Introducere în dezvoltarea de programe în limbajul Java folosind Eclipse

1. Scopul lucrării

Obiectivele acestei sesiuni de laborator sunt:

- Instalarea și familiarizarea cu mediul de dezvoltare Eclipse și cu facilitățile oferite de acesta
- Crearea, înțelegerea și depanarea câtorva aplicații Java simple
- Vizualizarea diagramei UML de clase a unei aplicații Java
- Crearea documentației unei aplicații Java
- Crearea și rularea unei aplicații de sine stătătoare în Java (fișier executabil)

2. Instalarea mediului de dezvoltare Eclipse și a plugin-ul ModelGoon

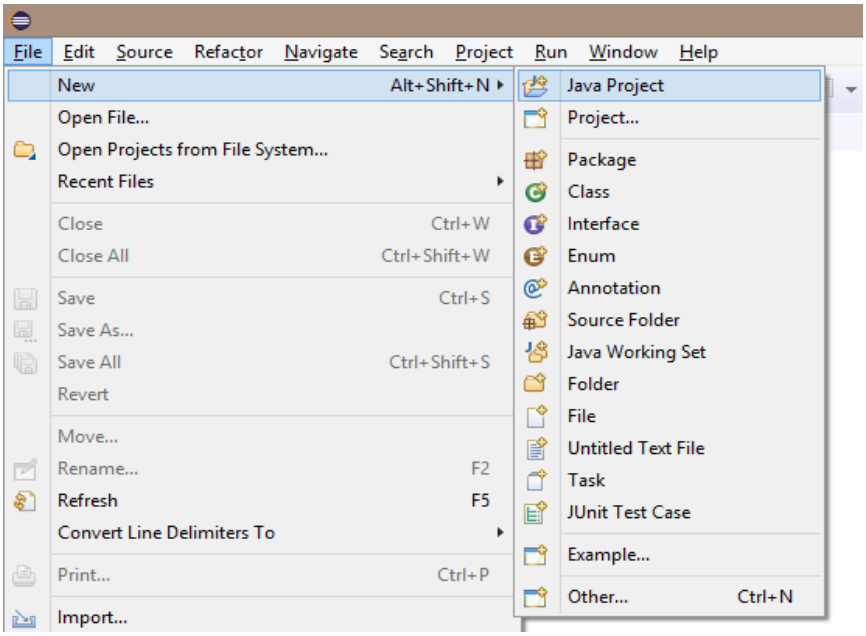
Pentru a putea executa programe scrise în limbajul Java trebuie să aveți instalat Java Runtime Environment (JRE). Începând cu versiuni mai noi, pachetul de instalare al mediului de dezvoltare Eclipse conține deja și JRE. Instalați Eclipse (cu JRE) iar apoi eventualele extensii (plugin-uri) pentru acesta.

1. Descărcați și instalați Eclipse cu JRE (este suficient să alegeți Eclipse IDE for Java Developers) accesând site-ul: https://eclipse.org/downloads/
2. Instalați în Eclipse plugin-ul ModelGoon care vă permite să vizualizați diagrama de clase corespunzătoare unui proiect Java - detalii se găsesc în secțiunea Installation pe site-ul: http://www.modelgoon.org/

Pentru a vă putea stoca proiectele personale într-un singur loc și pentru a le putea regăsi și folosi ușor, de fiecare dată când porniți Eclipse, setați directorul personal de lucru accesând meniul File -> Switch Workspace -> Other… și selectați directorul propriu de lucru.

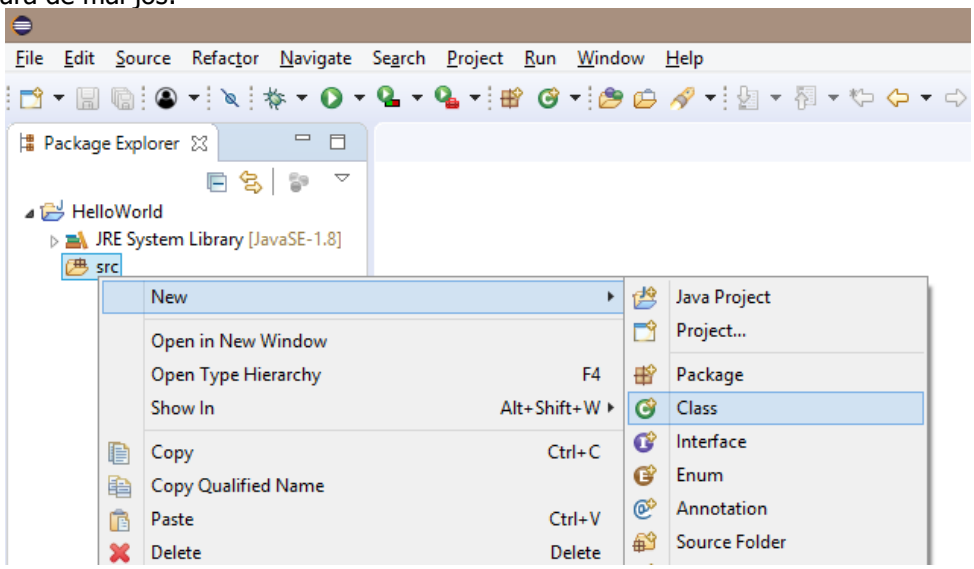
3. Crearea unei aplicații simple "Hello world"

3.1. Creați un proiect nou accesând meniul: File -> New -> Java Project

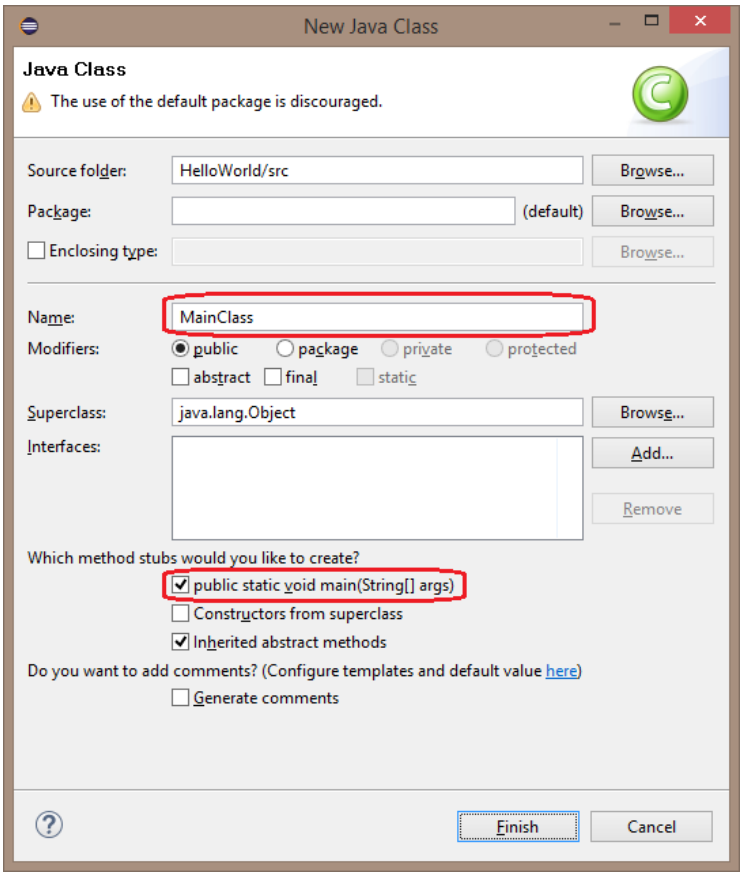


3.2. Dati un nume proiectului (Hello World), selectați directorul unde vreți să-l salvați pe disc (este recomandat să lăsați directorul de lucru curent selectat la pornirea Eclipse!), debifați (dacă există) *Create module-info.java file*, apoi apăsați butonul „Finish".

3.3. În proiectul creat, dați click dreapta pe pachetul *src* creat automat și creați o clasă nouă, ca în figura de mai jos:



3.4. Dați un nume clasei pe care o creați (numele claselor vor începe întotdeauna cu majusculă!). Puteți selecta opțiunea de a crea automat metoda *main* - ce va fi executată automat la rularea programului:



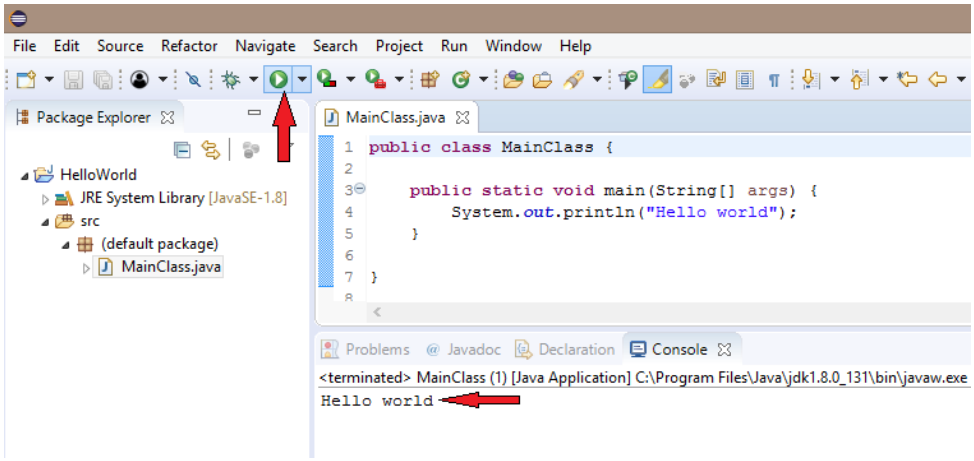
La apăsarea butonului Finish se va crea automat fișierul MainClass.java cu următorul cod sursă:

```
public class MainClass {
	public static void main(String[] args) {
		// TODO Auto-generated method stub
	}
}
```

3.5. Vom scrie codul sursă care dorim să se execute în metoda *main*. De exemplu, pentru afișarea mesajului "Hello world", vom scrie următoarea linie de cod:

```
System.out.println("Hello world");
```

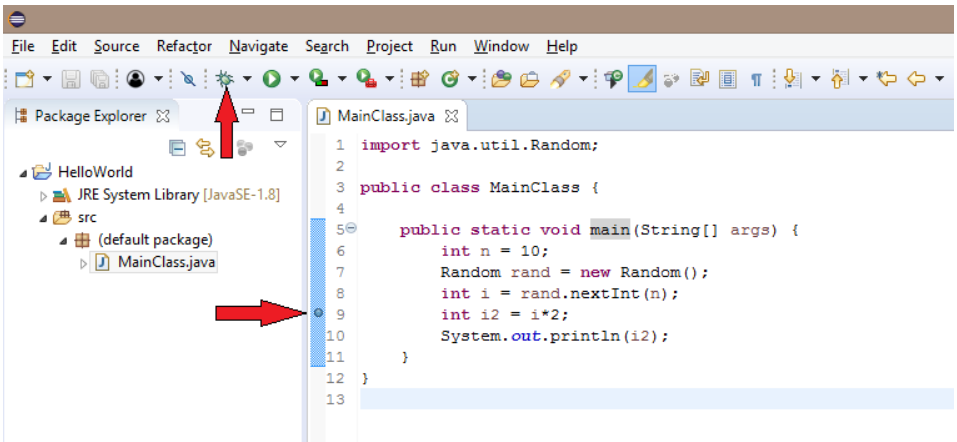
3.6. La rularea programului, rezultatul se afișează în consolă, ca în figura alăturată:



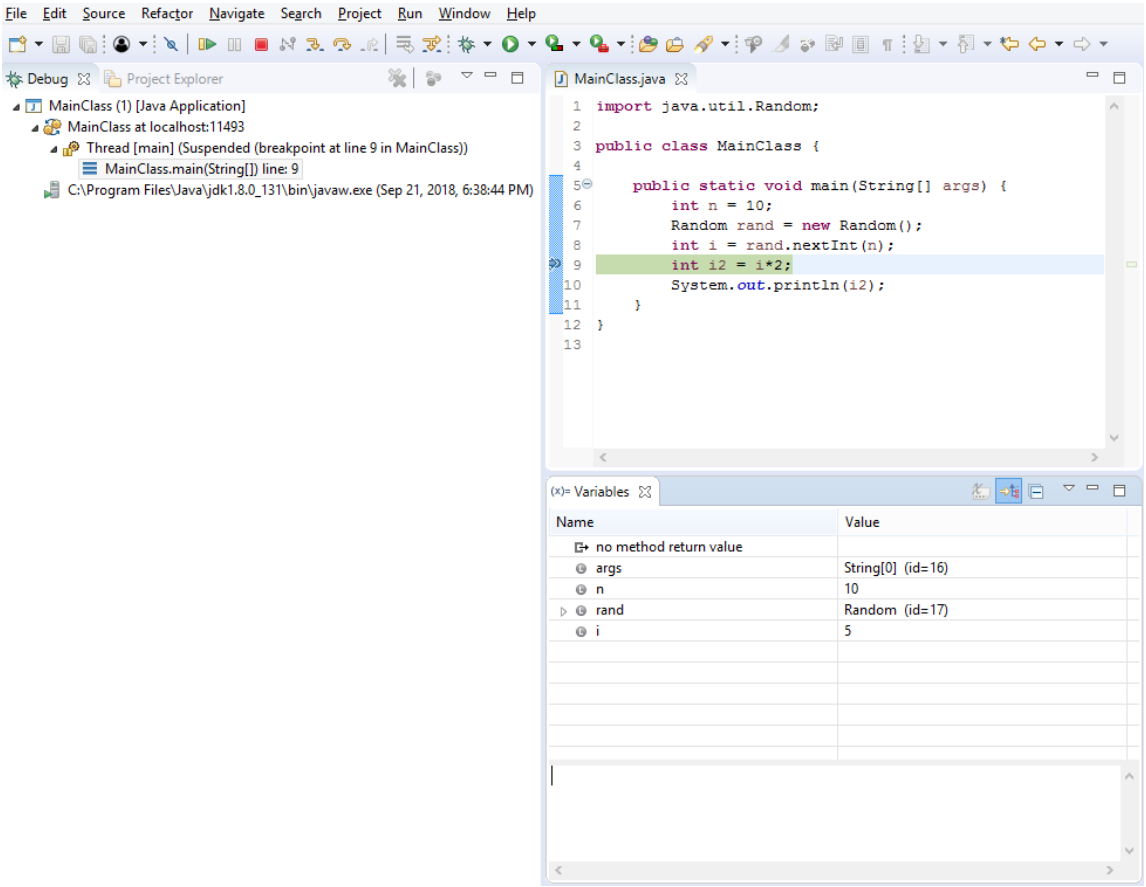
4. Folosirea *debugger*-ului

Pentru a depana un program și a găsi mai ușor greșelile, este recomandată folosirea *debugger*-ului. Depanarea presupune poziționarea de *break point*-uri în dreptul liniilor unde vrem să oprim execuția programului pentru a putea verifica valabilitatea valorilor din variabilele curente.

4.1. Activarea unui *break point* în dreptul liniei unde vrem să oprim execuția programului se face dând dublu click în dreptul acelei linii de cod, iar apoi rularea se face în modul *Debug*, ca în figura de mai jos:



4.2. La rulare, execuția se oprește în dreptul liniei unde am plasat *break point*-ul. Fereastra *Variables* se deschide automat. Aici puteți vizualiza valorile variabilelor locale.



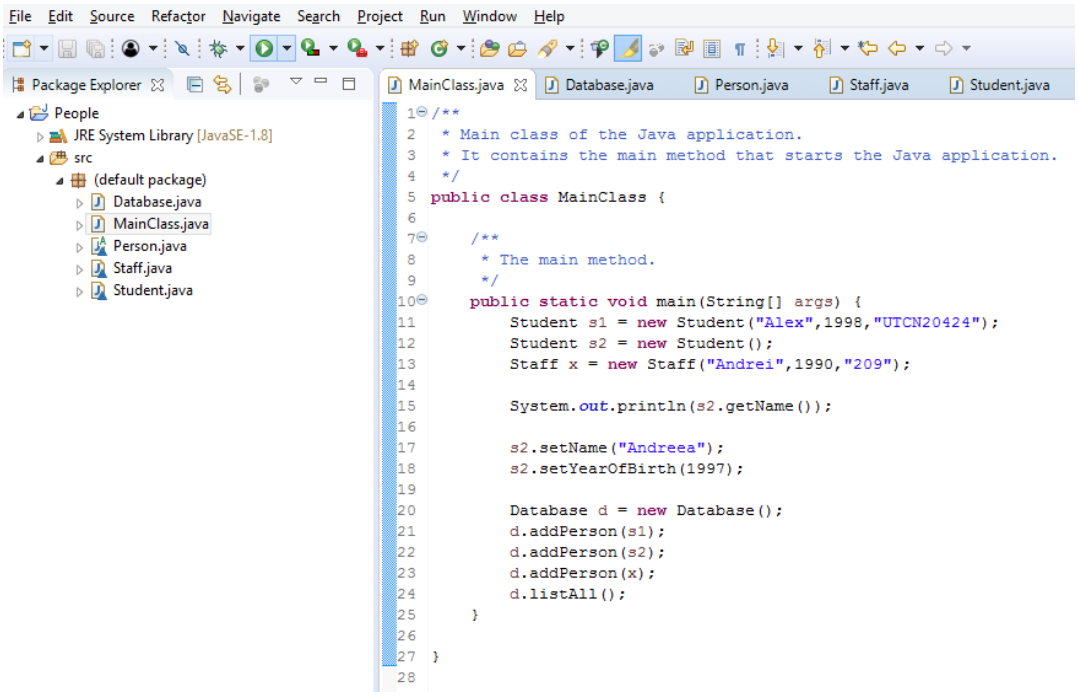
4.3. În continuare aveți posibilitatea
- de a continua rularea programului (*Resume* F8),
- de a merge la următoarea linie de cod care se execută (*Step Into* F5); dacă pe linia curentă este apelul unei metode, atunci opțiunea *Step Into* va intra în interiorul metodei.
- de a merge la următoarea linie de cod din metoda curentă (*Step Over* F6).

5. Crearea unei aplicații Java pe baza unor fișiere sursă existente

Descărcați arhiva *people.zip* pusă la dispoziție. Aceasta conține clasele/fișierele Person.java, Staff.java, Student.java, Database.java, MainClass.java și reprezintă un exemplu simplu de aplicație Java care gestionează personalul unei universități. Dezarhivați și salvați fișierele într-un director.

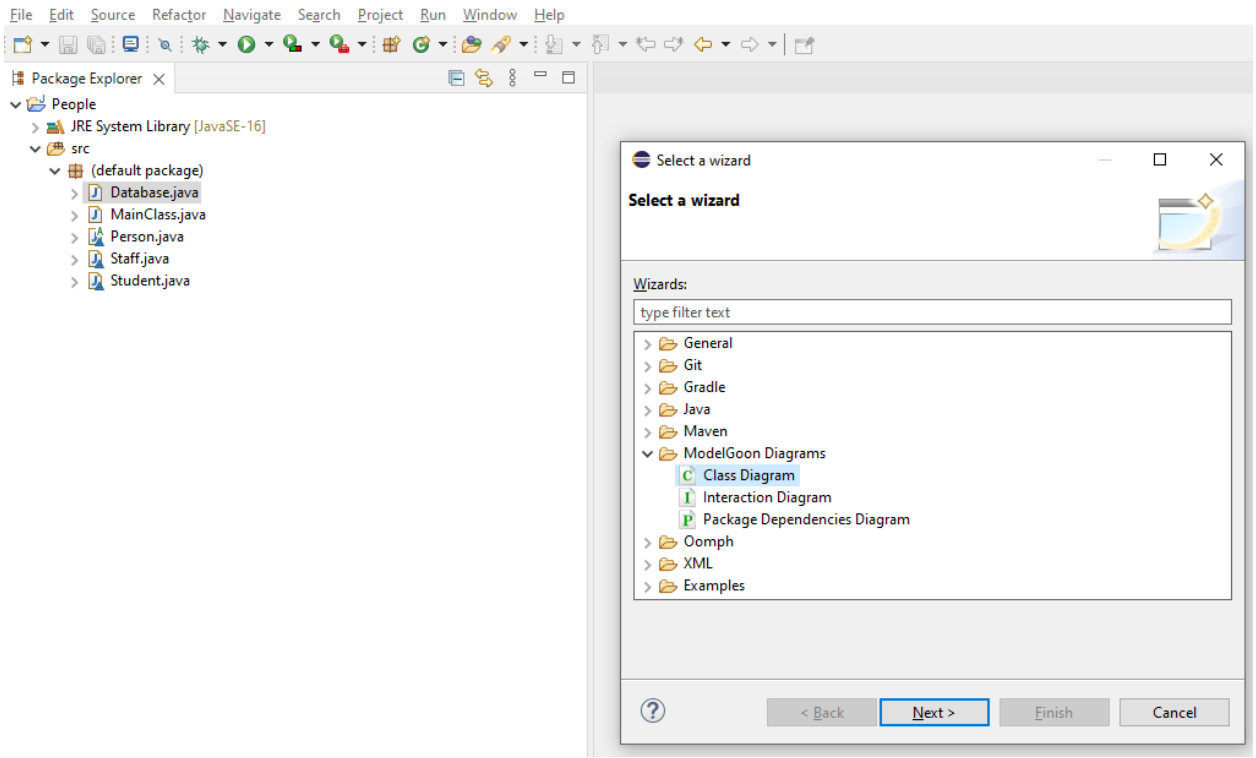
Creați un proiect nou Java numit People. Apoi importați în proiect toate fișierele .java salvate anterior în acel director. Acest lucru este posibil accesând meniul File->Import->General->File System, selectând directorul și toate fișierele .java

Deschideți clasa MainClass și executați aplicația:

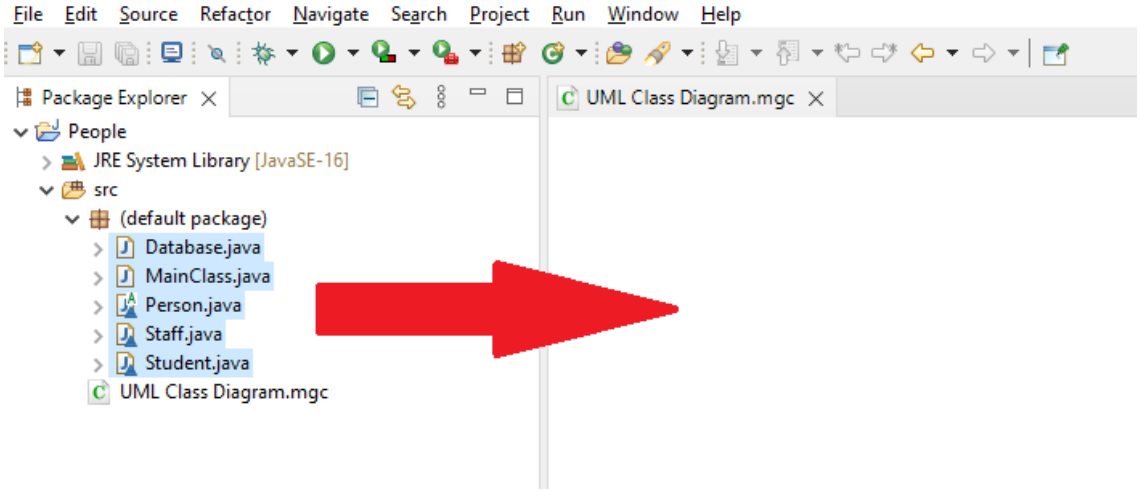


6. Crearea diagramei de clase UML a unei aplicații Java existente

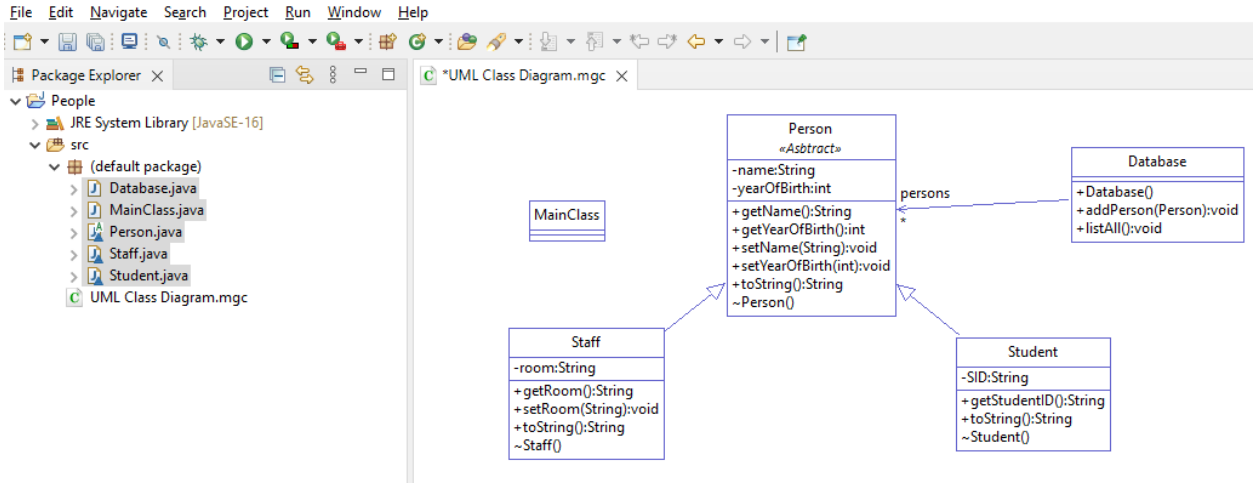
Pentru a genera diagrame de clase UML aferentă unei aplicații Java, având instalat plugin-ul ModelGoon accesați meniul File -> New -> Other… -> ModelGoon Diagrams -> Class Diagram:



Asignați un nume diagramei de clase care urmează a fi creată (de exemplu *UML Class Diagram*). Pentru a se genera diagrama UML de clase selectați toate clasele care doriți să apară pe diagramă și târâți-le (*drag and drop*) în fereastra *UML Class Diagram.mgc* :

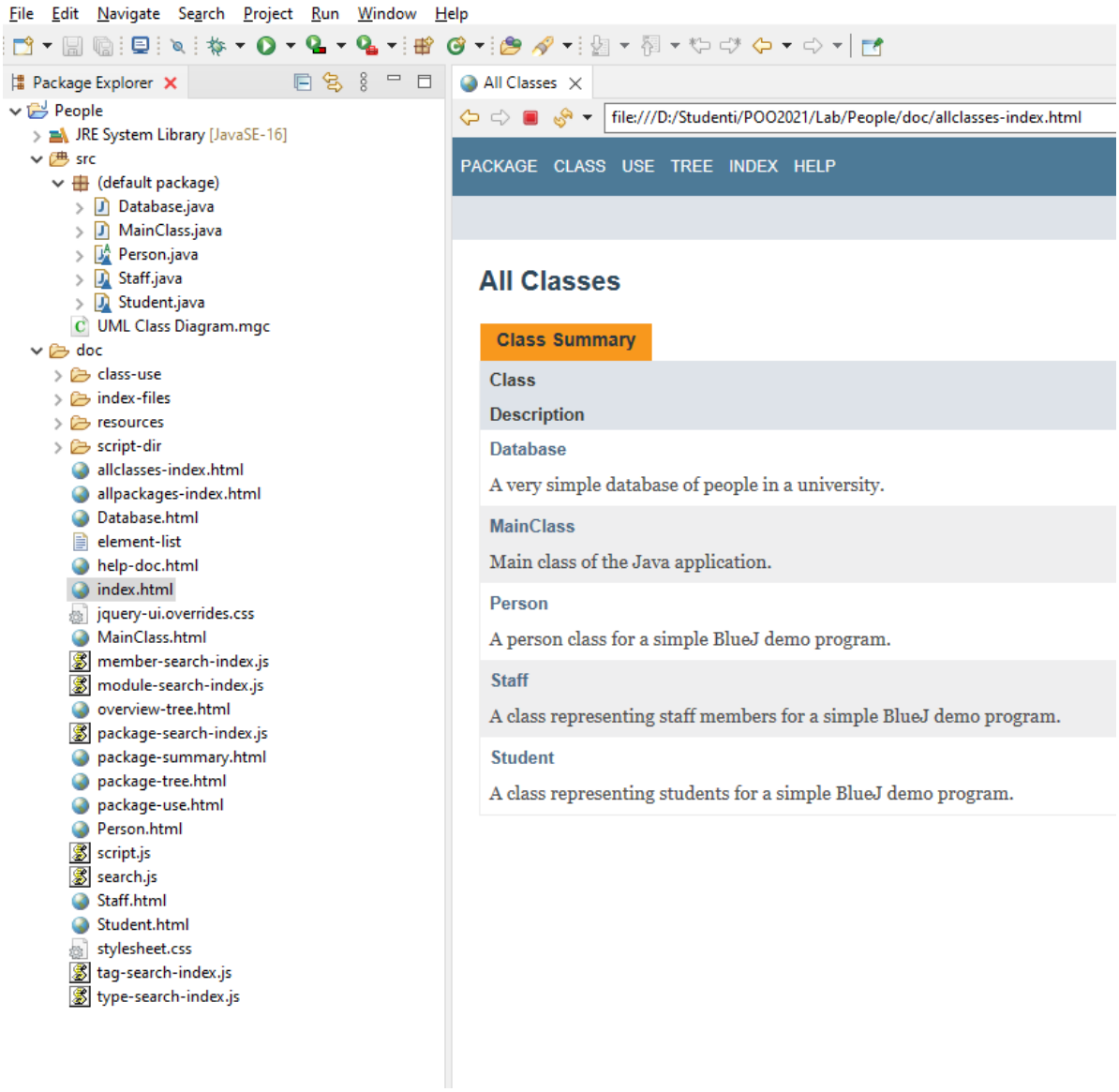


Veți obține diagrama UML de clase a aplicației în cadrul căreia veți putea rearanja entitățile și legăturile dintre ele. Puteți afișa toate atributele și metodele unei clase prin click-dreapta pe aceasta iar apoi accesați *Filter elements* și bifați totul.



7. Crearea documentației unui proiect Java

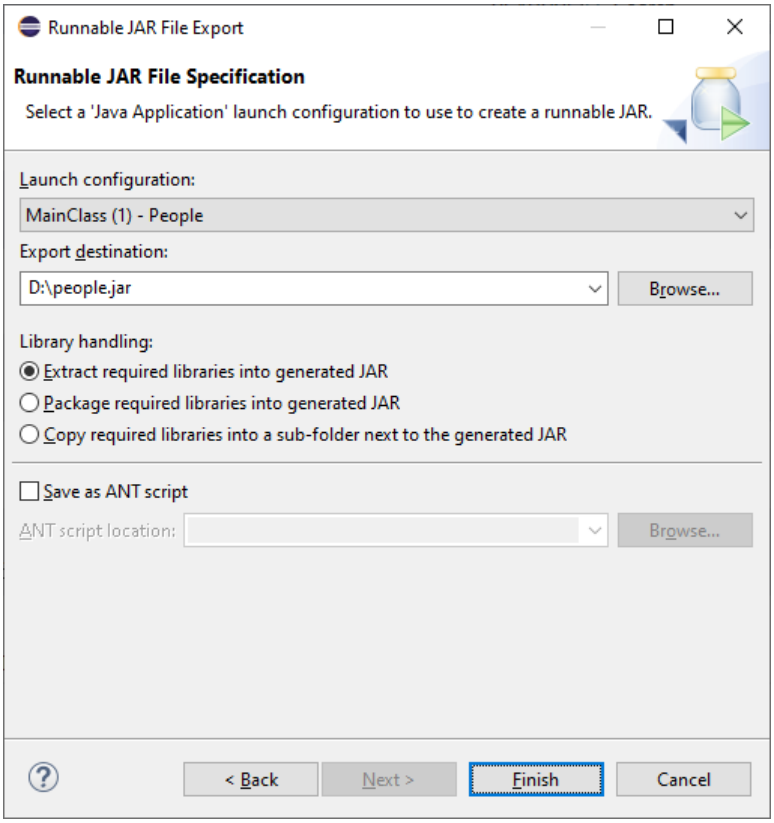
Pentru a genera automat documentația unui proiect Java accesați meniul Project->Generate Javadoc… Setați apoi Javadoc command la fișierul javadoc.exe din pachetul JRE inclus în Eclipse. De exemplu:
D:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jre\bin\javadoc.exe.
Selectați apoi toate fișierele cu cod sursă Java pentru care doriți să se genereze documentație. Acestea trebuie să conțină în prealabil tag-uri Javadoc /** … */ al căror conținut va apărea în documentația generată.
În urma generării documentației rezultă în principal o ierarhie de fișiere HTML. Pentru a vizualiza documentația generată deschideți fișierul principal index.html:



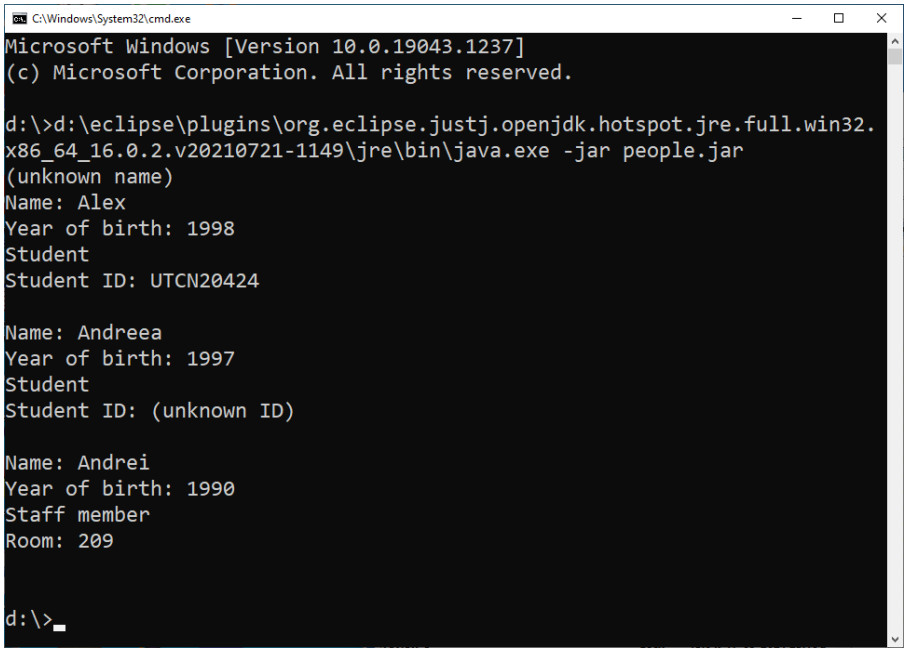
8. Crearea unei aplicații de sine stătătoare - un fișier JAR (Java ARchive)

Un fișier JAR (Java Archive) este o modalitate de împachetare a tuturor resurselor asociate unei aplicații Java (clase, imagini, etc.). Punerea acestora într-un singur fișier JAR permite distribuirea lor ca un singur fișier executabil sau nu, salvând astfel spațiu de stocare și simplificând procesul de descărcare. În cazul în care fișierul JAR creat este executabil atunci aplicația poate fi rulată având la dispoziție doar acest fișier JAR.

Pentru a genera un fișier JAR executabil pentru un proiect Java accesați meniul File->Export->Java-> Runnable JAR file. Selectați clasa care conține metoda main și locul unde va fi salvat fișierul JAR, apoi apăsați butonul Finish:



Dacă doriți să executați aplicația utilizând fișierul JAR executabil creat, deschideți o consolă în directorul unde este localizat fișierul JAR și rulați comanda: **java –jar nume_fisier.jar**
Aplicația va rula în cazul în care directorul din JRE (ex. D:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jre\bin\java.exe) care conține fișierul *java.exe* este deja inclus în variabila de sistem *PATH* a sistemului de operare Windows. În caz contrar (foarte probabil) se poate da calea completă către **java** în comanda prezentată mai sus.



9. Mersul lucrării

9.1. Pregătiți mediul de lucru pentru dezvoltarea de programe în limbajul Java urmărind pașii prezentați în secțiunea 2 (dacă nu este deja pregătit!). Creați-vă propriul director în care vă veți salva proiectele și care va fi accesat de fiecare dată când porniți mediul de dezvoltare Eclipse (vezi secțiunea 2).

9.2. Creați o aplicație Java simplă care să afișeze mesajul "Hello world" urmând pașii prezentați în secțiunea 3.

9.3. Modificați metoda *main* a aplicației creată anterior astfel încât să extrageți aleatoriu un număr întreg între 0 și 9 și tipăriți dublul acestui număr (vezi figura din secțiunea 4). Depanați programul realizat urmând pașii prezentați în secțiunea 4.

9.4. Creați aplicația People prin importarea de fișiere Java existente (vezi secțiunea 5). Realizați o vedere de ansamblu asupra codului sursă al aplicației și asupra rezultatelor furnizate în urma execuției ei.

9.5. Generați și apoi analizați diagrama UML de clase aferentă aplicației People (vezi secțiunea 6).

9.6. Generați și apoi analizați documentația aplicației People utilizând Javadoc (vezi secțiunea 7).

9.7. Creați o aplicație de sine stătătoare (un fișier JAR executabil) pe baza proiectului People. Executați într-o consolă fișierul JAR executabil creat (vezi secțiunea 8).