

**Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»**

**ІНСТИТУТ ЕЛЕКТРОЕНЕРГЕТИКИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інформаційних технологій та комп'ютерної інженерії**



Дисципліни “Програмування в середовищі Java”

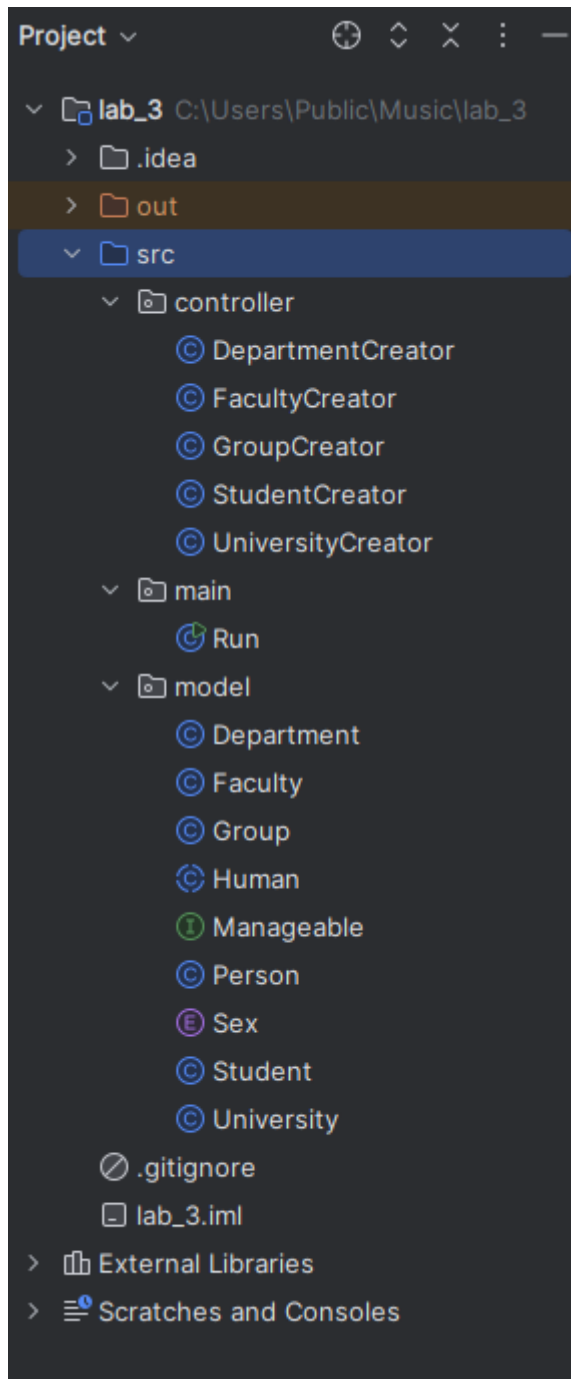
Виконав: студент групи 123-21-2
Яковенко Д.Е.

Перевірив:
Мінєєв Олександр Сергійович

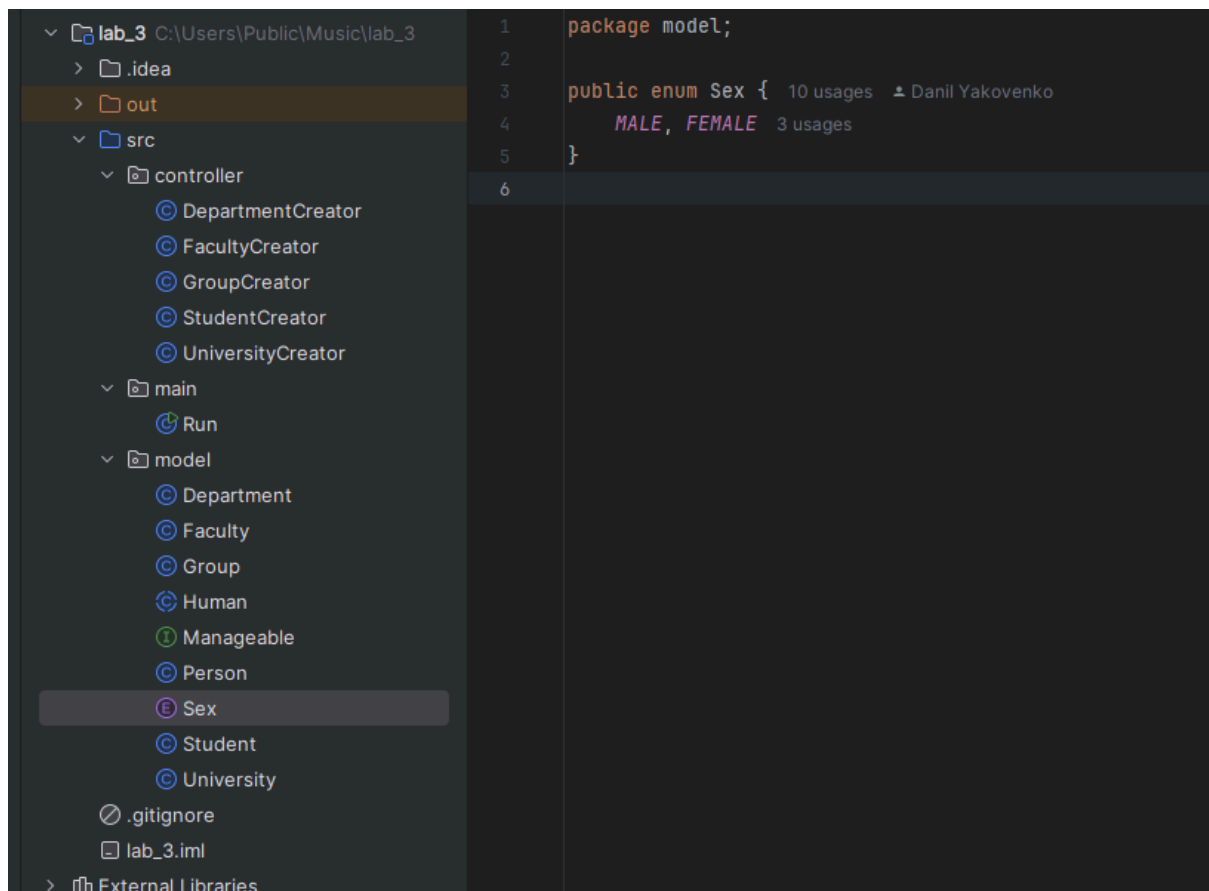
**Дніпро
2023**

Лабораторна робота номер 3

Спочатку створюємо 3 Package з такими назвами (Model, Controller, Main)
та додаємо Java class до кожного з Package класи

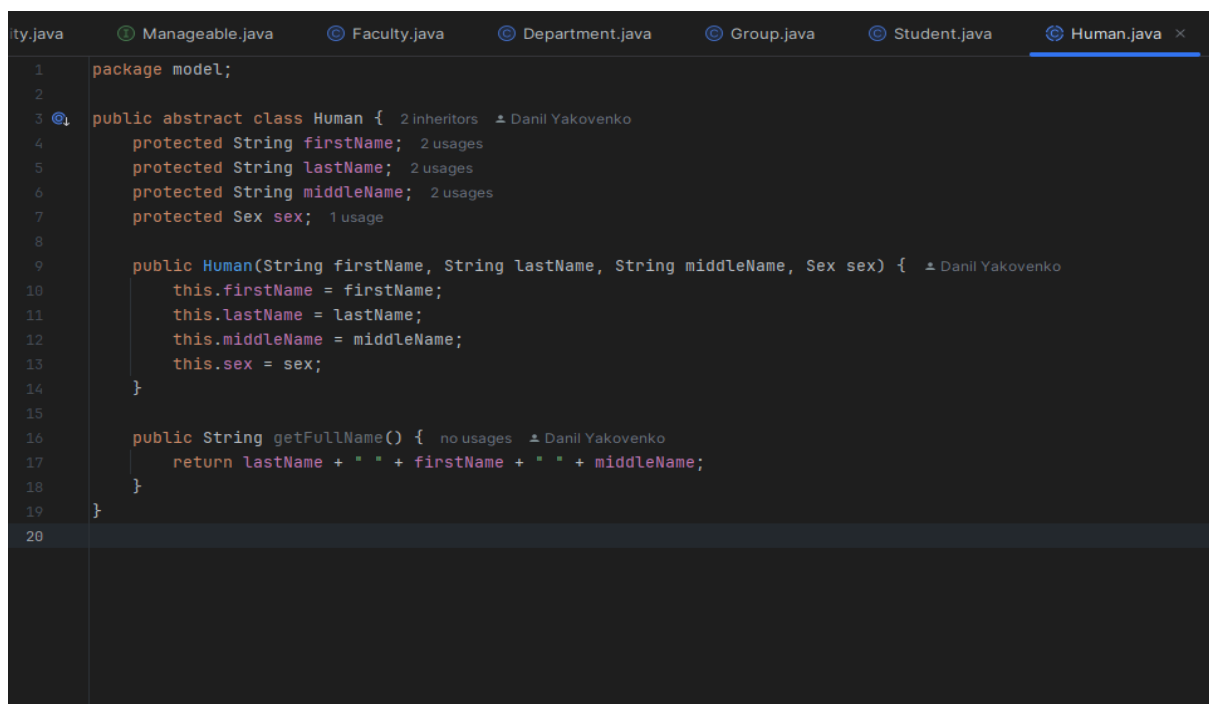


Прописуємо для статі (Sex) такий код



```
1 package model;
2
3 public enum Sex { 10 usages  ⬆ Danil Yakovenko
4     MALE, FEMALE 3 usages
5 }
6
```

Пропсужмо класу людина (Human)



```
1 package model;
2
3 public abstract class Human { 2 inheritors  ⬆ Danil Yakovenko
4     protected String firstName; 2 usages
5     protected String lastName; 2 usages
6     protected String middleName; 2 usages
7     protected Sex sex; 1 usage
8
9     public Human(String firstName, String lastName, String middleName, Sex sex) {  ⬆ Danil Yakovenko
10         this.firstName = firstName;
11         this.lastName = lastName;
12         this.middleName = middleName;
13         this.sex = sex;
14     }
15
16     public String getFullName() { no usages  ⬆ Danil Yakovenko
17         return lastName + " " + firstName + " " + middleName;
18     }
19 }
20
```

Для класу студент (Student) успадковує клас людини (Human)

```

1 package model;
2
3 public class Student extends Human { 10 usages  ⚡ Danil Yakovenko
4     private final String studentID; 2 usages
5
6     public Student(String firstName, String lastName, String middleName, Sex sex, String studentID) { 1 usage  ⚡ Danil Yakovenko
7         super(firstName, lastName, middleName, sex);
8         this.studentID = studentID;
9     }
10
11     public String getStudentID() { no usages  ⚡ Danil Yakovenko
12         return studentID;
13     }
14 }
15

```

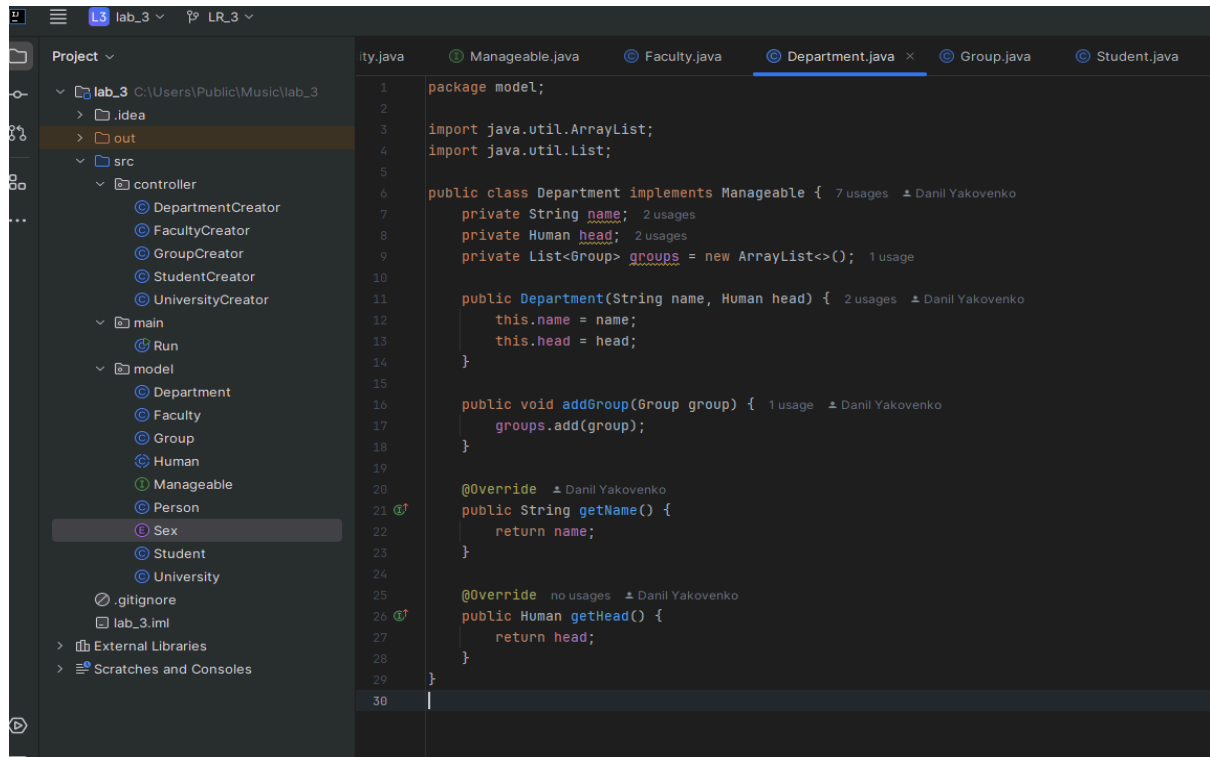
В клас Group висуємо такі значення

```

1 package model;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Group implements Manageable { 7 usages  ⚡ Danil Yakovenko
7     private String name; 2 usages
8     private Human head; 2 usages
9     private List<Student> students = new ArrayList<>(); 1 usage
10
11     public Group(String name, Human head) { 1 usage  ⚡ Danil Yakovenko
12         this.name = name;
13         this.head = head;
14     }
15
16     public void addStudent(Student student) { 1 usage  ⚡ Danil Yakovenko
17         students.add(student);
18     }
19
20     @Override  ⚡ Danil Yakovenko
21     public String getName() {
22         return name;
23     }
24
25     @Override no usages  ⚡ Danil Yakovenko
26     public Human getHead() {
27         return head;
28     }
29 }
30

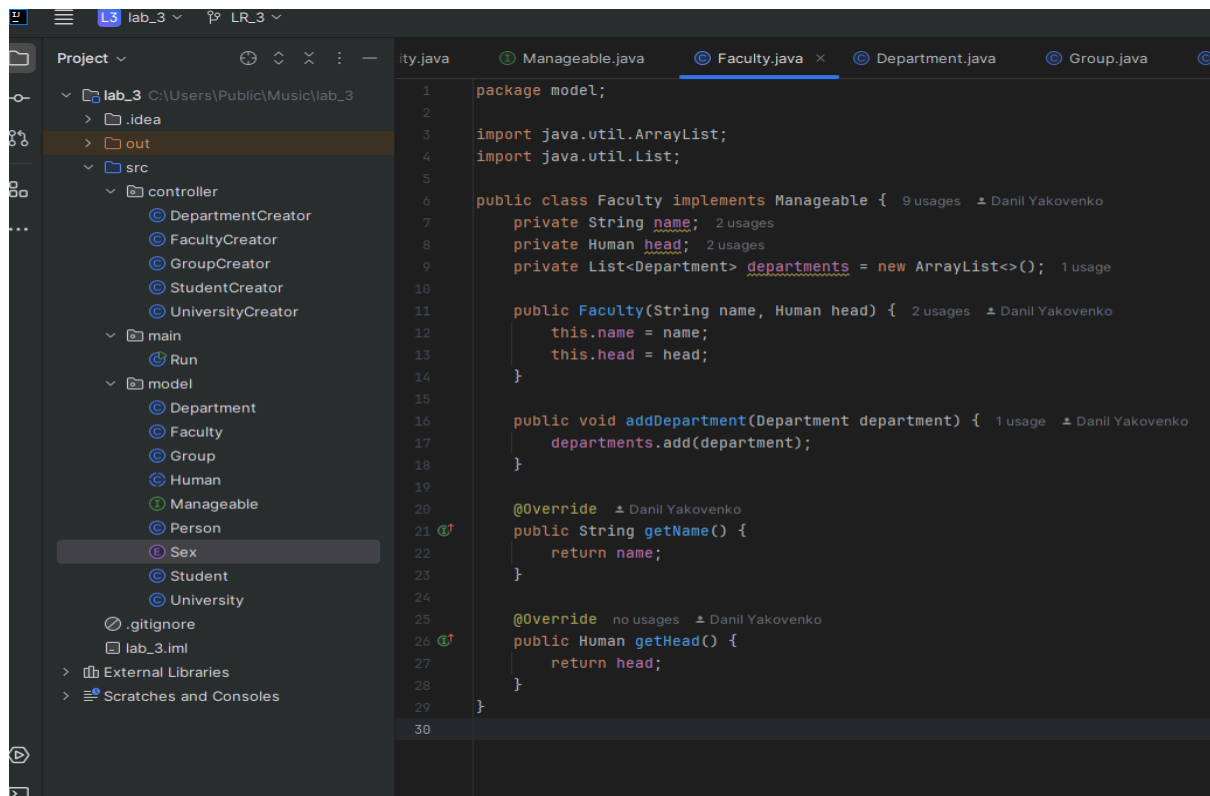
```

В Department пишемо такий код, де поля зберігають назви.

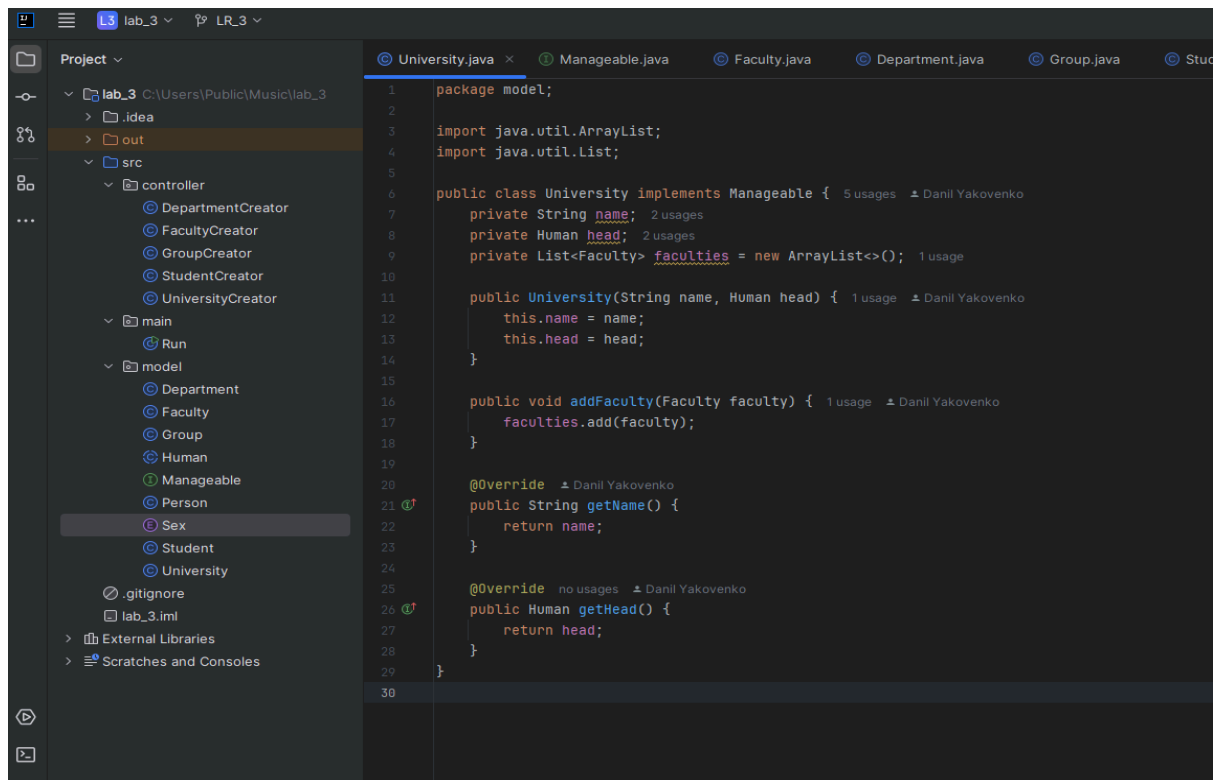


```
1 package model;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Department implements Manageable { 7 usages  ± Danil Yakovenko
7     private String name; 2 usages
8     private Human head; 2 usages
9     private List<Group> groups = new ArrayList<>(); 1 usage
10
11     public Department(String name, Human head) { 2 usages  ± Danil Yakovenko
12         this.name = name;
13         this.head = head;
14     }
15
16     public void addGroup(Group group) { 1 usage  ± Danil Yakovenko
17         groups.add(group);
18     }
19
20     @Override  ± Danil Yakovenko
21     public String getName() {
22         return name;
23     }
24
25     @Override  no usages  ± Danil Yakovenko
26     public Human getHead() {
27         return head;
28     }
29 }
30
```

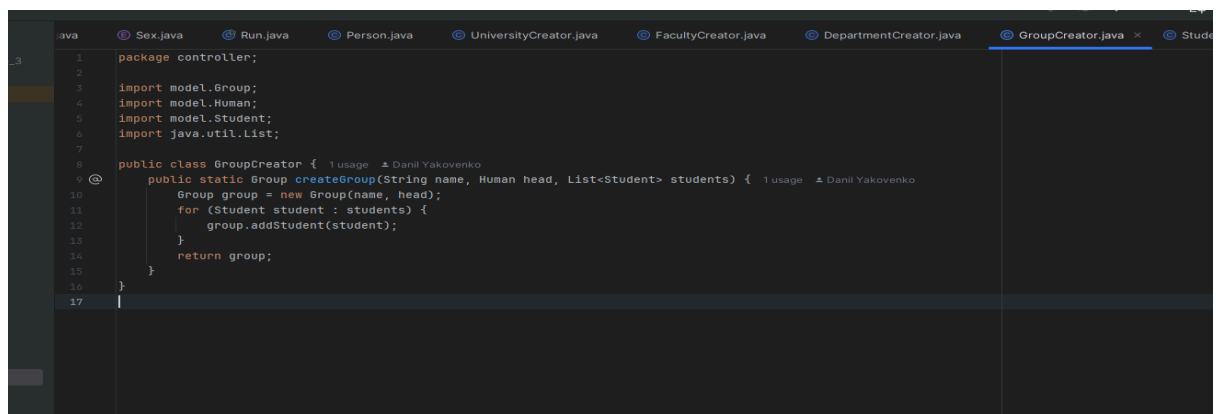
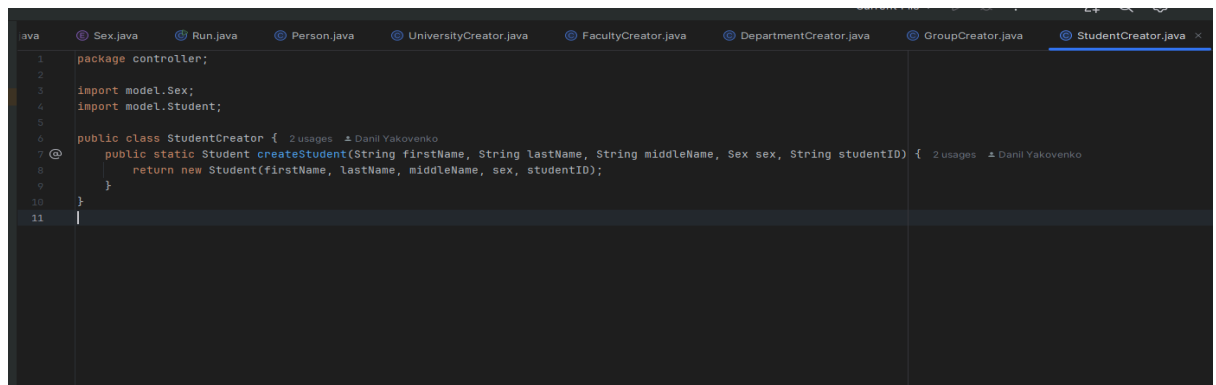
Пишемо клас факультета та університету



```
1 package model;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Faculty implements Manageable { 9 usages  ± Danil Yakovenko
7     private String name; 2 usages
8     private Human head; 2 usages
9     private List<Department> departments = new ArrayList<>(); 1 usage
10
11     public Faculty(String name, Human head) { 2 usages  ± Danil Yakovenko
12         this.name = name;
13         this.head = head;
14     }
15
16     public void addDepartment(Department department) { 1 usage  ± Danil Yakovenko
17         departments.add(department);
18     }
19
20     @Override  ± Danil Yakovenko
21     public String getName() {
22         return name;
23     }
24
25     @Override  no usages  ± Danil Yakovenko
26     public Human getHead() {
27         return head;
28     }
29 }
30
```



Далі прописуємо код для Package controller починаючи з StudentCreator.java, GroupCreator.java, FacultyCreator.java, UniversityCreator.java



```
ava Sex.java Run.java Person.java UniversityCreator.java FacultyCreator.java DepartmentCreator.java x Grou

1 package controller;
2
3 import model.Department;
4 import model.Human;
5
6 public class DepartmentCreator { no usages 1 Danil Yakovenko
7 @ public static Department createDepartment(String name, Human head) { no usages 1 Danil Yakovenko
8     return new Department(name, head);
9 }
10 }
11
```

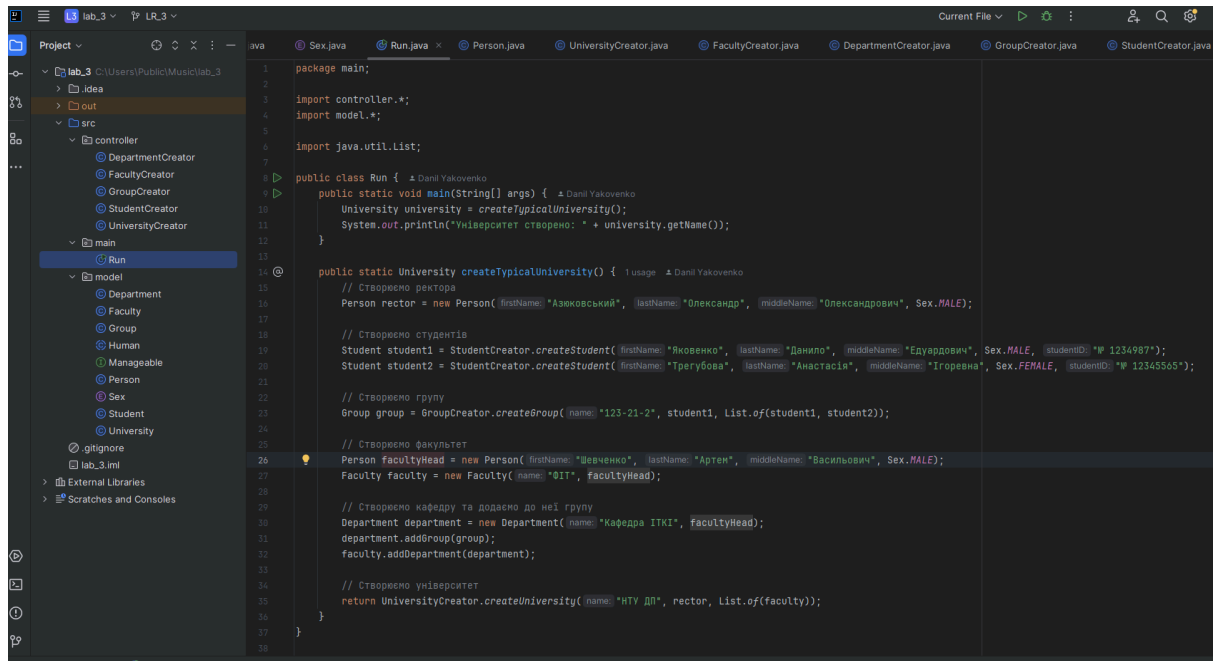
```
lab_3 java Sex.java Run.java Person.java UniversityCreator.java FacultyCreator.java x

1 package controller;
2
3 import model.Faculty;
4 import model.Human;
5
6 public class FacultyCreator { no usages 1 Danil Yakovenko
7 @ public static Faculty createFaculty(String name, Human head) { no usages 1 Danil Yakovenko
8     return new Faculty(name, head);
9 }
10 }
11
```

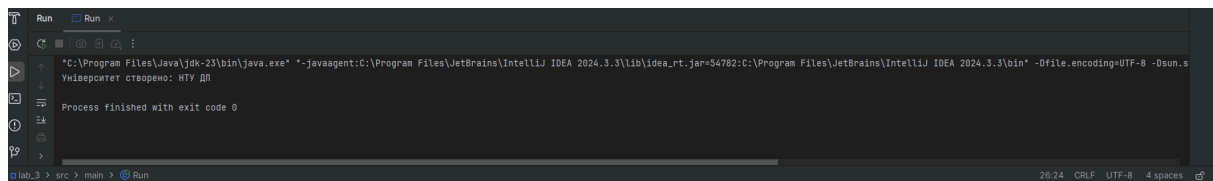
```
Current File lab_3 java Sex.java Run.java Person.java UniversityCreator.java x FacultyCreator.java DepartmentCreator.java GroupCreator.java StudentCreator.java

1 package controller;
2
3 import model.*;
4
5 import java.util.List;
6
7 public class UniversityCreator { 1 usage 1 Danil Yakovenko
8 @ public static University createUniversity(String name, Human head, List<Faculty> faculties) { 1 usage 1 Danil Yakovenko
9     University university = new University(name, head);
10     for (Faculty faculty : faculties) {
11         university.addFaculty(faculty);
12     }
13     return university;
14 }
15 }
16
```

Створюємо Package main для класу run де буде працювати наша програма



```
1 package main;
2
3 import controller.*;
4 import model.*;
5
6 import java.util.List;
7
8 public class Run {
9     public static void main(String[] args) {
10         University university = createTypicalUniversity();
11         System.out.println("Університет створено: " + university.getName());
12     }
13
14     public static University createTypicalUniversity() {
15         // Створимо ректора
16         Person rector = new Person(firstName: "Азиковський", lastName: "Олександр", middleName: "Олександрович", Sex.MALE);
17
18         // Створимо студентів
19         Student student1 = StudentCreator.createStudent(firstName: "Яковенко", lastName: "Данило", middleName: "Едуардович", Sex.MALE, studentID: "ID 1234987");
20         Student student2 = StudentCreator.createStudent(firstName: "Терябова", lastName: "Анастасія", middleName: "Ігорівна", Sex.FEMALE, studentID: "ID 12345565");
21
22         // Створимо групу
23         Group group = GroupCreator.createGroup(name: "123-21-2", student1, List.of(student1, student2));
24
25         // Створимо факультет
26         Person facultyHead = new Person(firstName: "Шевченко", lastName: "Артем", middleName: "Васильович", Sex.MALE);
27         Faculty faculty = new Faculty(name: "ФІТ", facultyHead);
28
29         // Створимо кафедру та додаємо до неї групу
30         Department department = new Department(name: "Кафедра ІТКІ", facultyHead);
31         department.addGroup(group);
32         faculty.addDepartment(department);
33
34         // Створимо університет
35         return UniversityCreator.createUniversity(name: "НТУ ДП", rector, List.of(faculty));
36     }
37 }
38
```



```
Run
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.3\lib\idea_rt.jar=54782:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.3\bin" -Dfile.encoding=UTF-8 -Dsun.s
Університет створено: НТУ ДП
Process finished with exit code 0
```