

TRƯỜNG ĐẠI HỌC MỞ HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN
MÔN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

NHÓM 12

Nguyễn Công Minh-1910A03
Trần Văn Ngọc Bảo-1910A05
Quang Thị Thúy Hằng-1910A04
Lê Công Tuấn-1910A03

Giảng viên hướng dẫn: Ths.Trịnh Thị Xuân

Hà Nội – 2020

MỤC LỤC

Phân công:	3
Phát biểu bài toán:	3
Các lớp và đối tượng cần quản lí:	4
Sơ đồ:.....	5
Code:.....	6

PHÂN CÔNG VÀ KẾ HOẠCH THỰC HIỆN

HỌ VÀ TÊN	CÔNG VIỆC	ĐÁNH GIÁ KẾT QUẢ	GHI CHÚ
Quang Thị Thúy Hằng	Phát biểu bài toán, viết chương trình	Hoàn thành	
Trần Văn Ngọc Bảo	Thiết kế sơ đồ lớp, viết chương trình	Hoàn thành	
Nguyễn Công Minh	Thiết kế sơ đồ lớp, viết chương trình, kiểm thử	Hoàn thành	
Lê Công Tuấn	Mô tả thuật toán thao tác, viết chương trình	Hoàn thành	

ĐỀ TÀI BÀI TẬP LỚN : Quản lý điểm của Sinh Viên Đại Học

PHÁT BIỂU BÀI TOÁN:

Quản lý điểm sinh viên đại học đáp ứng những nhu cầu cần thiết như: thêm điểm của từng sinh viên vào hệ thống, cũng có thể tự động thống kê điểm tổng cuối kì, sửa điểm khi cần thiết để điều chỉnh và theo dõi cũng như nắm bắt tình hình thực tế của sinh viên, có thể sao lưu nhằm tránh việc mất dữ liệu không mong muốn, không cần sổ sách tốn diện tích và thời gian xử lý tiết kiệm hơn, tiện cho việc theo dõi của sinh viên để biết tình hình học tập của mình, thao tác đơn giản không rườm rà phức tạp. Cho phép quản lý mọi nơi thông qua hình thức truy cập Internet giáo viên có thể quản lý điểm của sinh viên. Chính vì thế hệ thống quản lý điểm sinh viên đã ra đời để phục vụ những yêu cầu cần thiết cũng như thay đổi hiện đại theo xu hướng.

I. Các Lớp Đối Tượng Cần Quản Lý.

A. PERSON

1. Thuộc Tính

- Tên
- Tuổi
- Địa chỉ

1. Phương Thức

- Khởi Tạo mặc định
- get, set của các thuộc tính
- Nhập, Hiện PERSON

B. Sinh Viên

1. Thuộc tính: kế thừa PERSON

- Mã Sinh Viên
- Ngành học
- Điểm các môn

1. Phương Thức

- Khởi Tạo
- nhập, xuất
- get, set,

Điểm

1. Thuộc tính

- Điểm giải tích
- Điểm csdl
- Điểm lthdt
- Điểm tiếng anh
- Điểm ktmt
- Điểm chuyên cần
-

1. Phương Thức

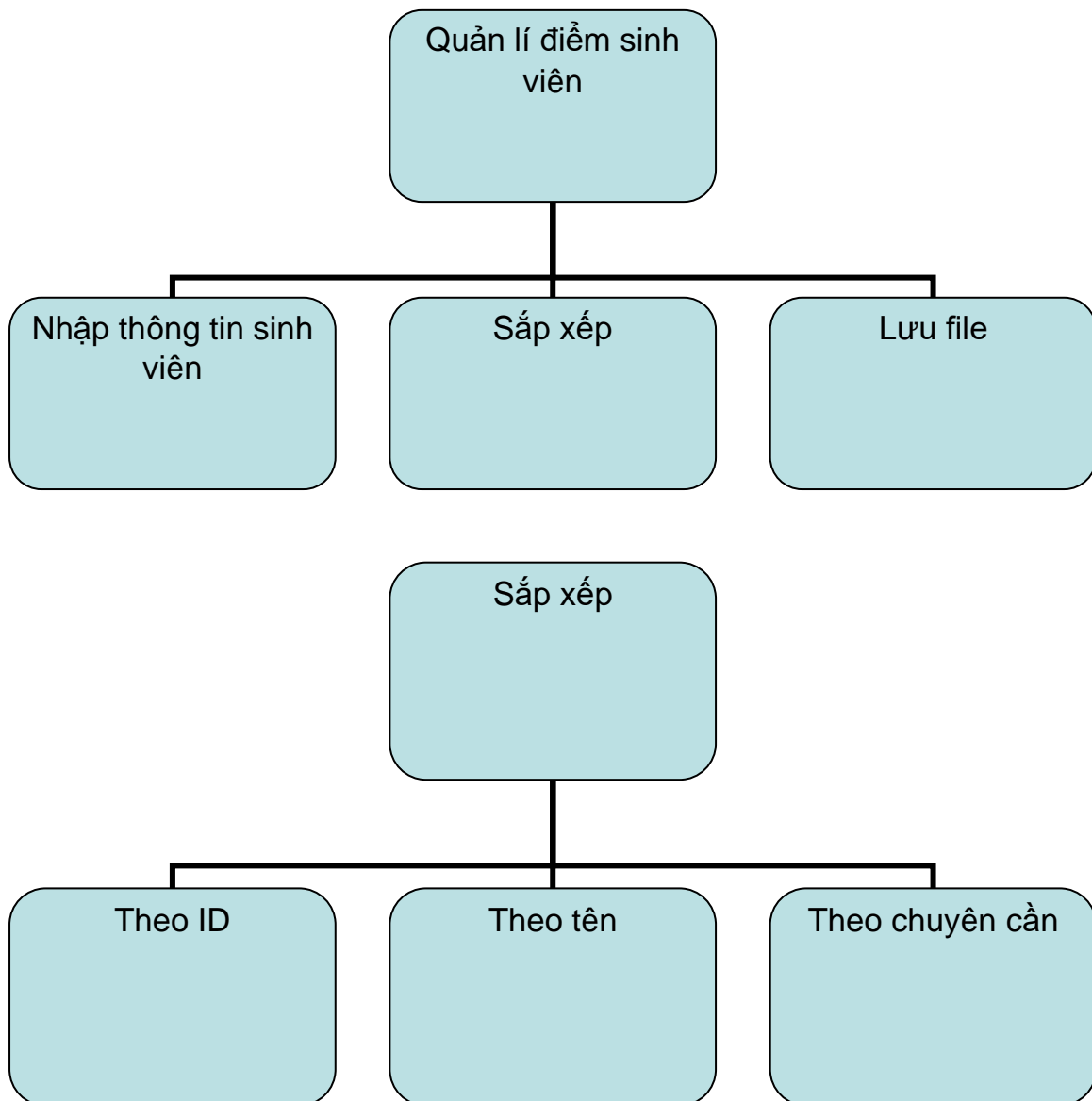
- Khởi tạo
- Nhập xuất
- Set, get

C. MAIN

1. Thuộc tính

- Menu
- 1. Phương Thức
 - Khởi Tạo
 - get, set cho các thuộc tính.
 - Thêm sinh viên
 - Hiển thị sinh viên
 - Sắp xếp
 - Xóa sinh viên
 - Đọc file
 - Ghi file

II.SƠ ĐỒ



III.CODE

Main

```
package btldone;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;
import java.util.Scanner;

public class main {
    static List<Student> studentList = new ArrayList<>();
    static Scanner input = new Scanner(System.in);

    public static void main(String[] args) {
        int choose;

        do {
            showMenu();
            choose = Integer.parseInt(input.nextLine());
            switch (choose) {
                case 1:
                    input();
                    ghiFile();
                    break;
                case 2:
                    editById();
                    break;
                case 3:
                    deleteById();
                    break;
                case 4:
                    sortByGpa();
                    break;
                case 5:
                    sortByName();
                    break;
                case 6:
                    showAll();
                    break;
            }
        }
    }
}
```

```

        case 7:
            docFile();
            break;
        default:
            System.out.println("EXIT!");
            break;
    }
} while (choose != 9);
}

static void input() {
    int n, i;
    Scanner sc = new Scanner(System.in);

    do {
        System.out.print("\nNHẬP SỐ SINH VIÊN: ");
        n = sc.nextInt();
    } while (n <= 0);
    studentList = new ArrayList<>(n);
    for (i = 0; i < n; i++) {
        Student x = new Student();
        System.out.printf("NHẬP SINH VIÊN THỨ %d: \n", i + 1);
        x.input();
        studentList.add(x);
    }
}

static void editById() {
    System.out.println("Nhập ID cần sửa : ");
    int id = Integer.parseInt(input.nextLine());

    for (Student student : studentList) {
        if (student.getId() == id) {
            //sua
            student.nhap();
            break;
        }
    }
}

static void deleteById() {
    System.out.println("Nhập ID cần xóa : ");
    int id = Integer.parseInt(input.nextLine());

    for (Student student : studentList) {
        if (student.getId() == id) {
            //sua
            studentList.remove(student);
            break;
        }
    }
}

```

```

    }
}

static void sortByGpa() {
    Collections.sort(studentList, new Comparator<Student>() {
        @Override
        public int compare(Student o1, Student o2) {
            return (o1.getCc() > o2.getCc()) ? 1 : -1;
        }
    });
}

static void sortByName() {
    Collections.sort(studentList, new Comparator<Student>() {
        @Override
        public int compare(Student o1, Student o2) {
            return o1.getFullname().compareTo(o2.getFullname());
        }
    });
}

static void showAll() {
    for (Student student : studentList) {
        student.display();
    }
}

static void ghiFile() {
    try {
        FileOutputStream fileOut = new FileOutputStream("input.txt");
        ObjectOutputStream objOut = new ObjectOutputStream(fileOut);
        objOut.flush();
        objOut.writeObject(studentList);
        objOut.close();
        fileOut.close();
    }
    catch (Exception ex) {
        ex.printStackTrace();
    }
}

static void docFile() {
    try {
        studentList = new ArrayList<Student>();
        FileInputStream fin = new FileInputStream("input.txt");
        ObjectInputStream fout = new ObjectInputStream(fin);
        studentList = (ArrayList)fout.readObject(); for(Student x:studentList) {
            x.display();
        }
        fin.close();
    }
}

```



```

        fout.close();

    }
    catch(FileNotFoundException e) {
        System.out.print("\nKHÔNG THẤY file.");
    }
    catch(Exception ex) {
        ex.printStackTrace();
    }
}
static void showMenu() {
    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 list by CC");
    System.out.println("5. Sort student list by Name");
    System.out.println("6. Show All Stundent");
    System.out.println("7. Read file input.txt and show all");
    System.out.println("8. Exit");
    System.out.println("Choose : ");
}
}

```

STUDENT

```
package btldone;
```

```

import java.util.Scanner;
public class Student extends Person {
    String fullname;
    String address;
    int age;
    int id;
    String faculty ;
    float cc;
    float giaitich;
    float lthdt;
    float tienganh;
    float ktmt;
    float csdl;
    public Student() {
        super();
        fullname = "";
        address = "";
        age =0;
        id =0;
        faculty = "";
        cc =0;
        giaitich =0;
        lthdt =0;
    }
}

```

```

        tienganh =0;
        ktmt =0;
        csdl =0;
    }

    public Student(String hoTen, String address, int age, int id, String faculty, float cc,
        float giaitich, float lthdt, float tienganh,
        float ktmt, float csdl) {
        this.fullname = hoTen;
        this.address= address;
        this.age= age;
        this.id = id;
        this.faculty = faculty;
        this.cc = cc;
        this.giaitich = giaitich;
        this.lthdt = lthdt;
        this.tienganh = tienganh;
        this.ktmt = ktmt;
        this.csdl = csdl;
    }

    public String getFullName() {
        return fullname;
    }

    public String getAddress() {
        return address;
    }

    public int getAge() {
        return age;
    }

    public int getId() {
        return id;
    }

    public float getCc() {
        return cc;
    }

    public float getGiaitich() {
        return giaitich;
    }

    public float getLthdt() {
        return lthdt;
    }

```

```

public float getTienganh() {
    return tienganh;
}

public float getKtmt() {
    return ktmt;
}

public float getCSDL() {
    return csdl;
}

public String getFaculty() {
    return faculty;
}

public void setFullname(String fullname) {
    this.fullname = fullname;
}

public void setAddress(String address) {
    this.address = address;
}

public void setAge(int age) {
    this.age = age;
}

public void setId(int id) {
    this.id = id;
}

public void setCc(float gpa) {
    this.cc = cc;
}

public void setGiatich(float giatich) {
    this.giatich = giatich;
}

public void setLthdt(float lthdt) {
    this.lthdt = lthdt;
}

public void setTienganh(float tienganh) {
    this.tienganh = tienganh;
}

public void setKtmt(float ktmt) {

```

```

        this.ktmt = ktmt;
    }

    public void setCSDL(float csdl) {
        this.csdl = csdl;
    }

    public void setFaculty(String faculty) {
        this.faculty = faculty;
    }
    @Override
    public String toString() {
        return "Student{" + "fullname=" + fullname + ", id=" + id + ", faculty="
            + faculty + ", address="
            + address + ", age=" + age + ", giải tích=" + giaitich + ", "
            + "lthdt=" + lthdt + ", tiếng anh=" + tienganh + ", ktmt=" + ktmt +
            ", csdl=" + csdl + ", cc=" + cc + '}';
    }
    {
    }
    public void display() {
        System.out.println(toString());
    }
    public void input() {
        Scanner input = new Scanner(System.in);

        System.out.println("Nhập ID : ");
        id = Integer.parseInt(input.nextLine());

        nhap();
    }
    public void nhap() {
        Scanner sc = new Scanner(System.in);
        System.out.print("NHẬP HỌ TÊN: ");
        fullname = sc.nextLine();
        System.out.print("NHẬP ĐỊA CHỈ: ");
        address = sc.nextLine();
        System.out.print("NHẬP TUỔI: ");
        age = Integer.parseInt(sc.nextLine());
        System.out.print("NHẬP TÊN KHOA: ");
        faculty = sc.nextLine();
        System.out.print("NHẬP ĐIỂM CHUYÊN CẦN: ");
        cc = Float.parseFloat(sc.nextLine());
        System.out.print("NHẬP ĐIỂM GIẢI TÍCH: ");
        giaitich = Float.parseFloat(sc.nextLine());
        System.out.print("NHẬP ĐIỂM LTHDT: ");
        lthdt = Float.parseFloat(sc.nextLine());
        System.out.print("NHẬP ĐIỂM TIẾNG ANH: ");
        tienganh = Float.parseFloat(sc.nextLine());
    }

```

```

        System.out.print("NHẬP ĐIỂM KTMT: ");
        ktmt = Float.parseFloat(sc.nextLine());
        System.out.print("NHẬP ĐIỂM CSDL: ");
        csdl = Float.parseFloat(sc.nextLine());
    }
}

```

PERSON

```

package btldone;

import java.io.Serializable;

public class Person implements Serializable {
    static String fullname;
    static String address;
    static int age;
    public Person() {
        super();
        fullname = "";
        address = "";
        age = 0;
    }
    public Person(String hoTen, String gioiTinh, int age) {
        this.fullname = hoTen;
        this.address = gioiTinh;
        this.age = age;
    }
    public String getFullName() {
        return fullname;
    }

    public String getAddress() {
        return address;
    }

    public int getAge() {
        return age;
    }

    public void setFullName(String fullname) {
        this.fullname = fullname;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public void setAge(int age) {
        this.age = age;
    }
}

```

```
}  
  
@Override  
public String toString() {  
    return "Student{" + "fullname=" + fullname + ", address=" +  
        + address + ", age=" + age + '}';  
}  
}
```