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

- Wizard.

Task 1:

Create a class BankAccount with private fields accountNumber and balance.

- Implement getter and setter methods to control access to these fields.
- Write a method to deposit and withdraw money from the account, ensuring that negative balances aren't allowed.

~

```
class BankAccount {
    private String accountNumber;
    private double balance;

    public BankAccount(String accountNumber, double balance) {
        this.accountNumber = accountNumber;
        if (balance >= 0) {
            this.balance = balance;
        } else {
            this.balance = 0;
        }
    }

    public String getAccountNumber() {
        return accountNumber;
    }

    public double getBalance() {
        return balance;
    }

    public void setBalance(double balance) {
        if (balance >= 0) {
            this.balance = balance;
        }
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
        }
    }

    public boolean withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            return true;
        }
        return false;
    }
}
```

```
public class main {
    public static void main(String[] args) {
        BankAccount account = new BankAccount("123456789", 500.00);
        System.out.println("account number: " + account.getAccountNumber());
        System.out.println("balance: " + account.getBalance());
        account.deposit(200);
        System.out.println("balance: " + account.getBalance());
        if (account.withdraw(100)) {
            System.out.println("successful, balance: " + account.getBalance());
        } else {
            System.out.println("withdrawal failed, insufficient ");
        }

        if (account.withdraw(700)) {
            System.out.println("successful, Balance: " + account.getBalance());
        } else {
            System.out.println("withdrawal failed, insufficient .");
        }

        account.setBalance(1000);
```

```
        System.out.println("balance: " + account.getBalance());
    }
}
```

```
~

[wizard@archlinux tuto1]$ java main
account number: 123456789
balance: 500.0
balance: 700.0
successful, balance: 600.0
withdrawal failed, insufficient .
balance: 1000.0
[wizard@archlinux tuto1]$
```

Task 2

Create an abstract class Employee with an abstract method calculateSalary() and a non-abstract method getDetails().

```
~

abstract class Employee {
    abstract void calculateSalary();
    void getDetails() {
        System.out.println("Some Details");
    }
}
```

Create two subclasses: FullTimeEmployee and PartTimeEmployee.

```
~

class FullTimeEmployee extends Employee{

}

class PartTimeEmployee extends Employee{

}
```

The FullTimeEmployee class should calculate salary based on a fixed monthly salary, while PartTimeEmployee calculates salary based on hourly wage and hours worked.

```
~

abstract class Employee {
    abstract void calculateSalary();
    void getDetails() {
        System.out.println("Some Details");
    }
}

class FullTimeEmployee extends Employee{

    public void calculateSalary(){
        System.out.println(3000 * 30);
    }
}

class PartTimeEmployee extends Employee{
    public void calculateSalary(int hRate, int hWorked){
        System.out.println(hRate * hWorked);
    }
}

public class main {

    public static void main(String[] args) {
        FullTimeEmployee obj1 = new FullTimeEmployee();
        obj1.calculateSalary();

        PartTimeEmployee obj2 = new PartTimeEmployee();
        obj2.calculateSalary(50,30);

    }
}
```

```
~

[wizard@archlinux tuto1]$ java main
90000
1500
[wizard@archlinux tuto1]$
```

Task 3:

Design an abstract class Vehicle with abstract methods fuelEfficiency() and topSpeed()

```
~

abstract class Vehicle {
    abstract void fuelEffeciency();
    abstract void topSpeed();
}
```

Create subclass Car and Bike, each providing its own calculation for fuel efficiency and top speed

```
~

abstract class Vehicle {
    abstract void fuelEffeciency();
    abstract void topSpeed();
}


class Car extends Vehicle{
    public void fuelEffeciency(){
        System.out.println("Some value for car");
    }

    public void topSpeed(){
        System.out.println("Some Top Speed");
    }
}

class Bike extends Vehicle{
    public void fuelEffeciency(){
        System.out.println("Some value for Bike");
    }

    public void topSpeed(){
        System.out.println("Some Top Speed");
    }

}

public class main {

    public static void main(String[] args) {
        Car obj1 = new Car();
        Bike obj2 = new Bike();
        obj1.fuelEffeciency();
        obj2.fuelEffeciency();

    }
}
```

```
~

[wizard@archlinux tuto1]$ java main
Some value for car
Some value for Bike
[wizard@archlinux tuto1]$
```

Task 4:

Define an interface Shape with methods calculateArea() and calculatePerimeter().

```
~

interface Shape {
    abstract void calculateArea();
    abstract void calculatePerimeter();
}
```

Implement this interface in classes Circle and Rectangle with appropriate calculations.

```
~

interface Shape {
    void calculateArea(int radius);
    void calculatePerimeter(int radius);

    void calculateArea(int length, int breadth);
    void calculatePerimeter(int length, int breadth);
}

class Circle implements Shape{
    public static final double PI = 3.141592653;

    public void calculateArea(int radius) {
        System.out.println("Area of Circle: " + PI * radius * radius);
    }

    public void calculatePerimeter(int radius) {
        System.out.println("Perimeter of Circle: " + 2 * PI * radius);
    }

    public void calculatePerimeter(int l,int b){}
    public void calculateArea(int l,int b){}

}

class Rectangle implements Shape{
    public void calculateArea(int length, int breadth) {
        System.out.println("Area of Rectangle: " + length * breadth);
    }

    public void calculatePerimeter(int length, int breadth) {
        System.out.println("Perimeter of Rectangle: " + 2 * (length + breadth));
    }
    public void calculatePerimeter(int r){}
    public void calculateArea(int r){}
}

public class main {
    public static void main(String[] args) {
        Circle obj1 = new Circle();
        Rectangle obj2 = new Rectangle();

        obj1.calculatePerimeter(5);
        obj1.calculateArea(5);

        obj2.calculatePerimeter(5, 5);
        obj2.calculateArea(5, 5);
    }
}
```

```
~

[wizard@archlinux tuto1]$ java main
Perimeter of Circle: 31.41592653
Area of Circle: 78.539816325
Perimeter of Rectangle: 20
Area of Rectangle: 25
[wizard@archlinux tuto1]$
```

Task 5

Create an interface Drivable with methods start(), accelerate(), and brake().

```
~

interface Drivable {
    void start();
    void accelerate();
    void brake();
}
```

Implement this interface in the classes Car and Truck.

```
class  Truck implements Drivable{

    public void start(){
        System.out.println("Truck Started.");
    }
    public void accelerate(){
        System.out.println("Truck accelerated");
    }
    public void brake(){
        System.out.println("Truck braked");
    }

}

class Car implements Drivable{
    public void start(){
        System.out.println("Car Started.");
    }
    public void accelerate(){
        System.out.println("Car accelerated");
    }

    public void brake(){
        System.out.println("Car braked");
    }

}
public class main {

    public static void main(String[] args) {
        Car obj1 = new Car();
        Truck obj2 = new Truck();
        obj1.start();
        obj1.accelerate();
        obj2.start();
        obj2.accelerate();
    }
}
```

```
~

[wizard@archlinux tuto1]$ java main
Car Started.
Car accelerated
Truck Started.
Truck accelerated
[wizard@archlinux tuto1]$
```

Task 6

Write a regular expression to valid email address and password.

```
~

public class main {
    public static void main(String[] args) {
        String email = "someemail@gmail.com";
        String password = "somepassword123";
        String emailRegex = "^[a-zA-Z0-9._]+@gmail\\.com$";
        String passwordRegex = "^[a-zA-Z0-9]{6,}$";
        if (password.matches(passwordRegex)){
            System.out.println("Valid password!");
        }else{
            System.out.println("InValid password!");
        }

        if (email.matches(emailRegex)){
            System.out.println("Valid email!");
        }else{
            System.out.println("InValid email!");
        }

    }
}
```

```
~ For : "someemail@gmail.com", "somepassword123"

[wizard@archlinux tuto1]$ java main
Valid password!
Valid email!
[wizard@archlinux tuto1]$

~ For: "~~@gmail.com", "123";
[wizard@archlinux tuto1]$ java main
InValid password!
InValid email!
[wizard@archlinux tuto1]$
```

File Handling:

Task 7

Create a file named “myFile.txt” and write the text “Java is a high level programming language”.

```
~

import java.io.*;
public class main {
    public static void main(String[] args) {
        try (FileWriter writer = new FileWriter("myFile.txt")) {
            writer.write("Java is a high level programming language");
        }catch (Exception e){
            System.out.println("Error"+e);
        }
    }
}
```

```
~

[wizard@archlinux tuto1]$ cat myFile.txt
Java is a high level programming language
[wizard@archlinux tuto1]$
```

Task 8

Write a java program to read the text from the above file named “myFile.txt”.

```
~

import java.io.*;

public class main {
    public static void main(String[] args) {
        try (FileReader reader = new FileReader("myFile.txt")) {
            for (int ch; (ch = reader.read()) != -1; ) {
                System.out.print((char) ch);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
~

[wizard@archlinux tuto1]$ java main
Java is a high level programming language
[wizard@archlinux tuto1]$ Z
```

Task 9:

Write the program to append the text into the existing text file.

```
~

import java.io.*;
public class main {
    public static void main(String[] args) {
        try (FileWriter writer = new FileWriter("myFile.txt",true)) {
            writer.write("\n Some Appended text");
        }catch (Exception e){
            System.out.println("Error"+e);
        }
    }
}
```

```
~

[wizard@archlinux tuto1]$ java main && cat myFile.txt
Java is a high level programming language
Some Appended text
[wizard@archlinux tuto1]$
```

Task 10:

Write the program to delete the existing text file.

~

```java
import java.io.*;

public class main{
    public static void main(String[] args) {
        File file = new File("myFile.txt");
        if (new File("myFile.txt").delete()) {
            System.out.println("File deleted successfully.");
        } else {
            System.out.println("Failed to delete the file.");
        }
    }
}
```

~

```
[wizard@archlinux tuto1]$ java main
File deleted successfully.
[wizard@archlinux tuto1]$ ls -a
.   BankAccount.class  Car.class     Drivable.class  FullTimeEmployee.class  main.java             Rectangle.class  thing.md
Vehicle.class
..  Bike.class         Circle.class  Employee.class  main.class              PartTimeEmployee.class  Shape.class
Truck.class
[wizard@archlinux tuto1]$
```