

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»
Кафедра системного аналізу та управління



Звіт з лабораторної роботи № 3
З дисципліни «Поглиблене програмування в середовищі Java»

Виконав:
ст. гр. 122-21-2
Помазан Микита
Максимович

Дніпро
2025

Лабораторна робота: ООП

Мета роботи: Розробити програму для створення та обробки комплексного об'єкта "університет" відповідно до архітектурного патерну MVC. Забезпечити структуру проєкту через пакети model, controller та реалізувати відповідні класи для представлення університету та його підрозділів.

Хід роботи:

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, по батькові 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 контролер (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 createTypycalUniversity.

```
Університет: Дніпровська політехніка, очолює: Коваленко Оксана Миколаївна, FEMALE
Факультет: IT, очолює: Шевченко Олена Іванівна, FEMALE
Кафедра: Комп'ютерні науки, очолює: Джонсон Михаїл Якович, MALE
Група: Група А, очолює: Джонсон Михаїл Якович, MALE
Студент: Помазан Микита Максимович, MALE, номер студента: 1
Студент: Петренко Марія Анатоліївна, FEMALE, номер студента: 2
```

Рис. 1 - Результат роботи програми

Код програми:

https://github.com/mykytapomazan/java-basic/tree/LR_3

або:

Моделі:

```
package model;

import java.util.List;

public class Department {

    private String name;

    private Human head;

    private List<Group> groups;

    public Department(String name, Human head, List<Group> groups) {

        this.name = name;

        this.head = head;

        this.groups = groups;

    }

    public String getName() {

        return name;

    }

    public Human getHead() {

        return head;

    }

    public List<Group> getGroups() {

        return groups;

    }

    @Override

    public String toString() {

        return name + " headed by " + head.toString();

    }

}
```

```
package model;

import java.util.List;

public class Faculty {
    private String name;
    private Human head;
    private List<Department> departments;

    public Faculty(String name, Human head, List<Department> departments) {
        this.name = name;
        this.head = head;
        this.departments = departments;
    }

    public String getName() {
        return name;
    }

    public Human getHead() {
        return head;
    }

    public List<Department> getDepartments() {
        return departments;
    }

    @Override
    public String toString() {
        return name + " headed by " + head.toString();
    }
}
```

```
package model;

import java.util.List;

public class Group {
    private String name;
    private Human head;
    private List<Student> students;

    public Group(String name, Human head, List<Student> students) {
        this.name = name;
        this.head = head;
        this.students = students;
    }

    public String getName() {
        return name;
    }

    public Human getHead() {
        return head;
    }
}
```

```

public List<Student> getStudents() {
    return students;
}

@Override
public String toString() {
    return name + " headed by " + head.toString();
}
}

```

```

package model;

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

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

    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public String getMiddleName() {
        return middleName;
    }

    public Sex getSex() {
        return sex;
    }

    @Override
    public String toString() {
        return lastName + " " + firstName + " " + middleName + ", " + sex;
    }
}

```

```

package model;

public enum Sex {
    MALE, FEMALE
}

```

```

package model;

public class Student extends Human {
    private String studentId;

```

```

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

    public String getStudentId() {
        return studentId;
    }

    @Override
    public String toString() {
        return super.toString() + ", номер студента: " + studentId;
    }
}

```

```

package model;

public class Student extends Human {
    private String studentId;

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

    public String getStudentId() {
        return studentId;
    }

    @Override
    public String toString() {
        return super.toString() + ", номер студента: " + studentId;
    }
}

```

Контролери:

```

package controller;

import model.Faculty;
import model.Human;
import model.Department;

import java.util.List;

public class FacultyCreator {
    public static Faculty createFaculty(String name, Human head, List<Department> departments) {
        return new Faculty(name, head, departments);
    }
}

```

```

package controller;

import model.Group;

```

```
import model.Human;
import model.Student;

import java.util.List;

public class GroupCreator {
    public static Group createGroup(String name, Human head, List<Student> students) {
        return new Group(name, head, students);
    }
}
```

```
package controller;

import model.Student;
import model.Sex;

public class StudentCreator {
    public static Student createStudent(String firstName, String lastName, String middleName, Sex sex, String studentId) {
        return new Student(firstName, lastName, middleName, sex, studentId);
    }
}
```

```
package controller;

import model.University;
import model.Human;
import model.Faculty;

import java.util.List;

public class UniversityCreator {
    public static University createUniversity(String name, Human head, List<Faculty> faculties) {
        return new University(name, head, faculties);
    }
}
```

Main:

```
import model.*;
import controller.*;

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

public class Main {
    public static void main(String[] args) {
        List<Student> students = new ArrayList<>();

        students.add(StudentCreator.createStudent("Микита", "Помазан", "Максимович", Sex.MALE, "1"));
        students.add(StudentCreator.createStudent("Марія", "Петренко", "Анатоліївна", Sex.FEMALE, "2"));

        Human departmentHead = new Human("Михайл", "Джонсон", "Якович", Sex.MALE);

        List<Group> groups = new ArrayList<>();
        groups.add(GroupCreator.createGroup("Група А", departmentHead, students));
    }
}
```

```
List<Department> departments = new ArrayList<>();
departments.add(new Department("Комп'ютерні науки", departmentHead, groups));

Human facultyHead = new Human("Олена", "Шевченко", "Іванівна", Sex.FEMALE);
List<Faculty> faculties = new ArrayList<>();
faculties.add(FacultyCreator.createFaculty("IT", facultyHead, departments));

Human universityHead = new Human("Оксана", "Коваленко", "Миколаївна", Sex.FEMALE);
University university = UniversityCreator.createUniversity("Дніпровська політехніка",
universityHead, faculties);

System.out.println(university);
}
}
```

Висновок: Розробив програму для створення та обробки комплексного об'єкта "університет" відповідно до архітектурного патерну MVC. Забезпечити структуру проєкту через пакети model, controller та реалізувати відповідні класи для представлення університету та його підрозділів.