

# Lista 03

João Vitor Espig

2 de junho de 2024

Usando L<sup>A</sup>T<sub>E</sub>X

## Exercício 1

A interface **IVehicle** define um mecanismo para mover um veículo. A classe **Airplane** implementa a interface **IVehicle** e sobreescreve o método **move()** da interface.

---

```
// Main.java
package src;

interface IVehicle {
    void move();
}

class Airplane implements IVehicle {
    @Override
    public void move() {
        System.out.println("Zoom");
    }
}

public class Main {
    public static void main(String[] args) {
        System.out.println("Hello World");
        Airplane c = new Airplane();
        c.move();
    }
}
```

---

## Exercício 2

---

```
// Main.java
import java.util.*;

class Password {
    Scanner in = new Scanner(System.in);
    String password = "";
    public int attempts = 0;
    public boolean blocked = false;

    public Password(String password) {
        this.password = password;
    }

    public boolean entraSenha() {
        if (blocked) {
            System.out.println("Password blocked!");
            return false;
        }

        boolean correct = inputPassword(in);
        if (correct) {
            attempts = 0;
            return true;
        } else {
            attempts++;
            block();
            if (!blocked)
                System.out.println("Wrong password");
        }

        return entraSenha();
    }

    public void trocarSenha() {
        if (!entraSenha()){
            return;
        }

        System.out.println("Type your new password");
        System.out.print("> ");
        String newPassword = in.nextLine();
    }
}
```

```

        password = newPassword;
    }

    boolean inputPassword(Scanner in) {
        System.out.println("Type your password");
        System.out.print("> ");
        String password = in.nextLine();
        return this.password.equals(password);
    }

    void block() {
        if (attempts >= 3) {
            blocked = true;
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Password s = new Password("UEPA1234");
        s.trocarSenha();
        System.out.println(s.entraSenha());
    }
}

```

---

## Exercício 3

---

```

// Main.java
class Port {
    private boolean isOpen;
    public String color;
    public Float posX;
    public Float posY;
    public Float posZ;

    public void open() {
        this.isOpen = true;
    }

    public void close() {
        this.isOpen = false;
    }
}

```

```

    }

    public void paint(String color) {
        this.color = color;
    }

    public boolean isOpen() {
        return this.isOpen;
    }
}

public class Main {
    public static void main(String[] args) {
        Port port = new Port();
        port.open();
        port.paint("red");
        System.out.println(port.isOpen());
    }
}

```

---

## Exercício 4

---

```

// Main.java
class Port {
    private boolean isOpen;
    public String color;
    public Float posX;
    public Float posY;
    public Float posZ;

    public Port(String color, Float posX, Float posY, Float posZ) {
        this.isOpen = true;
        this.color = color;
        this.posX = posX;
        this.posY = posY;
        this.posZ = posZ;
    }

    public void open() {
        this.isOpen = true;
    }
}

```

```

    public void close() {
        this.isOpen = false;
    }

    public void paint(String color) {
        this.color = color;
    }

    public boolean isOpen() {
        return this.isOpen;
    }
}

class House {
    Port port1;
    Port port2;
    Port port3;
    String color;

    public House() {
        port1 = new Port("", 0.0f, 0.0f, 0.0f);
        port2 = new Port("", 0.0f, 0.0f, 0.0f);
        port3 = new Port("", 0.0f, 0.0f, 0.0f);
        color = "";
    }

    public void paint(String color) {
        this.color = color;
    }

    public String getColor() {
        return this.color;
    }

    public int getAmountOfOpenPorts() {
        int amount = 0;
        if (port1.isOpen()) {
            amount++;
        }
        if (port2.isOpen()) {
            amount++;
        }
    }
}

```

```

        if (port3.isOpen()) {
            amount++;
        }
        return amount;
    }

    public Port getPort1() {
        return port1;
    }

    public Port getPort2() {
        return port2;
    }

    public Port getPort3() {
        return port3;
    }
}

class TestHouse {
    House house;

    public TestHouse() {
        house = new House();
    }

    public void testHouse() {
        house.paint("red");
        house.getPort1().open();
        house.getPort2().open();
        house.getPort3().open();
        System.out.println(house.getAmountOfOpenPorts());
        System.out.println(house.getColor());
        assert house.getAmountOfOpenPorts() == 3 : "A casa deve ter
            3 portas abertas";
        assert house.getColor().equals("red") : "A casa deve ter a
            cor vermelha";
    }
}

public class Main {
    public static void main(String[] args) {
        TestHouse testHouse = new TestHouse();
    }
}

```

```
        testHouse.testHouse();
    }
}
```

---

## Exercício 5

---

// Main.java

```
class Calculadora {
    public float somar(float a, float b) {
        return a + b;
    }
    public float subtrair(float a, float b) {
        return a - b;
    }
    public float dividir(float a, float b) {
        return a / b;
    }
    public float multiplicar(float a, float b) {
        return a * b;
    }
}

class CalculadoraCientifica extends Calculadora {
    public float potencia(float a, float b) {
        return (float) Math.pow(a, b);
    }
}

class TestaCalculadoras {
    public void testaCalculadoraNormal() {
        Calculadora calculadora = new Calculadora();
        float resultado = calculadora.somar(1.0f, 2.0f);
        System.out.println("O resultado da soma de 1.0 e 2.0 e: " +
            resultado);
        assert resultado == 3.0f;
        resultado = calculadora.subtrair(1.0f, 2.0f);
        System.out.println("O resultado da subtracao de 1.0 e 2.0 e:
            " + resultado);
        assert resultado == -1.0f;
    }
}
```

```

        resultado = calculadora.dividir(1.0f, 2.0f);
        System.out.println("O resultado da divisao de 1.0 e 2.0 e: "
            + resultado);
        assert resultado == 0.5f;
        resultado = calculadora.multiplicar(1.0f, 2.0f);
        System.out.println("O resultado da multiplicacao de 1.0 e
            2.0 e: " + resultado);
        assert resultado == 2.0f;
    }

    public void testaCalculadoraCientifica() {
        CalculadoraCientifica calculadora = new
            CalculadoraCientifica();
        float resultado = calculadora.potencia(2.0f, 3.0f);
        System.out.println("O resultado da potencia e: " +
            resultado);
        assert resultado == 8.0f;
    }
}

public class Main {
    public static void main(String[] args) {
        TestaCalculadoras test = new TestaCalculadoras();
        test.testaCalculadoraNormal();
        test.testaCalculadoraCientifica();
    }
}

```

---

## Exercício 6

---

```

// Main.java
import java.time.LocalDate;

class Data {
    int dia;
    int mes;
    int ano;

    public Data() {
        LocalDate data = LocalDate.now();
        this.dia = data.getDayOfMonth();
    }
}

```



```

        this.mes = data.getMonthValue();
        this.ano = data.getYear();
    }

    public Data(int dia, int mes, int ano) {
        this.dia = dia;
        this.mes = mes;
        this.ano = ano;
        if (!isValid()) {
            throw new IllegalArgumentException("Data invalida");
        }
    }

    public boolean isValid() {
        try {
            LocalDate.of(this.ano, this.mes, this.dia);
            return true;
        } catch (Exception e) {
            return false;
        }
    }

    public void proximoDia() {
        LocalDate data = LocalDate.of(this.ano, this.mes, this.dia);
        data = data.plusDays(1);
        this.dia = data.getDayOfMonth();
        this.mes = data.getMonthValue();
        this.ano = data.getYear();
    }

    public String toString() {
        return this.dia + "/" + this.mes + "/" + this.ano;
    }

    public void imprime() {
        System.out.println(this.toString());
    }
}

public class Main {
    public static void main(String[] args) {
        Data data = new Data();
        for (int i = 0; i < 10; i++) {

```

```

        data.proximoDia();
        data.imprime();
        try {
            Thread.sleep(100);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }

    // Teste de data invalida
    // Data dataErrada = new Data(30, 2, 2024);
    // dataErrada.imprime();
}
}

```

---

## Exercício 7

Respostas:

**a. O que é exibido na tela quando o código da esquerda é executado? Por quê?**

R.: Será exibido o valor 10, pois lamp2 é uma referência para lamp1 (logo iguais) e também porque i++ incrementa apenas após a chamada da função println.

**b. O que é exibido na tela quando o código da direita é executado? Por quê?**

R.: Não será exibido nenhum valor, pois as variáveis lamp1 e lamp2 são duas instâncias diferentes da classe Lampada, portanto não iguais.

**c. Qual é o valor de i ao final da execução do código da esquerda?**

R.: O valor de i após a execução do código é 11.

**d. Qual é o valor de i ao final da execução do código da direita?**

R.: 10

**e. O que acontece se acrescentarmos as duas linhas abaixo no código da esquerda? E no da direita? Por quê?**

**lamp1.trocarTipo("Halógena");**

**lamp2.mostrarInformacoesGerais();**

R.: No código da esquerda: Mostrará as informações atualizadas da lamp1. No código da direita: Mostrará as informações da lamp2 onde o seu tipo continua sendo "LED"