BÀI THỰC HÀNH

Lập trình Java

Viết chương trình quản lý sinh viên (StudentManagement).

Mỗi đối tượng sinh viên có các thuộc tính sau: id, name, age, address và gpa (điểm trung bình). Yêu cầu: tạo ra một menu với các chức năng sau:

/***/

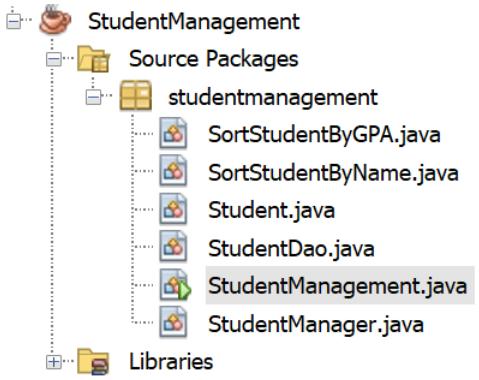
1. Add student.
2. Edit student by id.
3. Delete student by id.
4. Sort student by gpa.
5. Sort student by name.
6. Show student.
0. Exit.

/***/

Yêu cầu: Danh sách sinh viên được lưu vào file "***student.txt***" CSDL.

Bài làm:

- Cấu trúc của project



- Lớp ***Student*** để lưu thông tin cho mỗi sinh viên.
- Lớp ***StudentDao*** để đọc và ghi sinh viên vào file.
- Lớp ***SortStudentByGPA*** được implements Comparator để sắp xếp sinh viên tăng dần theo điểm trung bình.
- Lớp ***SortStudentByName*** được implements Comparator để sắp xếp sinh viên tăng dần theo tên.
- Lớp ***StudentManager*** cung cấp các phương thức để quản lý sinh viên như thêm, sửa, xóa, sắp xếp và hiển thị sinh viên.
- Lớp ***StudentManagerment*** chứa phương thức public static void main() để chạy ứng dụng và menu như yêu cầu của bài toán.

a) Student.java

```
package studentmanagement;
import java.io.Serializable;

// Lớp Student có thể tuần tự hoá
public class Student implements Serializable {
    private int id;
    private String name;
    private byte age;
    private String address;
    /* điểm trung bình của sinh viên */
    private float gpa;

    public Student() {}

    public Student(int id, String name, byte age,
            String address, float gpa) {
        super();
        this.id = id;
        this.name = name;
        this.age = age;
        this.address = address;
        this.gpa = gpa;
    }

    public int getId() {
```

```
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public byte getAge() {
        return age;
    }
    public void setAge(byte age) {
        this.age = age;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public float getGpa() {
        return gpa;
    }
    public void setGpa(float gpa) {
        this.gpa = gpa;
    }
}
```

b) StudentDao.java

```
package studentmanagement;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
```

```java
import java.io.OutputStream;
import java.util.ArrayList;
import java.util.List;

public class StudentDao {
    // tệp lưu trữ các object đã tuần tự hoá
    private static final String STUDENT_FILE_NAME = "student.txt";

    public void write(List<Student> studentList) {
        FileOutputStream fos = null;
        ObjectOutputStream oos = null;
        try {
            fos = new FileOutputStream(new File(STUDENT_FILE_NAME));
            oos = new ObjectOutputStream(fos);
            oos.writeObject(studentList);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            closeStream(fos);
            closeStream(oos);
        }
    }

    public List<Student> read() {
        List<Student> studentList = new ArrayList<>();
        FileInputStream fis = null;
        ObjectInputStream ois = null;
        try {
            fis = new FileInputStream(new File(STUDENT_FILE_NAME));
            ois = new ObjectInputStream(fis);
            studentList = (List<Student>) ois.readObject();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } finally {
            closeStream(fis);
            closeStream(ois);
        }
        return studentList;
    }
```

```
    private void closeStream(InputStream is) {
        if (is != null) {
            try {
                is.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    private void closeStream(OutputStream os) {
        if (os != null) {
            try {
                os.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

c) SortStudentByGPA.java

```
package studentmanagement;

import java.util.Comparator;

public class SortStudentByGPA implements Comparator<Student> {
    @Override
    public int compare(Student student1, Student student2) {
        if (student1.getGpa() > student2.getGpa()) {
            return 1;
        }
        return -1;
    }
}
```

d) SortStudentByName.java

```
package studentmanagement;
```

```java
import java.util.Comparator;

public class SortStudentByName implements Comparator<Student> {
    @Override
    public int compare(Student student1, Student student2) {
        return student1.getName().compareTo(student2.getName());
    }
}
```

e) StudentManager.java

```java
/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package studentmanagement;

import java.util.Collections;
import java.util.List;
import java.util.Scanner;

public class StudentManager {

    public static Scanner scanner = new Scanner(System.in);
    private List<Student> studentList;
    private StudentDao studentDao;

    public StudentManager() {
        studentDao = new StudentDao();
        studentList = studentDao.read();
    }

    public void add() {
        int id = (studentList.size() > 0) ? (studentList.size() + 1) : 1;
        System.out.println("student id = " + id);
        String name = inputName();
        byte age = inputAge();
        String address = inputAddress();
        float gpa = inputGpa();
        Student student = new Student(id, name, age, address, gpa);
```

```
        studentList.add(student);
        studentDao.write(studentList);
    }

    public void edit(int id) {
        boolean isExisted = false;
        int size = studentList.size();
        for (int i = 0; i < size; i++) {
            if (studentList.get(i).getId() == id) {
                isExisted = true;
                studentList.get(i).setName(inputName());
                studentList.get(i).setAge(inputAge());
                studentList.get(i).setAddress(inputAddress());
                studentList.get(i).setGpa(inputGpa());
                break;
            }
        }
        if (!isExisted) {
            System.out.printf("id = %d not existed.\n", id);
        } else {
            studentDao.write(studentList);
        }
    }

    public void delete(int id) {
        Student student = null;
        int size = studentList.size();
        for (int i = 0; i < size; i++) {
            if (studentList.get(i).getId() == id) {
                student = studentList.get(i);
                break;
            }
        }
        if (student != null) {
            studentList.remove(student);
            studentDao.write(studentList);
        } else {
            System.out.printf("id = %d not existed.\n", id);
        }
    }

    public void sortStudentByName() {
        Collections.sort(studentList, new SortStudentByName());
    }
```

```java
    public void sortStudentByGPA() {
        Collections.sort(studentList, new SortStudentByGPA());
    }

    public void show() {
        for (Student student : studentList) {
            System.out.format("%5d | ", student.getId());
            System.out.format("%20s | ", student.getName());
            System.out.format("%5d | ", student.getAge());
            System.out.format("%20s | ", student.getAddress());
            System.out.format("%10.1f%n", student.getGpa());
        }
    }

    public int inputId() {
        System.out.print("Input student id: ");
        while (true) {
            try {
                int id = Integer.parseInt((scanner.nextLine()));
                return id;
            } catch (NumberFormatException ex) {
                System.out.print("invalid! Input student id again: ");
            }
        }
    }

    private String inputName() {
        System.out.print("Input student name: ");
        return scanner.nextLine();
    }

    private String inputAddress() {
        System.out.print("Input student address: ");
        return scanner.nextLine();
    }

    private byte inputAge() {
        System.out.print("Input student age: ");
        while (true) {
            try {
                byte age = Byte.parseByte((scanner.nextLine()));
                if (age < 0 && age > 100) {
                    throw new NumberFormatException();
                }
                return age;
```

```
            } catch (NumberFormatException ex) {
                System.out.print("invalid! Input student id again: ");
            }
        }
    }

    private float inputGpa() {
        System.out.print("Input student gpa: ");
        while (true) {
            try {
                float gpa = Float.parseFloat((scanner.nextLine()));
                if (gpa < 0.0 && gpa > 10.0) {
                    throw new NumberFormatException();
                }
                return gpa;
            } catch (NumberFormatException ex) {
                System.out.print("invalid! Input student age again: ");
            }
        }
    }
    // getter && setter

    public List<Student> getStudentList() {
        return studentList;
    }

    public void setStudentList(List<Student> studentList) {
        this.studentList = studentList;
    }
}
```

f) StudentManagement.java

```
package studentmanagement;

import java.util.Scanner;

public class StudentManagement {

    public static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        String choose = null;
```

```
        boolean exit = false;
        StudentManager studentManager = new StudentManager();
        int studentId;
        // show menu
        showMenu();
        while (true) {
            choose = scanner.nextLine();
            switch (choose) {
                case "1":
                    studentManager.add();
                    break;
                case "2":
                    studentId = studentManager.inputId();
                    studentManager.edit(studentId);
                    break;
                case "3":
                    studentId = studentManager.inputId();
                    studentManager.delete(studentId);
                    break;
                case "4":
                    studentManager.sortStudentByGPA();
                    break;
                case "5":
                    studentManager.sortStudentByName();
                    break;
                case "6":
                    studentManager.show();
                    break;
                case "0":
                    System.out.println("exited!");
                    exit = true;
                    break;
                default:
                    System.out.println("invalid! please choose action in
below menu:");
                    break;
            }
            if (exit) {
                break;
            }
            // show menu
            showMenu();
        }

    }
```

```
    public static void showMenu() {
        System.out.println("-----------menu------------");
        System.out.println("1. Add student.");
        System.out.println("2. Edit student by id.");
        System.out.println("3. Delete student by id.");
        System.out.println("4. Sort student by gpa.");
        System.out.println("5. Sort student by name.");
        System.out.println("6. Show student.");
        System.out.println("0. exit.");
        System.out.println("--------------------------");
        System.out.print("Please choose: ");
    }

}
```

<<KẾT QUẢ CHẠY CHƯƠNG TRÌNH>>

- Thêm một student

```
----------menu------------
1. Add student.
2. Edit student by id.
3. Delete student by id.
4. Sort student by gpa.
5. Sort student by name.
6. Show student.
0. exit.
--------------------------
Please choose: 1
student id = 2
Input student name: Ha Cong Tung
Input student age: 21
Input student address: Ha Nam
Input student gpa: 6.5
----------menu------------
1. Add student.
2. Edit student by id.
3. Delete student by id.
4. Sort student by gpa.
5. Sort student by name.
6. Show student.
0. exit.
--------------------------
Please choose: 6
    1 |                  Nam |    21 |               Ha Noi |        9.5
    2 |         Ha Cong Tung |    21 |               Ha Nam |        6.5
```

- Sửa student

```
----------menu------------
1. Add student.
2. Edit student by id.
3. Delete student by id.
4. Sort student by gpa.
5. Sort student by name.
6. Show student.
0. exit.
--------------------------
Please choose: 2
Input student id: 1
Input student name: Pham Thanh Nam
Input student age: 22
Input student address: Thai Binh
Input student gpa: 8.5
----------menu------------
1. Add student.
2. Edit student by id.
3. Delete student by id.
4. Sort student by gpa.
5. Sort student by name.
6. Show student.
0. exit.
--------------------------
Please choose: 6
    1 |       Pham Thanh Nam |    22 |            Thai Binh |        8.5
    2 |         Ha Cong Tung |    21 |               Ha Nam |        6.5
```

- Hiển thị student

```
----------menu------------
1. Add student.
2. Edit student by id.
3. Delete student by id.
4. Sort student by gpa.
5. Sort student by name.
6. Show student.
0. exit.
--------------------------
Please choose: 6
    1 |       Pham Thanh Nam |    22 |            Thai Binh |        8.5
    2 |         Ha Cong Tung |    21 |               Ha Nam |        6.5
    3 |        Hoang Thi Lan |    21 |             Bac Ninh |        7.5
    4 |     Nguyen Cong Vinh |    21 |              ung Yen |        7.0
    5 |       Cao Van Nguyen |    21 |             Nam Dinh |        8.0
```

- Sort student by gpa

```
----------menu------------
1. Add student.
2. Edit student by id.
3. Delete student by id.
4. Sort student by gpa.
5. Sort student by name.
6. Show student.
0. exit.
---------------------------
Please choose: 4
----------menu------------
1. Add student.
2. Edit student by id.
3. Delete student by id.
4. Sort student by gpa.
5. Sort student by name.
6. Show student.
0. exit.
---------------------------
Please choose: 6
    2 |          Ha Cong Tung |    21 |              Ha Nam |        6.5
    4 |      Nguyen Cong Vinh |    21 |             ung Yen |        7.0
    3 |         Hoang Thi Lan |    21 |            Bac Ninh |        7.5
    5 |        Cao Van Nguyen |    21 |            Nam Dinh |        8.0
    1 |        Pham Thanh Nam |    22 |           Thai Binh |        8.5
```

- Sort student by name

```
----------menu------------
1. Add student.
2. Edit student by id.
3. Delete student by id.
4. Sort student by gpa.
5. Sort student by name.
6. Show student.
0. exit.
--------------------------
Please choose: 5
----------menu------------
1. Add student.
2. Edit student by id.
3. Delete student by id.
4. Sort student by gpa.
5. Sort student by name.
6. Show student.
0. exit.
--------------------------
Please choose: 6
    5 |       Cao Van Nguyen |    21 |             Nam Dinh |        8.0
    2 |         Ha Cong Tung |    21 |               Ha Nam |        6.5
    3 |        Hoang Thi Lan |    21 |             Bac Ninh |        7.5
    4 |     Nguyen Cong Vinh |    21 |              ung Yen |        7.0
    1 |       Pham Thanh Nam |    22 |            Thai Binh |        8.5
```