

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

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



Лабораторна робота №3

з дисципліни:  
«Базова Джава»

*Виконав:*  
студент групи 123-21-1  
Скоропад Є.В.  
*Перевірів:*  
Доцент кафедри САіУ  
Мінєєв О.С.

**Дніпро**  
**2025**

## Завдання

Створити програму що буде створювати та обробляти комплексний об'єкт під назвою університет(university). Програма повинна складатися з трьох частин: модель вид та контролер згідно з парадигмою mvc (Model View Controller). Кожній з цих груп повинна відповідати package з відповідною назвою. В моделі повинні знаходитись усі класи що відповідають за структурні підрозділи університету. Серед них: університет, факультет, кафедра, група, студент, людина (Human). Усі вони повинні містити назву типу string та голову типу Human. Студент також повинен бути породжений від Human. Human повинен мати поля ім'я, прізвище, побатькові та стать. Усі поля повинні бути строковими окрім поля стать. Стать повинна використовувати спеціальний enum типу Sex(стать).

В цій лабораторній роботі група View Нам не потрібна.

Що стосується групи контроллер (controller) то вона повинна містити менеджери що дозволяють нам створити відповідні підрозділи наприклад StudentCreator, FacultyCreator, GroupCreator та інші, кожен з яких повинен використовувати можливості нижчого за рівнем створювача. Програма повинна також містити клас Run, в якому буде знаходитись точка входу та методи, що повинні дати можливість створити університет. Процес створення університету повинен бути зроблений в методі createTypycalUniversit.

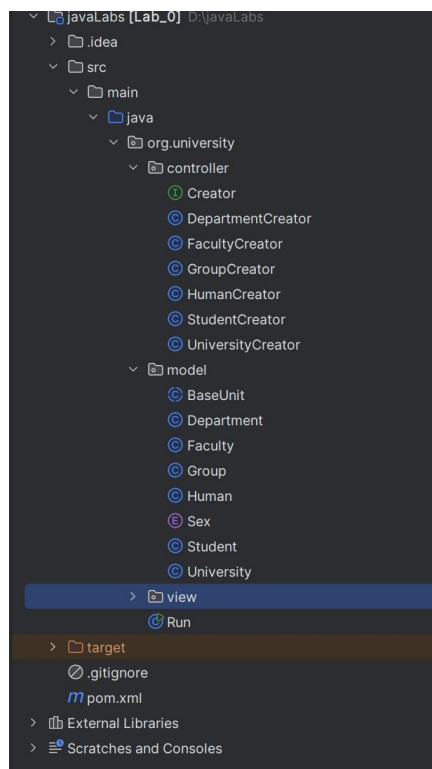


Рисунок 1 – Структура проєкту.

## Код модуля controller

### Interface Creator

```
package org.university.controller;

public interface Creator<T> {
    T create();
}
```

### Клас DepartmentCreator

```
package org.university.controller;

import org.university.model.Department;
import org.university.model.Human;

public class DepartmentCreator implements Creator<Department> {
    private String name;
    private Human head;
    private GroupCreator[] groupCreators;

    public DepartmentCreator(String name, Human head, GroupCreator[]
groupCreators) {
        this.name = name;
        this.head = head;
        this.groupCreators = groupCreators;
    }

    @Override
    public Department create() {
        Department department = new Department(name, head);
        for (GroupCreator creator : groupCreators) {
            department.addGroup(creator.create());
        }
        return department;
    }
}
```

### Клас FacultyCreator

```
package org.university.controller;

import org.university.model.Faculty;
import org.university.model.Human;

public class FacultyCreator implements Creator<Faculty> {
    private String name;
    private Human head;
    private DepartmentCreator[] departmentCreators;

    public FacultyCreator(String name, Human head, DepartmentCreator[]
departmentCreators) {
        this.name = name;
        this.head = head;
        this.departmentCreators = departmentCreators;
    }

    @Override
    public Faculty create() {
```

```

        Faculty faculty = new Faculty(name, head);
        for (DepartmentCreator creator : departmentCreators) {
            faculty.addDepartment(creator.create());
        }
        return faculty;
    }
}

```

## Клас GroupCreator

```

package org.university.controller;

import org.university.model.Group;
import org.university.model.Human;

public class GroupCreator implements Creator<Group> {
    private String name;
    private Human head;
    private StudentCreator[] students;

    public GroupCreator(String name, Human head, StudentCreator[] students) {
        this.name = name;
        this.head = head;
        this.students = students;
    }

    @Override
    public Group create() {
        Group group = new Group(name, head);
        for (StudentCreator creator : students) {
            group.addStudent(creator.create());
        }
        return group;
    }
}

```

## Клас HumanCreator

```

package org.university.controller;

import org.university.model.Human;
import org.university.model.Sex;

public class HumanCreator implements Creator<Human> {
    private String firstName;
    private String lastName;
    private String patronymic;
    private Sex sex;

    public HumanCreator(String firstName, String lastName, String patronymic,
Sex sex) {
        this.firstName = firstName;
        this.lastName = lastName;
        this.patronymic = patronymic;
        this.sex = sex;
    }

    @Override
    public Human create() {
        return new Human(firstName, lastName, patronymic, sex);
    }
}

```

## Клас StudentCreator

```
package org.university.controller;

import org.university.model.Sex;
import org.university.model.Student;

public class StudentCreator implements Creator<Student> {
    private String firstName;
    private String lastName;
    private String patronymic;
    private Sex sex;
    private String studentId;

    public StudentCreator(String firstName, String lastName, String
patronymic, Sex sex, String studentId) {
        this.firstName = firstName;
        this.lastName = lastName;
        this.patronymic = patronymic;
        this.sex = sex;
        this.studentId = studentId;
    }

    @Override
    public Student create() {
        return new Student(firstName, lastName, patronymic, sex, studentId);
    }
}
```

## Клас UniversityCreator

```
package org.university.controller;

import org.university.model.University;
import org.university.model.Human;

public class UniversityCreator implements Creator<University> {
    private String name;
    private Human head;
    private FacultyCreator[] facultyCreators;

    public UniversityCreator(String name, Human head, FacultyCreator[]
facultyCreators) {
        this.name = name;
        this.head = head;
        this.facultyCreators = facultyCreators;
    }

    @Override
    public University create() {
        University university = new University(name, head);
        for (FacultyCreator creator : facultyCreators) {
            university.addFaculty(creator.create());
        }
        return university;
    }
}
```

## Код модуля model

### Абстрактный класс BaseUnit

```
package org.university.model;

public abstract class BaseUnit {
    protected String name;
    protected Human head;

    public BaseUnit(String name, Human head) {
        this.name = name;
        this.head = head;
    }

    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public Human getHead() { return head; }
    public void setHead(Human head) { this.head = head; }
}
```

### Клас Department

```
package org.university.model;

import java.util.ArrayList;
import java.util.List;

public class Department extends BaseUnit {
    private List<Group> groups;

    public Department(String name, Human head) {
        super(name, head);
        this.groups = new ArrayList<>();
    }

    public List<Group> getGroups() { return groups; }
    public void addGroup(Group group) { groups.add(group); }
}
```

### Клас Faculty

```
package org.university.model;

import java.util.ArrayList;
import java.util.List;

public class Faculty extends BaseUnit {
    private List<Department> departments;

    public Faculty(String name, Human head) {
        super(name, head);
        this.departments = new ArrayList<>();
    }

    public List<Department> getDepartments() { return departments; }
    public void addDepartment(Department department) {
        departments.add(department); }
}
```

## Клас Group

```
package org.university.model;

import java.util.ArrayList;
import java.util.List;

public class Group extends BaseUnit {
    private List<Student> students;

    public Group(String name, Human head) {
        super(name, head);
        this.students = new ArrayList<>();
    }

    public List<Student> getStudents() { return students; }
    public void addStudent(Student student) { students.add(student); }
}
```

## Клас Human

```
package org.university.model;

public class Human {
    private String firstName;
    private String lastName;
    private String patronymic;
    private Sex sex;

    public Human(String firstName, String lastName, String patronymic, Sex
sex) {
        this.firstName = firstName;
        this.lastName = lastName;
        this.patronymic = patronymic;
        this.sex = sex;
    }

    public String getFirstName() { return firstName; }
    public void setFirstName(String firstName) { this.firstName = firstName; }

    public String getLastName() { return lastName; }
    public void setLastName(String lastName) { this.lastName = lastName; }
    public String getPatronymic() { return patronymic; }
    public void setPatronymic(String patronymic) { this.patronymic =
patronymic; }
    public Sex getSex() { return sex; }
    public void setSex(Sex sex) { this.sex = sex; }
}
```

## Клас Enum Sex

```
package org.university.model;

public enum Sex {
    MALE,
    FEMALE,
}
```

## Клас Student

```
package org.university.model;

public class Student extends Human {
    private String studentId;

    public Student(String firstName, String lastName, String patronymic, Sex
sex, String studentId) {
        super(firstName, lastName, patronymic, sex);
        this.studentId = studentId;
    }

    public String getStudentId() { return studentId; }
    public void setStudentId(String studentId) { this.studentId = studentId;
}
}
```

## Клас University

```
package org.university.model;

import java.util.ArrayList;
import java.util.List;

public class University extends BaseUnit {
    private List<Faculty> faculties;

    public University(String name, Human head) {
        super(name, head);
        this.faculties = new ArrayList<>();
    }

    public List<Faculty> getFaculties() { return faculties; }
    public void addFaculty(Faculty faculty) { faculties.add(faculty); }
}
```

## Головний клас Run

```
package org.university;

import org.university.controller.*;
import org.university.model.*;
import org.university.view.UniversityView;

public class Run {
    public static void main(String[] args) {
        University university = createTypicalUniversity();
        UniversityView view = new UniversityView();
        view.displayUniversity(university);
    }
}
```



```

    public static University createTypicalUniversity() {
        // Create university rector
        Human rector = new HumanCreator("John", "Smith", "Grigorivich",
Sex.MALE).create();

        // Create faculty dean
        Human dean = new HumanCreator("Michael", "Johnson", "Abob",
Sex.MALE).create();

        // Create department head
        Human departmentHead = new HumanCreator("Sarah", "Bobrova",
"Sergeyevich", Sex.FEMALE).create();

        // Create group curator
        Human groupCurator = new HumanCreator("David", "Babrash",
"Olegovich", Sex.MALE).create();

        // Create students
        StudentCreator[] studentCreators = {
            new StudentCreator("Alice", "Brown", "", Sex.FEMALE, "25"),
            new StudentCreator("Bob", "Taylor", "", Sex.MALE, "3"),
            new StudentCreator("Charlie", "Anderson", "", Sex.MALE, "4")
        };

        // Create group
        GroupCreator groupCreator = new GroupCreator("123-21-1",
groupCurator, studentCreators);

        // Create department
        DepartmentCreator departmentCreator = new DepartmentCreator(
            "Computer Science",
            departmentHead,
            new GroupCreator[]{groupCreator}
        );

        // Create faculty
        FacultyCreator facultyCreator = new FacultyCreator(
            "Faculty of Science",
            dean,
            new DepartmentCreator[]{departmentCreator}
        );

        // Create university
        UniversityCreator universityCreator = new UniversityCreator(
            "National University",
            rector,
            new FacultyCreator[]{facultyCreator}
        );

        return universityCreator.create();
    }
}

```

## Результат виконання коду

```
University: National University
Rector: John Smith Grigorivich
  Faculty: Faculty of Science
    Dean: Michael Johnson Abob
      Department: Computer Science
        Head: Sarah Bobrova Sergeyevich
          Group: 123-21-1
            Curator: David Babrash Olegovich
              Student: Alice Brown (ID: 25)
                Student: Bob Taylor (ID: 3)
                  Student: Charlie Anderson (ID: 4)

Process finished with exit code 0
```