Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра автоматизованих систем управління



**Звіт**

до лабораторної роботи № 2 з

курсу

*Прикладне програмувння*  на

тему:

**«Робота з класами»**

Виконав: студент ОІ-24

**Стащишин Юрій**

Прийняв:

**Чорненький В. Я.**

Львів – 2024

***Лабораторна робота № 2***

**Тема:** **Робота з класами**

**Мета роботи:** Мета цієї лабораторної роботи полягає в закріпленні навичок роботи з класами об'єктно-орієнтованого підходу в програмуванні на мові Java.

***Послідовність роботи:***

1. Створити проект, що складається з двох класів: основного (Main) та класу для представлення об’єкта відповідно специфікації, що наведена у таблиці 1. Кожний клас повинен бути розміщений у окремому пакеті. У створеному класі визначити приватні поля для зберігання указаних даних, конструктори для створення об’єктів та відкриті методи setValue(), getValue(), toString() для доступу до полів об’єкту.

2. В основному класі програми визначити методи, що створюють масив об'єктів. Задати критерії вибору даних та вивести ці дані на консоль. Для кожного критерію створити окремий метод.

3. Виконати програму, та пересвідчитись, що дані зберігаються та коректно виводяться на екран відповідно до вказаних критеріїв.

Варіант – 22

**Завдання лабораторної роботи**

Student: id, Прізвище, Ім'я, По батькові, Дата народження, Адреса, Телефон, Факультет, Курс, Група.

Скласти масив об'єктів. Вивести:

a) список студентів заданого факультету;

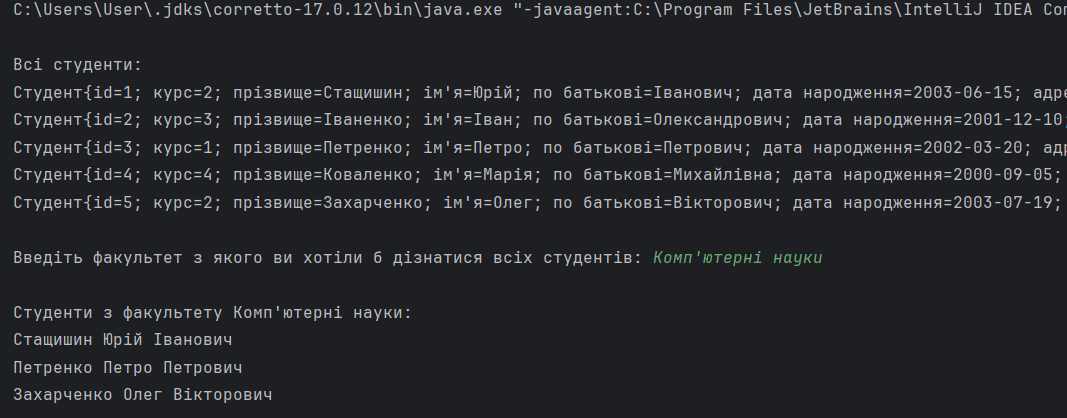
b) список студентів, які народились після заданого року;

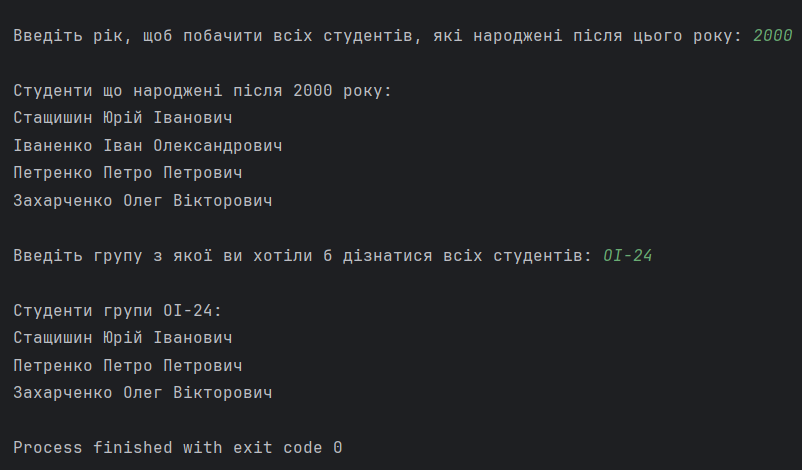
c) список навчальної групи.

**Код розробленої програми**

package student;  
  
import java.time.LocalDate;  
  
public class Student {  
 private int id;  
 private int course;  
 private String surname;  
 private String name;  
 private String middleName;  
 private LocalDate birthDay;  
 private String address;  
 private String phoneNumber;  
 private String faculty;  
 private String group;  
  
 */\*\*  
 \* Constructs a new Student with the specified details.  
 \*  
 \* @param id the student's ID  
 \* @param course the student's course number  
 \* @param surname the student's surname  
 \* @param name the student's first name  
 \* @param middleName the student's middle name  
 \* @param birthDay the student's birth date  
 \* @param address the student's address  
 \* @param phoneNumber the student's phone number  
 \* @param faculty the student's faculty  
 \* @param group the student's group  
 \*/* public Student(int id, int course, String surname, String name, String middleName, LocalDate birthDay,  
 String address, String phoneNumber, String faculty, String group) {  
 this.id = id;  
 this.course = course;  
 this.surname = surname;  
 this.name = name;  
 this.middleName = middleName;  
 this.birthDay = birthDay;  
 this.address = address;  
 this.phoneNumber = phoneNumber;  
 this.faculty = faculty;  
 this.group = group;  
 }  
  
 */\*\*  
 \* @return a string representation of the student  
 \*/* public String toString(){  
 return "Студент{" +  
 "id=" + id +  
 "; курс=" + course +  
 "; прізвище=" + surname +  
 "; ім'я=" + name +  
 "; по батькові=" + middleName +  
 "; дата народження=" + birthDay +  
 "; адреса=" + address +  
 "; номер телефону=" + phoneNumber +  
 "; факультет=" + faculty +  
 "; група=" + group + "}";  
  
 }  
  
 */\*\*  
 \* @param faculty prints the student's full name if their faculty matches the specified faculty.  
 \*/* public void studentByFaculty(String faculty){  
 if (faculty.equals(this.faculty)){  
 System.*out*.printf("%s %s %s\n", getSurname(), getName(), getMiddleName());  
 }  
 }  
  
 */\*\*  
 \* @param year prints the student's full name if they were born after the specified year.  
 \*/* public void printStudentBornAfter(int year){  
 if (this.birthDay.getYear() > year){  
 System.*out*.printf("%s %s %s\n", surname, name, middleName);  
 }  
 }  
  
 */\*\*  
 \* @param group prints the student's full name if their group matches the specified group.  
 \*/* public void printStudentInGroup(String group){  
 if (group.equals(this.group)){  
 System.*out*.printf("%s %s %s\n", surname, name, middleName);  
 }  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public int getCourse() {  
 return course;  
 }  
  
 public String getSurname() {  
 return surname;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public String getMiddleName() {  
 return middleName;  
 }  
  
 public LocalDate getBirthDay() {  
 return birthDay;  
 }  
  
 public String getAddress() {  
 return address;  
 }  
  
 public String getPhoneNumber() {  
 return phoneNumber;  
 }  
  
 public String getFaculty() {  
 return faculty;  
 }  
  
 public String getGroup() {  
 return group;  
 }  
  
  
 public void setId(int id) {  
 if (id <= 0){  
 System.*out*.println("Ви ввели неправильне ID\n");  
 return;  
 }  
 this.id = id;  
 }  
  
 public void setCourse(int course) {  
 if (course <= 0){  
 System.*out*.println("Ви ввели невірний курс курс");  
 return;  
 }  
 this.course = course;  
 }  
  
 public void setSurname(String surname) {  
 this.surname = surname;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public void setMiddleName(String middleName) {  
 this.middleName = middleName;  
 }  
  
 public void setBirthDay(LocalDate birthDay) {  
 this.birthDay = birthDay;  
 }  
  
 public void setAddress(String address) {  
 this.address = address;  
 }  
  
 public void setPhoneNumber(String phoneNumber) {  
 this.phoneNumber = phoneNumber;  
 }  
  
 public void setFaculty(String faculty) {  
 this.faculty = faculty;  
 }  
  
 public void setGroup(String group) {  
 this.group = group;  
 }  
}

import student.Student;  
  
import java.util.Scanner;  
import java.time.LocalDate;  
import java.util.ArrayList;  
import java.util.List;  
  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 List<Student> students = new ArrayList<>();  
 students.add(new Student(1, 2, "Стащишин", "Юрій", "Іванович",  
 LocalDate.*of*(2003, 6, 15), "Львів, вул. Січових Стрільців, 1",  
 "0987654321", "Комп'ютерні науки", "ОІ-24"));  
  
 students.add(new Student(2, 3, "Іваненко", "Іван", "Олександрович",  
 LocalDate.*of*(2001, 12, 10), "Київ, вул. Шевченка, 10",  
 "0671234567", "Економіка", "Е-32"));  
  
 students.add(new Student(3, 1, "Петренко", "Петро", "Петрович",  
 LocalDate.*of*(2002, 3, 20), "Львів, вул. Личаківська, 7",  
 "0936543210", "Комп'ютерні науки", "ОІ-24"));  
  
 students.add(new Student(4, 4, "Коваленко", "Марія", "Михайлівна",  
 LocalDate.*of*(2000, 9, 5), "Одеса, вул. Дерибасівська, 15",  
 "0509876543", "Математика", "М-21"));  
  
 students.add(new Student(5, 2, "Захарченко", "Олег", "Вікторович",  
 LocalDate.*of*(2003, 7, 19), "Харків, вул. Сумська, 22",  
 "0734567890", "Комп'ютерні науки", "ОІ-24"));  
  
 System.*out*.println("\nВсі студенти:");  
 for (Student stud : students){  
 System.*out*.println(stud.toString());  
 }  
  
 System.*out*.print("\nВведіть факультет з якого ви хотіли б дізнатися всіх студентів: ");  
 String faculty = scanner.nextLine();  
 System.*out*.printf("\nСтуденти з факультету %s:\n", faculty);  
 for (Student stud : students){  
 stud.studentByFaculty(faculty);  
 }  
  
 int year;  
 while (true) {  
 System.*out*.print("\nВведіть рік, щоб побачити всіх студентів, які народжені після цього року: ");  
 if (scanner.hasNextInt()) {  
 year = scanner.nextInt();  
 break;  
 } else {  
 System.*out*.println("Помилка: Введіть ціле число.");  
 scanner.next();  
 }  
 }  
 System.*out*.printf("\nСтуденти що народжені після %d року:\n", year);  
 for (Student stud : students){  
 stud.printStudentBornAfter(year);  
 }  
 scanner.nextLine();  
  
 System.*out*.print("\nВведіть групу з якої ви хотіли б дізнатися всіх студентів: ");  
 String group = scanner.nextLine();  
 System.*out*.printf("\nСтуденти групи %s:\n", group);  
 for (Student stud : students){  
 stud.printStudentInGroup(group);  
 }  
  
 scanner.close();  
 }  
}

**Результат**



**Висновок**

Під час виконання лабораторної роботи було закріплено навички роботи з класами у середовищі Java, а саме — створення класів та їх екземплярів, використання інкапсуляції, методів доступу до полів (гетери та сетери), а також методів для виконання різних операцій на об'єктах класів.

Посилання на Git: https://github.com/Float36/labs