

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»



ЗВІТ
про виконання лабораторної роботи
з дисципліни
«Поглиблене програмування в середовищі java»
Лабораторна робота № 3

Виконав:

студент гр. 126-21-1

Мовлик Дмитро Дмитрович

Прийняв:

Доцент

Мінеєв Олександр Сергійович

Дніпро
2025

Хід роботи

1. 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).
2. В цій лабораторній роботі група View не потрібна.
3. 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.
4. В програмі активно рекомендується використовувати абстрактні класи та інтерфейси.

Все було реалізовано, та нижче буде перелічено коди програм.

Код контролерів (Рис.1-3)

```
package com.movlyk.LR3.controller;

import com.movlyk.LR3.model.Faculty;
import com.movlyk.LR3.model.Group;

public class GroupCreator { 1 usage  👤 dimam
    public static void addGroup(Faculty faculty, Group group) { 1 usage  👤 dimam
        faculty.addGroup(group);
    }
}
```

Рис.1 - код контролера GroupCreator

```
upCreator.java  © StudentCreator.java  ×

package com.movlyk.LR3.controller;

import com.movlyk.LR3.model.Group;
import com.movlyk.LR3.model.Student;

public class StudentCreator { 2 usages  👤 dimam
    public static void addStudent(Group group, Student student) { 2 usages  👤 dimam
        group.addStudent(student);
    }
}
```

Рис.2 – код контролера StudentCreator

```
upCreator.java  © StudentCreator.java  © UniversityCreator.java  ×

package com.movlyk.LR3.controller;

import com.movlyk.LR3.model.Faculty;
import com.movlyk.LR3.model.University;

public class UniversityCreator { 1 usage  👤 dimam
    public static void addFaculty(University university, Faculty faculty) { 1 usage  👤 dimam
        university.addFaculty(faculty);
    }
}
```

Рис.3 - код контролера UniversityCreator

Коди реалізації з папки model (Рис.4-9)

```
© Faculty.java ×
1  package com.movlyk.LR3.model;
2
3  import java.util.ArrayList;
4  import java.util.List;
5
6  public class Faculty { 10 usages  ⚡ dimam
7      private String name; 2 usages
8      private List<Group> groups = new ArrayList<>(); 2 usages
9
10     public Faculty(String name) { 1 usage  ⚡ dimam
11         this.name = name;
12     }
13
14     public void addGroup(Group group) { 1 usage  ⚡ dimam
15         groups.add(group);
16     }
17
18     public String getName() { 1 usage  ⚡ dimam
19         return name;
20     }
21
22     public List<Group> getGroups() { 1 usage  ⚡ dimam
23         return groups;
24     }
25 }
```

Рис.4 – код програми Faculty

```
© Group.java ×
1  package com.movlyk.LR3.model;
2
3  import java.util.ArrayList;
4  import java.util.List;
5
6  public class Group { 10 usages  ⚡ dimam
7      private String name; 2 usages
8      private List<Student> students = new ArrayList<>(); 2 usages
9
10     public Group(String name) { 1 usage  ⚡ dimam
11         this.name = name;
12     }
13
14     public void addStudent(Student student) { 1 usage  ⚡ dimam
15         students.add(student);
16     }
17
18     public String getName() { 1 usage  ⚡ dimam
19         return name;
20     }
21
22     public List<Student> getStudents() { 1 usage  ⚡ dimam
23         return students;
24     }
25 }
```

Рис.5 - код програми Group

```

Human.java  © Human.java  ×
package com.movlyk.LR3.model;

public class Human { 6 usages 1 inheritor  ⬆ dimam
    private String firstName; 2 usages
    private String lastName; 2 usages
    private String middleName; 2 usages
    private Sex gender; 2 usages

    public Human(String firstName, String lastName, String middleName, Sex gender) { 2 usages  ⬆ dimam
        this.firstName = firstName;
        this.lastName = lastName;
        this.middleName = middleName;
        this.gender = gender;
    }

    public String getFullName() { 2 usages  ⬆ dimam
        return lastName + " " + firstName + " " + middleName;
    }

    public Sex getGender() { no usages  ⬆ dimam
        return gender;
    }
}

```

Рис.6 - код програми Human

```

Sex.java  ×
1 package com.movlyk.LR3.model;
2
3 public enum Sex { 7 usages  ⬆ dimam
4     MALE, FEMALE; 2 usages
5 }

```

Рис.7 – код програми Sex

```

Student.java  ×
1 package com.movlyk.LR3.model;
2
3 public class Student extends Human { 10 usages  ⬆ dimam
4     private String group; 2 usages
5
6     public Student(String firstName, String lastName, String middleName, Sex gender, String group) { 2 usages  ⬆ dimam
7         super(firstName, lastName, middleName, gender);
8         this.group = group;
9     }
10
11     public String getGroup() { 1 usage  ⬆ dimam
12         return group;
13     }
14 }

```

Рис.8 – код програми Student

```

5
6 public class University { 4 usages 1 dimam
7     private String name; 2 usages
8     private Human rector; 2 usages
9     private List<Faculty> faculties = new ArrayList<>(); 2 usages
10
11     public University(String name, Human rector) { 1 usage 1 dimam
12         this.name = name;
13         this.rector = rector;
14     }
15
16     public void addFaculty(Faculty faculty) { 1 usage 1 dimam
17         faculties.add(faculty);
18     }
19
20     public String getName() { 1 usage 1 dimam
21         return name;
22     }
23
24     public Human getRector() { 1 usage 1 dimam
25         return rector;
26     }
27
28     public List<Faculty> getFaculties() { 1 usage 1 dimam
29         return faculties;
30     }
31 }

```

Рис.9 - код програми University

Код реалізації програми, просте кажучи Runner. (Рис.10-11)

```

1 package com.movlyk.LR3.view;
2
3 import com.movlyk.LR3.model.*;
4 import com.movlyk.LR3.controller.*;
5
6 public class Run { 1 dimam
7     public static void main(String[] args) { 1 dimam
8         Human rector = new Human("Іван", "Петров", "Михайлович", Sex.MALE);
9         University university = new University("Київський університет", rector);
10
11         Faculty faculty = new Faculty("Факультет інформатики");
12         UniversityCreator.addFaculty(university, faculty); // Використовуємо UniversityCreator
13
14         Group group = new Group("IT-21");
15         GroupCreator.addGroup(faculty, group); // Використовуємо GroupCreator
16
17         Student student1 = new Student("Олександр", "Іванов", "Сергійович", Sex.MALE, "IT-21");
18         Student student2 = new Student("Марина", "Петренко", "Олексіївна", Sex.FEMALE, "IT-21");
19
20         StudentCreator.addStudent(group, student1);
21         StudentCreator.addStudent(group, student2);
22
23         System.out.println("Університет: " + university.getName());
24         System.out.println("Ректор: " + university.getRector().getFullName());
25
26         for (Faculty f : university.getFaculties()) {
27             System.out.println("Факультет: " + f.getName());
28             for (Group g : f.getGroups()) {
29                 System.out.println("Група: " + g.getName());
30                 for (Student s : g.getStudents()) {
31                     System.out.println("Студент: " + s.getFullName() + ", Група: " + s.getGroup());
32                 }
33             }
34         }
35     }
36 }

```

Рис.10 – Код запускаючий програму

```
Університет: Київський університет  
Ректор: Петров Іван Михайлович  
Факультет: Факультет інформатики  
Група: IT-21  
Студент: Іванов Олександр Сергійович, Група: IT-21  
Студент: Петренко Марина Олексіївна, Група: IT-21
```

Рис.11 – Результат програми

Висновок: у ході практичної роботи, була розроблена програма, що створює та обробляє комплексний об'єкт університету з використання парадигми mvc. Програма яка була створена складається з трьох основних частин: моделі виду та контролера. Всі сутності мають поля, та відповідають вимогам.