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 дисципліни

*Прикладне програмування*

# на тему:

## “Робота з класами”

Виконала: студентка групи ОІ-24

Окілка Марта

Прийняв: асистент каф. АСУ

Чорненький В. Я.

**Львів – 2024**

***Лабораторна робота № 2***

**Тема роботи: “Робота з класами”**

**Мета роботи:** опанувати основи ООП на Java, зокрема. Навчитися створювати та використовувати класи, визначати їх поля, конструктори та методи. Навчитися створювати масиви об'єктів та працювати з ними. Навчитися розробляти алгоритми пошуку та фільтрації даних в масивах об'єктів за заданими критеріями. Здобути практичні навички роботи з класами, об'єктами, масивами та методами в Java.

**Завдання лабораторної роботи**

**1.** Створити проект, що складається з двох класів: основного (Main) та класу для представлення об’єкта відповідно специфікації, що наведена у варіанті індивідуального завдання. 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 ( використати метод toString() ). Для кожного критерію створити окремий метод.

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

**Індивідуальне завдання:**

**Варіант – 7 (17)**

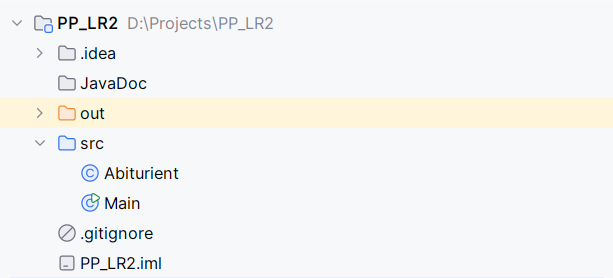
Abiturient: id, Прізвище, Ім'я, По батькові, Адреса, Телефон, Середній бал.

Скласти масив об'єктів. Вивести:

1. список абітурієнтів із вказаним іменем;
2. список абітурієнтів, середній бал у яких вище заданого;
3. вибрати задане число n абітурієнтів, що мають найвищий середній бал.

**Виконання завдань лабораторної роботи:**

**1.** Структура проєкту:



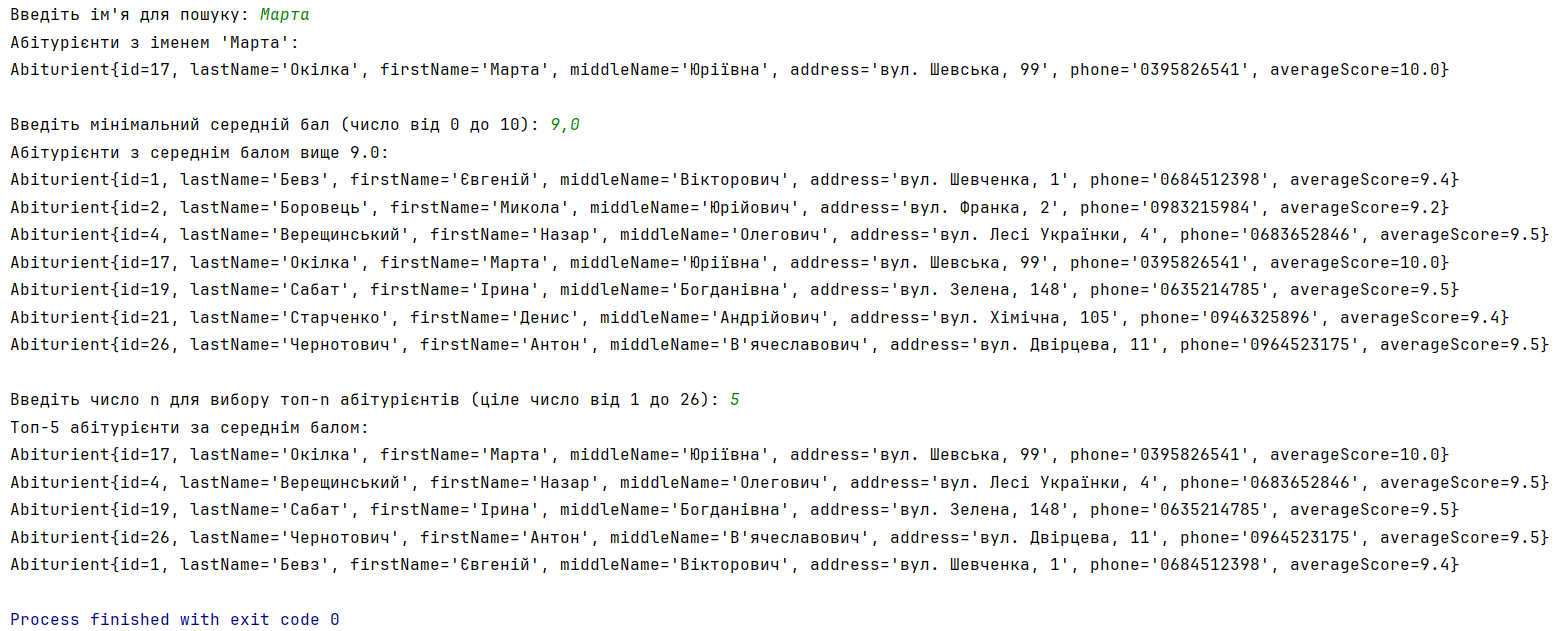
**2.** Вміст файлу Abiturient.java:

import java.util.Arrays;  
import java.util.Comparator;  
import java.util.InputMismatchException;  
import java.util.Scanner;  
  
*/\*\*  
 \* Клас Abiturient представляє абітурієнта з його ідентифікатором, прізвищем, ім'ям, по батькові, адресою, номером телефону та середнім балом.  
 \*/*public class Abiturient {  
  
 private int id;  
 private String lastName;  
 private String firstName;  
 private String middleName;  
 private String address;  
 private String phone;  
 private double averageScore;  
  
 */\*\*  
 \* Конструктор для створення об'єкта Abiturient.  
 \*  
 \* @param id ідентифікатор абітурієнта  
 \* @param lastName прізвище абітурієнта  
 \* @param firstName ім'я абітурієнта  
 \* @param middleName по батькові абітурієнта  
 \* @param address адреса абітурієнта  
 \* @param phone номер телефону абітурієнта  
 \* @param averageScore середній бал абітурієнта  
 \*/* public Abiturient(int id, String lastName, String firstName, String middleName, String address, String phone, double averageScore) {  
 this.id = id;  
 this.lastName = lastName;  
 this.firstName = firstName;  
 this.middleName = middleName;  
 this.address = address;  
 this.phone = phone;  
 this.averageScore = averageScore;  
 }  
  
 *//Геттери та сеттери  
 /\*\*  
 \* Повертає ідентифікатор абітурієнта.  
 \*  
 \* @return ідентифікатор абітурієнта  
 \*/* public int getId() {  
 return id;  
 }  
  
 */\*\*  
 \* Встановлює ідентифікатор абітурієнта.  
 \*  
 \* @param id новий ідентифікатор абітурієнта  
 \*/* public void setId(int id) {  
 this.id = id;  
 }  
  
 */\*\*  
 \* Повертає прізвище абітурієнта.  
 \*  
 \* @return прізвище абітурієнта  
 \*/* public String getLastName() {  
 return lastName;  
 }  
  
 */\*\*  
 \* Встановлює прізвище абітурієнта.  
 \*  
 \* @param lastName нове прізвище абітурієнта  
 \*/* public void setLastName(String lastName) {  
 this.lastName = lastName;  
 }  
  
 */\*\*  
 \* Повертає ім'я абітурієнта.  
 \*  
 \* @return ім'я абітурієнта  
 \*/* public String getFirstName() {  
 return firstName;  
 }  
  
 */\*\*  
 \* Встановлює ім'я абітурієнта.  
 \*  
 \* @param firstName нове ім'я абітурієнта  
 \*/* public void setFirstName(String firstName) {  
 this.firstName = firstName;  
 }  
  
 */\*\*  
 \* Повертає по батькові абітурієнта.  
 \*  
 \* @return по батькові абітурієнта  
 \*/* public String getMiddleName() {  
 return middleName;  
 }  
  
 */\*\*  
 \* Встановлює по батькові абітурієнта.  
 \*  
 \* @param middleName нове по батькові абітурієнта  
 \*/* public void setMiddleName(String middleName) {  
 this.middleName = middleName;  
 }  
  
 */\*\*  
 \* Повертає адресу абітурієнта.  
 \*  
 \* @return адреса абітурієнта  
 \*/* public String getAddress() {  
 return address;  
 }  
  
 */\*\*  
 \* Встановлює адресу абітурієнта.  
 \*  
 \* @param address нова адреса абітурієнта  
 \*/* public void setAddress(String address) {  
 this.address = address;  
 }  
  
 */\*\*  
 \* Повертає номер телефону абітурієнта.  
 \*  
 \* @return номер телефону абітурієнта  
 \*/* public String getPhone() {  
 return phone;  
 }  
  
 */\*\*  
 \* Встановлює номер телефону абітурієнта.  
 \*  
 \* @param phone новий номер телефону абітурієнта  
 \*/* public void setPhone(String phone) {  
 this.phone = phone;  
 }  
  
 */\*\*  
 \* Повертає середній бал абітурієнта.  
 \*  
 \* @return середній бал абітурієнта  
 \*/* public double getAverageScore() {  
 return averageScore;  
 }  
  
 */\*\*  
 \* Встановлює середній бал абітурієнта.  
 \*  
 \* @param averageScore новий середній бал абітурієнта  
 \*/* public void setAverageScore(double averageScore) {  
 this.averageScore = averageScore;  
 }  
  
 */\*\*  
 \* Перевизначений метод для рядкового представлення об'єкта Abiturient.  
 \*  
 \* @return рядкове представлення об'єкта Abiturient  
 \*/* @Override  
 public String toString() {  
 return "Abiturient{" +  
 "id=" + id +  
 ", lastName='" + lastName + '\'' +  
 ", firstName='" + firstName + '\'' +  
 ", middleName='" + middleName + '\'' +  
 ", address='" + address + '\'' +  
 ", phone='" + phone + '\'' +  
 ", averageScore=" + averageScore +  
 '}';  
 }  
}

**3.** Вміст файлу Main.java:

import java.util.Arrays;  
import java.util.Comparator;  
import java.util.InputMismatchException;  
import java.util.Scanner;  
  
*/\*\*  
 \* Головний клас програми для роботи з масивом абітурієнтів.  
 \* Дозволяє знаходити абітурієнтів за іменем, середнім балом вище заданого та виводити топ-n абітурієнтів за середнім балом.  
 \*/*public class Main {  
  
 */\*\*  
 \* Головний метод програми.  
 \*  
 \* @param args аргументи командного рядка (не використовуються)  
 \*/* public static void main(String[] args) {  
  
 *// Створення масиву об'єктів Abiturient* Abiturient[] abiturients = new Abiturient[26];  
  
 *// Заповнення масиву тестовими даними* abiturients[0] = new Abiturient(1, "Бевз", "Євгеній", "Вікторович", "вул. Шевченка, 1", "0684512398", 9.4);  
 abiturients[1] = new Abiturient(2, "Боровець", "Микола", "Юрійович", "вул. Франка, 2", "0983215984", 9.2);  
 abiturients[2] = new Abiturient(3, "Веремчук", "Тарас", "Олександрович", "вул. Грушевського, 3", "0684578912", 7.8);  
 abiturients[3] = new Abiturient(4, "Верещинський", "Назар", "Олегович", "вул. Лесі Українки, 4", "0683652846", 9.5);  
 abiturients[4] = new Abiturient(5, "Вівчар", "Олександр", "Вікторович", "вул. Хмельницького, 5", "0987245698", 8.0);  
 abiturients[5] = new Abiturient(6, "Гунькін", "Михайло", "Юрійович", "вул. Лінкольна, 33", "0986674298", 3.2);  
 abiturients[6] = new Abiturient(7, "Дмитрів", "Юрій", "Романович", "вул. Городоцька, 156", "0688874129", 7.2);  
 abiturients[7] = new Abiturient(8, "Живоронко", "Роман", "Ігорович", "вул. Патона, 10", "0986654657", 7.9);  
 abiturients[8] = new Abiturient(9, "Зубач", "Руслан", "Романович", "вул. Бандери, 96", "0986634845", 8.0);  
 abiturients[9] = new Abiturient(10, "Івануса", "Олена", "Олександрівна", "вул. Стрийська, 120", "0984575369", 6.8);  
 abiturients[10] = new Abiturient(11, "Кіц", "Маркіян", "Романович", "вул. Наукова, 99", "0685213249", 8.4);  
 abiturients[11] = new Abiturient(12, "Кіш", "Ніколетт", "Василівна", "вул. Авіаційна, 32", "0665248921", 8.7);  
 abiturients[12] = new Abiturient(13, "Круглова", "Анастасія", "Романівна", "вул. Антоновича, 18", "0632548951", 8.1);  
 abiturients[13] = new Abiturient(14, "Левкович", "Назарій", "Петрович", "вул. Барвінських, 42", "0932233651", 5.8);  
 abiturients[14] = new Abiturient(15, "Мірошниченко", "Михайло", "Олександрович", "вул. Яблунева, 100", "0974125836", 6.9);  
 abiturients[15] = new Abiturient(16, "Ніжегородова", "Анастасія", "Василівна", "вул. Щтрецька, 47", "0985236974", 8.6);  
 abiturients[16] = new Abiturient(17, "Окілка", "Марта", "Юріївна", "вул. Шевська, 99", "0395826541", 10.0);  
 abiturients[17] = new Abiturient(18, "Олексюк", "Ілля", "Романович", "вул. Шпитальна, 21", "0985541235", 8.6);  
 abiturients[18] = new Abiturient(19, "Сабат", "Ірина", "Богданівна", "вул. Зелена, 148", "0635214785", 9.5);  
 abiturients[19] = new Abiturient(20, "Сикуринець", "Руслан", "Русланович", "вул. Чорновола, 82", "0965832147", 7.8);  
 abiturients[20] = new Abiturient(21, "Старченко", "Денис", "Андрійович", "вул. Хімічна, 105", "0946325896", 9.4);  
 abiturients[21] = new Abiturient(22, "Стащишин", "Юрій", "Ярославович", "вул. Університецька, 28", "0931296345", 7.9);  
 abiturients[22] = new Abiturient(23, "Тріщук", "Максим", "Андрійович", "вул. Руданського, 41", "0968745612", 7.8);  
 abiturients[23] = new Abiturient(24, "Умриш", "Софія", "Романівна", "вул. Міцкевича, 753", "0985566321", 8.4);  
 abiturients[24] = new Abiturient(25, "Урбан", "Володимир", "Володимирович", "вул. Залізнична, 102", "0987452136", 9.0);  
 abiturients[25] = new Abiturient(26, "Чернотович", "Антон", "В'ячеславович", "вул. Двірцева, 11", "0964523175", 9.5);  
  
 Scanner scanner = new Scanner(System.*in*);  
  
 *// a) Список абітурієнтів із вказаним іменем* System.*out*.print("Введіть ім'я для пошуку: ");  
 String firstName = scanner.nextLine();  
 System.*out*.println("Абітурієнти з іменем '" + firstName + "':");  
 for (Abiturient abiturient : *findByFirstName*(abiturients, firstName)) {  
 System.*out*.println(abiturient);  
 }  
  
 *// b) Список абітурієнтів, середній бал у яких вище заданого* double minAverageScore = 0.0; *// Ініціалізація для уникнення помилки компіляції* while (true) {  
 System.*out*.print("\nВведіть мінімальний середній бал (число від 0 до 10): ");  
 try {  
 minAverageScore = scanner.nextDouble();  
 if (minAverageScore >= 0 && minAverageScore <= 10) {  
 break; *// Вихід з циклу, якщо дані введені коректно* } else {  
 System.*out*.println("Помилка: введіть число від 0 до 10.");  
 }  
 } catch (InputMismatchException e) {  
 System.*out*.println("Помилка: введено не числове значення.");  
 scanner.next(); *// Очищення буфера сканера* }  
 }  
 System.*out*.println("Абітурієнти з середнім балом вище " + minAverageScore + ":");  
 for (Abiturient abiturient : *findByAverageScoreHigherThan*(abiturients, minAverageScore)) {  
 System.*out*.println(abiturient);  
 }  
  
 *// c) Вибрати задане число n абітурієнтів, що мають найвищий середній бал* int n = 0; *// Ініціалізація для уникнення помилки компіляції* while (true) {  
 System.*out*.print("\nВведіть число n для вибору топ-n абітурієнтів (ціле число від 1 до " + abiturients.length + "): ");  
 try {  
 n = scanner.nextInt();  
 if (n >= 1 && n <= abiturients.length) {  
 break; *// Вихід з циклу, якщо дані введені коректно* } else {  
 System.*out*.println("Помилка: введіть ціле число від 1 до " + abiturients.length + ".");  
 }  
 } catch (InputMismatchException e) {  
 System.*out*.println("Помилка: введено не ціле число.");  
 scanner.next(); *// Очищення буфера сканера* }  
 }  
 System.*out*.println("Топ-" + n + " абітурієнти за середнім балом:");  
 for (Abiturient abiturient : *findTopNByAverageScore*(abiturients, n)) {  
 System.*out*.println(abiturient);  
 }  
  
 scanner.close();  
 }  
  
 */\*\*  
 \* Знаходить абітурієнтів з заданим іменем.  
 \*  
 \* @param abiturients масив абітурієнтів  
 \* @param firstName ім'я для пошуку  
 \* @return масив абітурієнтів з заданим іменем  
 \*/* public static Abiturient[] findByFirstName(Abiturient[] abiturients, String firstName) {  
 int count = 0;  
 for (Abiturient abiturient : abiturients) {  
 if (abiturient.getFirstName().equals(firstName)) {  
 count++;  
 }  
 }  
  
 Abiturient[] result = new Abiturient[count];  
 int index = 0;  
 for (Abiturient abiturient : abiturients) {  
 if (abiturient.getFirstName().equals(firstName)) {  
 result[index++] = abiturient;  
 }  
 }  
 return result;  
 }  
  
 */\*\*  
 \* Знаходить абітурієнтів з середнім балом вище заданого.  
 \*  
 \* @param abiturients масив абітурієнтів  
 \* @param score мінімальний середній бал  
 \* @return масив абітурієнтів з середнім балом вище заданого  
 \*/* public static Abiturient[] findByAverageScoreHigherThan(Abiturient[] abiturients, double score) {  
 int count = 0;  
 for (Abiturient abiturient : abiturients) {  
 if (abiturient.getAverageScore() > score) {  
 count++;  
 }  
 }  
  
 Abiturient[] result = new Abiturient[count];  
 int index = 0;  
 for (Abiturient abiturient : abiturients) {  
 if (abiturient.getAverageScore() > score) {  
 result[index++] = abiturient;  
 }  
 }  
 return result;  
 }  
  
 */\*\*  
 \* Знаходить топ-n абітурієнтів за середнім балом.  
 \*  
 \* @param abiturients масив абітурієнтів  
 \* @param n число абітурієнтів для вибору  
 \* @return масив топ-n абітурієнтів за середнім балом  
 \*/* public static Abiturient[] findTopNByAverageScore(Abiturient[] abiturients, int n) {  
 Arrays.*sort*(abiturients, Comparator.*comparingDouble*(Abiturient::getAverageScore).reversed());  
 return Arrays.*copyOf*(abiturients, Math.*min*(n, abiturients.length));  
 }  
}

**4.** Зразок виконання програми:



**5.** Посилання на публічний GitHub репозиторій:

https://github.com/OkayMarta/Applied-programming.-Java/tree/main/lab2

**Висновок**

В ході виконання лабораторної роботи було успішно створено проект на Java, що складається з класу для представленя абіткр’єнта та основного класу Main. Було реалізовано функціонал для створення масивів об'єктів, а також методи для пошуку та фільтрації даних за заданими критеріями. В результаті виконання роботи я закріпила знання з об'єктно-орієнтованого програмування, отримала практичні навички роботи з класами, об'єктами, масивами та методами в Java. Було продемонстровано, що дані коректно зберігаються в об'єктах та виводяться на екран відповідно до заданих критеріїв. Лабораторна робота дозволила мені поглибити розуміння принципів ООП та їх застосування на практиці.