Homework Hustlers: https://discord.gg/aJ55rZBV

- Wizard.

Create a class Student with private fields id, name, age, and grade(A, B, C, D, E, F). Provide getter and setter methods to access and modify these fields.

```
~

public class main {
    public static void main(String[] args) {
        Student obj1 = new Student(1, "someone", 20, 'A');
        System.out.println(obj1.getName() + obj1.getGrade());
    }
}

class Student {
    private int id, age;
    private String name;
    private char grade;

    public Student(int id, String name, int age, char grade) {
        this.id = id;
        this.name = name;
        this.age = age;
        this.grade = grade;
    }

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

    public char getGrade() { return grade; }
    public void setGrade(char grade) { this.grade = grade; }
}
```

```
~

[wizard@archlinux w3]$ java main
someone A
[wizard@archlinux w3]$
```

Create a class Car with private fields model, price and fuelLevel. Provide getter and setter methods for model and price, but ensure that the fuelLevel field is read-only

```
~

class Car {
    private String model;
    private double price;
    private int fuelLevel;

    public Car(String model, double price, int fuelLevel) {
        this.model = model;
        this.price = price;
        this.fuelLevel = fuelLevel;
    }

    public String getModel() { return model; }
    public void setModel(String model) { this.model = model; }

    public double getPrice() { return price; }
    public void setPrice(double price) { this.price = price; }

    public int getFuelLevel() { return fuelLevel; }
}

public class main {
    public static void main(String[] args) {
        Car car = new Car("Some car", 50.00, 50);
        System.out.println(String.format("Car: %s, Price: %f, fuel:
%d",car.getModel(),car.getPrice(),car.getFuelLevel()));
    }
}
```

```
~

[wizard@archlinux w3]$ java main
Car: Some car, Price: 50.000000, fuel: 50
[wizard@archlinux w3]$
```

Create an abstract class Vehicle with abstract methods startEngine() and stopEngine(). Then create two classes Car and Motorcycle that extend Vehicle and implement these methods differently.

```
~

abstract class Vehicle {
    abstract void startEngine();
    abstract void stopEngine();
}

class Car extends Vehicle {
    void startEngine() {
        System.out.println("Engine started");
    }
    void stopEngine() {
        System.out.println("Engine stopped");
    }
}

class Motorcycle extends Vehicle {
    void startEngine() {
        System.out.println("Engine started");
    }
    void stopEngine() {
        System.out.println("Engine stopped");
    }
}

public class main {
    public static void main(String[] args) {
        Motorcycle bike = new Motorcycle();
        Car car = new Car();
        car.startEngine();
        car.stopEngine();
        bike.startEngine();
        bike.stopEngine();

    }
}
```

```
~

[wizard@archlinux w3]$ java main
Engine started
Engine stopped
Engine started
Engine stopped
[wizard@archlinux w3]$
```

Create an abstract class GameCharacter with abstract methods like attack() and defend(). Then, create subclasses Warrior and Archer with different attack and defense behaviors

```
~

abstract class GameCharacter {
    abstract void attack();
    abstract void defend();
}

class Warrior extends GameCharacter {
    void attack() {
        System.out.println("Warrior attacks");
    }
    void defend() {
        System.out.println("Warrior defends");
    }
}

class Archer extends GameCharacter {
    void attack() {
        System.out.println("Archer attacking");
    }
    void defend() {
        System.out.println("Archer defends");
    }
}

public class main {
    public static void main(String[] args) {
        Archer  ar= new Archer();
        Warrior wa = new Warrior();
        ar.attack();
        ar.defend();
        wa.attack();
        wa.defend();
    }
}
```

```
~

[wizard@archlinux w3]$ java main
Archer attacking
Archer defends
Warrior attacks
Warrior defends
[wizard@archlinux w3]$
```

Create an interface PaymentMethod with a method processPayment(double amount). Implement it in classes Esewa and Khalti.

```
~

interface PaymentMethod {
    void processPayment(double amount);
}

class Esewa implements PaymentMethod{

    public void processPayment(double amount){
        System.out.println("Processed Payment of "+amount+" Via Esewa");
    }
}

class Khalti implements PaymentMethod{

    public void processPayment(double amount){
        System.out.println("Processed Payment of "+amount+" Via Khalti");
    }

}
public class main {
    public static void main(String[] args) {
        Esewa es = new Esewa();
        es.processPayment(100.00);

        Khalti kh = new Khalti();
        kh.processPayment(100.00);
    }
}
```

```
~

[wizard@archlinux w3]$ java main
Processed Payment of 100.0 Via Esewa
Processed Payment of 100.0 Via Khalti
[wizard@archlinux w3]$
```

Create an interface RemoteControl with methods powerOn() and powerOff(). Implement this interface in classes TV and AC, which turn on and off their respective devices.

~

```
interface RemoteControl{
    void powerOn();
    void powerOff();
}

class TV implements RemoteControl{
    public void powerOn(){
        System.out.println("Turned on TV");
    }
    public void powerOff(){
        System.out.println("Turned off TV");
    }
}

class AC implements RemoteControl{
    public void powerOn(){
        System.out.println("Turned on AC");
    }
    public void powerOff(){
        System.out.println("Turned off AC");
    }

}
public class main {
    public static void main(String[] args) {
        TV tv = new TV();
        AC ac = new AC();
        tv.powerOn();
        tv.powerOff();
        ac.powerOn();
        ac.powerOff();



    }

}
```

~

```
[wizard@archlinux w3]$ java main
Turned on TV
Turned off TV
Turned on AC
Turned off AC
[wizard@archlinux w3]$
```

Write a program to take the name of foods as inputs from the user and store them in a .txt file

```
~

import java.io.FileWriter;
import java.util.Scanner;

public class main {
    public static void main(String[] args) {
        try (Scanner sc = new Scanner(System.in); FileWriter writer = new FileWriter("foods.txt")) {
            System.out.print("Enter food names: ");
            String[] foods = sc.nextLine().split(",");
            for (String food : foods){
                writer.write(food + "\n");
            }
        } catch (Exception e) {
            System.out.println("Error");
        }
    }
}
```

```
~

[wizard@archlinux w3]$ java main
Enter food names: Apple,Banana,Something,Something-Else
[wizard@archlinux w3]$ cat foods.txt
Apple
Banana
Something
Something-Else
[wizard@archlinux w3]$
```

Create a class Student with private fields name, age, grade(A, B, C, D, E, F). Then, write a program that stores student information(id, name, age, grade) into a .csv file.

```
~

import java.io.FileWriter;
import java.util.Scanner;

class Student {
    private int id, age;
    private String name;
    private char grade;

    public Student(int id, String name, int age, char grade) {
        this.id = id;
        this.name = name;
        this.age = age;
        this.grade = grade;
    }

    public void write() {
        try (FileWriter writer = new FileWriter("students.csv", true)) {
            writer.write(String.format("%d,%s,%d,%c%n", id, name, age, grade));
        } catch (Exception e) {
            System.out.println("Error");
        }
    }
}

public class main {
    public static void main(String[] args) {
        try (Scanner sc = new Scanner(System.in)) {
            System.out.println("Enter students (format: id,name,age,grade-id,name,age,grade):");
            String[] students = sc.nextLine().split("-");

            for (String student : students) {
                String[] parts = student.split(",");
                int id = Integer.parseInt(parts[0]);
                String name = parts[1];
                int age = Integer.parseInt(parts[2]);
                char grade = parts[3].charAt(0);

                Student std = new Student(id, name, age, grade);
                std.write();
            }

            System.out.println("Student details saved to students.csv");
        } catch (Exception e) {
            System.out.println("Error " );
        }
    }
}
```

```
~

[wizard@archlinux w3]$ java main
Enter students (format: id,name,age,grade-id,name,age,grade):
10,someone,19,A-11,someoneelse,20,B
Student details saved to students.csv
[wizard@archlinux w3]$ cat students.csv
10,someone,19,A
11,someoneelse,20,B
[wizard@archlinux w3]$
```

Write a program that reads a list of students data from a csv file and stores them in a list. Then display the list of students according to their grade.

~

```java
import java.io.FileReader;
import java.util.ArrayList;
import java.util.Scanner;

class Student {
    private int id, age;
    private String name;
    private char grade;

    public Student(int id, String name, int age, char grade) {
        this.id = id;
        this.name = name;
        this.age = age;
        this.grade = grade;
    }

    public char getGrade() {
        return grade;
    }

    public String toString() {
        return String.format("ID: %d, Name: %s, Age: %d, Grade: %c", id, name, age, grade);
    }
}

public class main {
    public static void main(String[] args) {
        ArrayList<Student> students = new ArrayList<>();
        try (Scanner scanner = new Scanner(new FileReader("students.csv"))) {
            while (scanner.hasNextLine()) {
                String line = scanner.nextLine();
                String[] parts = line.split(",");
                int id = Integer.parseInt(parts[0]);
                String name = parts[1];
                int age = Integer.parseInt(parts[2]);
                char grade = parts[3].charAt(0);
                students.add(new Student(id, name, age, grade));
            }
        } catch (Exception e) {
            System.out.println("Error");
        }

        for (int i = 0; i < students.size() - 1; i++) {
            for (int j = 0; j < students.size() - 1 - i; j++) {
                if (students.get(j).getGrade() > students.get(j + 1).getGrade()) {
                    Student temp = students.get(j);
                    students.set(j, students.get(j + 1));
                    students.set(j + 1, temp);
                }
            }
        }

        for (Student s : students) {
            System.out.println(s);
        }
    }
}
```

```
~

[wizard@archlinux w3]$ java main
ID: 1, Name: Someone, Age: 12, Grade: A
ID: 10, Name: someone, Age: 19, Grade: A
ID: 11, Name: someoneelse, Age: 20, Grade: B
[wizard@archlinux w3]$
```