МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний технічний університет «Дніпровська політехніка»

Изображение выглядит как текст

Автоматически созданное описание

Кафедра ПЗКС

ЗВІТ

з лабораторної роботи №3

з дисципліни «Поглиблене програмування в середовищі Java»

Виконав: ст. гр. 122-21-3

Пак Даніїл Євгенович

Перевірив: Мінєєв О.С.

Дніпро

2025

**Тема:** ООП.

**Мета:** Опанувати ООП та створити програму що буде створювати та обробляти комплексний об'єкт під назвою університет (university) в середовищі Java.

**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, 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 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 createTypycalUniversity.

  В програмі активно рекомендується використовувати абстрактні класи та інтерфейси

**Хід роботи**

1. Виконуємо створення нової гілки та папку проекта

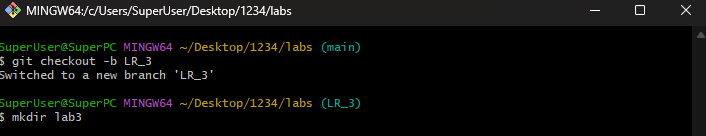


Рис. 1. Створення нової гілки

2. Створити package для контроллеру та моделі

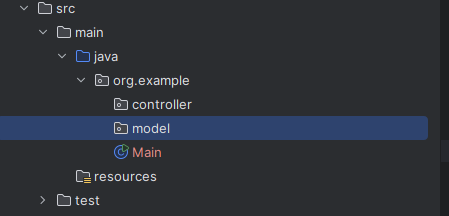


Рис. 2. Файлова структура створеного проекту

3. Створити моделі та контролери застосунку

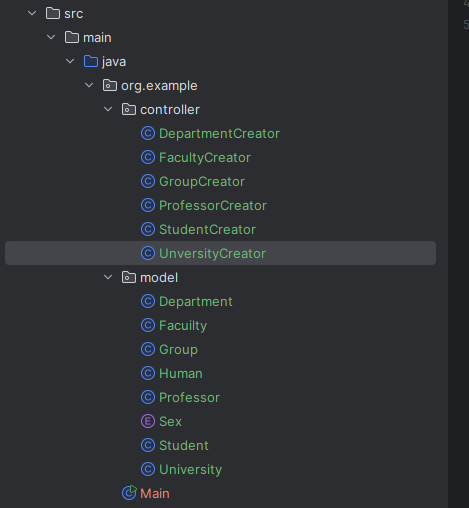
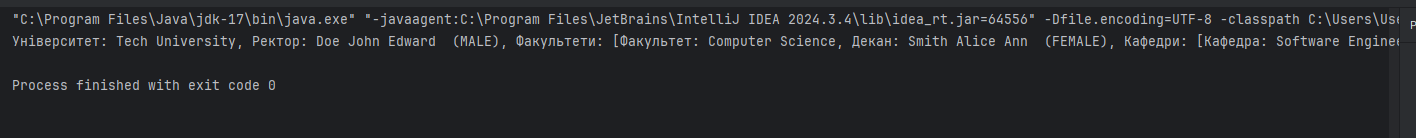


Рис. 3. Розроблена структура додатку

4. Відобразити результат виконання програми

Рис. 4. Результат виконання програми

5. Виконати злиття гілки «LR\_3» з основною

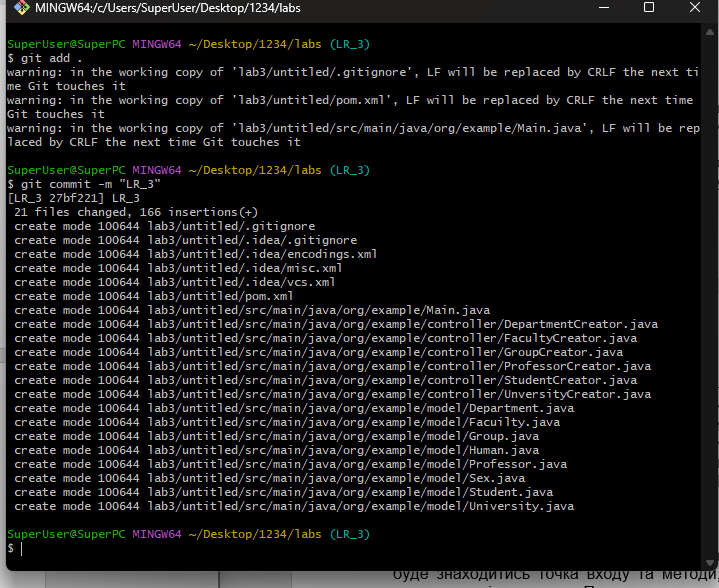


Рис. 5. Виконання коміту третьої лабораторної роботи

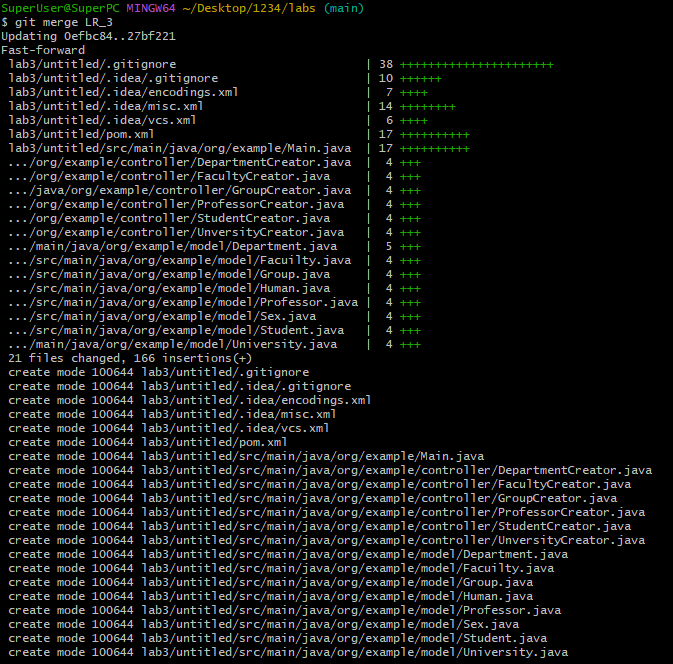


Рис. 6. Виконання злиття гілок

**Код програми**

**// Файл «Human.java»**

package org.example.model;

public abstract class Human {

private String firstName;

private String lastName;

private String middleName;

private Sex gender;

public Human(String firstName, String lastName, String middleName, Sex gender) {

this.firstName = firstName;

this.lastName = lastName;

this.middleName = middleName;

this.gender = gender;

}

@Override

public String toString() {

return lastName + " " + firstName + " " + middleName + " " + " (" + gender + ")";

}

}

**// Файл«Sex.java»**

package org.example.model;

public enum Sex {

MALE, FEMALE

}

*Лістинг «Student.java»*

package org.example.model;

public class Student extends Human {

public Student(String firstName, String lastName, String middleName, Sex gender) {

super(firstName, lastName, middleName, gender);

}

}

**// Файл «Professor.java»**

package org.example.model;

public class Professor extends Human {

public Professor(String firstName, String lastName, String middleName, Sex gender) {

super(firstName, lastName, middleName, gender);

}

}

**// Файл «Group.java»**

package org.example.model;

import java.util.ArrayList;

import java.util.List;

public class Group {

private String name;

private Human head;

private List<Student> students;

public Group(String name, Human head) {

this.name = name;

this.head = head;

this.students = new ArrayList<>();

}

public void addStudent(Student student) {

students.add(student);

}

public String getName() {

return name;

}

public Human getHead() {

return head;

}

public List<Student> getStudents() {

return students;

}

@Override

public String toString() {

return "Група: " + name + ", Куратор: " + head + ", Студенти: " + students;

}

}

**// Файл «Faculty.java»**

package org.example.model;

import java.util.ArrayList;

import java.util.List;

public class Faculty {

private String name;

private Human head;

private List<Department> departments;

public Faculty(String name, Human head) {

this.name = name;

this.head = head;

this.departments = new ArrayList<>();

}

public void addDepartment(Department department) {

departments.add(department);

}

public String getName() {

return name;

}

public Human getHead() {

return head;

}

public List<Department> getDepartments() {

return departments;

}

@Override

public String toString() {

return "Факультет: " + name + ", Декан: " + head + ", Кафедри: " + departments;

}

}

**// Файл «Department.java»**

package org.example.model;

import java.util.ArrayList;

import java.util.List;

public class Department {

private String name;

private Human head;

private List<Group> groups;

public Department(String name, Human head) {

this.name = name;

this.head = head;

this.groups = new ArrayList<>();

}

public void addGroup(Group group) {

groups.add(group);

}

public String getName() {

return name;

}

public Human getHead() {

return head;

}

public List<Group> getGroups() {

return groups;

}

@Override

public String toString() {

return "Кафедра: " + name + ", Завідувач: " + head + ", Групи: " + groups;

}

}

**// Файл «University.java»**

package org.example.model;

import java.util.ArrayList;

import java.util.List;

public class University {

private String name;

private Human head;

private List<Faculty> faculties;

public University(String name, Human head) {

this.name = name;

this.head = head;

this.faculties = new ArrayList<>();

}

public void addFaculty(Faculty faculty) {

faculties.add(faculty);

}

public String getName() {

return name;

}

public Human getHead() {

return head;

}

public List<Faculty> getFaculties() {

return faculties;

}

@Override

public String toString() {

return "Університет: " + name + ", Ректор: " + head + ", Факультети: " + faculties;

}

}

**// Файл «StudentCreator.java»**

package org.example.controller;

import org.example.model.Sex;

import org.example.model.Student;

public class StudentCreator {

public static Student createStudent(String firstName, String lastName, String middleName, Sex gender) {

return new Student(firstName, lastName, middleName, gender);

}

}

**// Файл «ProfessorCreator.java»**

package org.example.controller;

import org.example.model.Professor;

import org.example.model.Sex;

public class ProfessorCreator {

public static Professor createProfessor(String firstName, String lastName, String middleName, Sex gender) {

return new Professor(firstName, lastName, middleName, gender);

}

}

**// Файл** *«***FacultyCreator.java»**

package org.example.controller;

import org.example.model.Faculty;

import org.example.model.Human;

public class FacultyCreator {

public static Faculty createFaculty(String name, Human head) {

return new Faculty(name, head);

}

}

**// Файл «DepartmentCreator.java»**

package org.example.controller;

import org.example.model.Department;

import org.example.model.Human;

public class DepartmentCreator {

public static Department createDepartment(String name, Human head) {

return new Department(name, head);

}

}

**// Файл «GroupCreator.java»**

package org.example.controller;

import org.example.model.Group;

import org.example.model.Human;

public class GroupCreator {

public static Group createGroup(String name, Human head) {

return new Group(name, head);

}

}

**// Файл «UniversityCreator.java»**

package org.example.controller;

import org.example.model.\*;

public class UniversityCreator {

public static University createTypicalUniversity() {

Professor universityHead = ProfessorCreator.createProfessor("John", "Doe", "Edward", Sex.MALE);

University university = new University("Tech University", universityHead);

Professor facultyHead = ProfessorCreator.createProfessor("Alice", "Smith", "Ann", Sex.FEMALE);

Faculty faculty = FacultyCreator.createFaculty("Computer Science", facultyHead);

university.addFaculty(faculty);

Professor departmentHead = ProfessorCreator.createProfessor("Bob", "Brown", "Michael", Sex.MALE);

Department department = DepartmentCreator.createDepartment("Software Engineering", departmentHead);

faculty.addDepartment(department);

Professor groupHead = ProfessorCreator.createProfessor("Charlie", "Davis", "Lee", Sex.MALE);

Group group = GroupCreator.createGroup("SE-2025", groupHead);

department.addGroup(group);

group.addStudent(StudentCreator.createStudent("Eva", "White", "Maria", Sex.FEMALE));

group.addStudent(StudentCreator.createStudent("Daniel", "Black", "James", Sex.MALE));

return university;

}

}

**// Файл «Run.java»**

package org.example;

import org.example.controller.UniversityCreator;

import org.example.model.University;

public class Run {

public static void main(String[] args) {

University university = UniversityCreator.createTypicalUniversity();

System.out.println(university);

}

}

**Висновок**

Навчився створювати клас та застосовувати їх в програному додатку згідно концепції ООП. Створено комплексний об’єкт університет, та протестовано створений додаток.