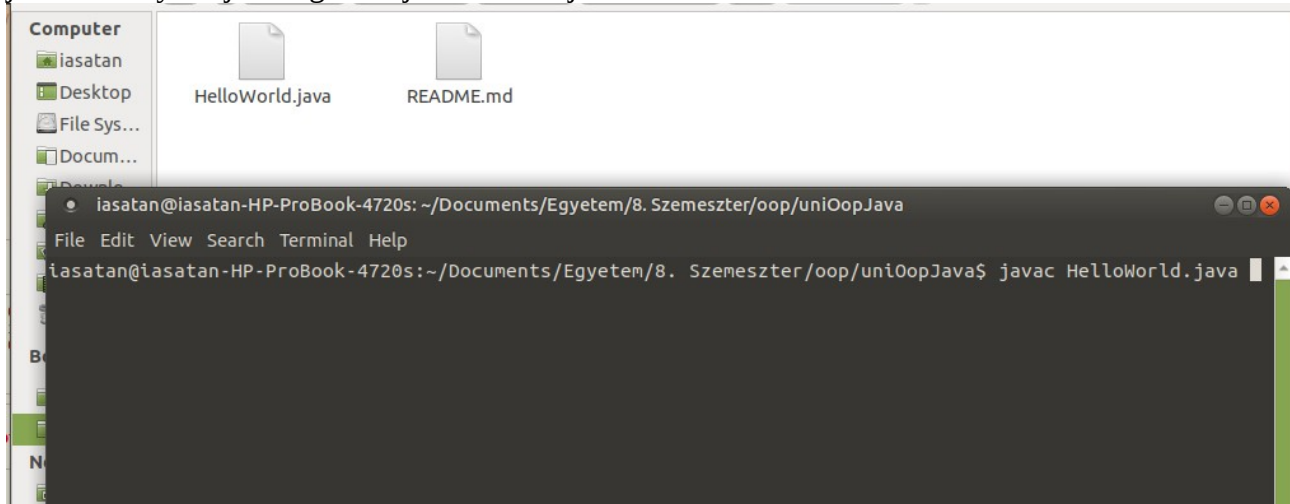
next next finish

Elkészült a javadoc, amit a megadott mappába mentett.

Böngészővel lehet megnézni.

Fordítás és futtatás terminálból

javac osztálynév.java legenerálja a .class fájlt



Java className futtatja a lefordított fájlt

