

Lista 3

Alunos: Estevão G. e Yean J.

1.

```
// Instrumento.java
public interface Instrumento {
    void tocar();
    void afinar();
}

// Guitarra.java
public class Guitarra implements Instrumento {
    @Override
    public void tocar() {
        System.out.println("Tocando a guitarra");
    }

    @Override
    public void afinar() {
        System.out.println("Afinando a guitarra");
    }
}

// Piano.java
public class Piano implements Instrumento {

    @Override
    public void tocar() {
        System.out.println("Tocando o piano");
    }

    @Override
    public void afinar() {
        System.out.println("Afinando o piano");
    }
}

// Main.java
public class Main {
    public static void main(String[] args) {
        Instrumento guitarra = new Guitarra();
        Instrumento piano = new Piano();

        guitarra.tocar();
        guitarra.afinar();

        piano.tocar();
        piano.afinar();
    }
}
```

```
}  
}
```

2.

```
// Senha.java  
public class Senha {  
    private String senha;  
    private int tentativas = 0;  
  
    public Senha(String senha){  
        this.senha = senha;  
    }  
  
    public void entraSenha(String senha) {  
        if (this.tentativas < 3) {  
            if (senha == this.senha){  
                this.tentativas = 0;  
                System.out.println("Senha correta.");  
            }  
            else {  
                this.tentativas++;  
                System.out.println("Senha incorreta. " + (3-this.tentativas) + "  
tentativas restantes.");  
            }  
        }  
        else {  
            System.out.println("Senha bloqueada.");  
        }  
    }  
  
    public void trocarSenha(String old_senha, String new_senha) {  
        if (this.tentativas ≥ 3) {  
            System.out.println("Senha bloqueada.");  
        }  
        else if (old_senha.equals(this.senha)) {  
            this.senha = new_senha;  
            System.out.println("Senha alterada.");  
        }  
        else {  
            this.tentativas++;  
            System.out.println("Senha antiga incorreta.");  
        }  
    }  
  
    public void printarSenha() {  
        System.out.println(this.senha);  
    }  
}
```

```
// Main.java
public class Main {
    public static void main(String[] args) {
        Senha senha = new Senha("abcde");
        senha.printarSenha();
        senha.entraSenha("bla");
        senha.entraSenha("bla");
        senha.entraSenha("bla");
        senha.entraSenha("bla");
        senha.entraSenha("abcde");
        senha.trocarSenha("bla", "ble");
        senha.trocarSenha("abcde", "12345");
        senha.printarSenha();
    }
}
```

3.

```
// Porta.java
package ex3;

public class Porta {
    private boolean aberta;
    private String cor;
    private double dimensaoX, dimensaoY, dimensaoZ;

    public Porta(boolean aberta, String cor, double dimensaoX, double dimensaoY,
double dimensaoZ) {
        if (dimensaoX ≤ 0 || dimensaoY ≤ 0 || dimensaoZ ≤ 0) {
            throw new IllegalArgumentException("As dimensões devem ser maiores
que zero.");
        }
        this.aberta = aberta;
        this.cor = cor;
        this.dimensaoX = dimensaoX;
        this.dimensaoY = dimensaoY;
        this.dimensaoZ = dimensaoZ;
    }

    @Override
    public String toString() {
        return "Porta{" +
            "aberta=" + aberta +
            ", cor='" + cor + '\'' +
            ", dimensaoX=" + dimensaoX +
            ", dimensaoY=" + dimensaoY +
            ", dimensaoZ=" + dimensaoZ +
            '}';
    }

    public void abre() {this.aberta = true;}
    public void fecha() {this.aberta = false;}
}
```

```
    public void pinta (String cor) {this.cor = cor;}  
    public boolean estaAberta() {return this.aberta;}  
}
```

4.

```
// Casa.java  
package ex4;  
import ex3.Porta;  
  
public class Casa {  
  
    private Porta porta1, porta2, porta3;  
    private String cor;  
  
    public Casa(Porta porta1, Porta porta2, Porta porta3, String cor) {  
        this.porta1 = porta1;  
        this.porta2 = porta2;  
        this.porta3 = porta3;  
        this.cor = cor;  
    }  
  
    @Override  
    public String toString() {  
        return "Casa{" +  
            "porta1=" + porta1 +  
            ", porta2=" + porta2 +  
            ", porta3=" + porta3 +  
            ", cor='" + cor + '\'' +  
            '}';  
    }  
  
    public Porta getPorta1() {  
        return porta1;  
    }  
  
    public Porta getPorta2() {  
        return porta2;  
    }  
  
    public Porta getPorta3() {  
        return porta3;  
    }  
  
    public String getCor() {  
        return cor;  
    }  
  
    public int quantasPortasEstaoAbertas() {  
        return ((porta1.estaAberta()?1:0) + (porta2.estaAberta()?1:0) +  
        (porta3.estaAberta()? 1: 0));  
    }  
}
```

```

    public void pinta (String cor)
    {
        this.cor = cor;
    }
}

// TestaCasa.java
package ex4;
import ex3.Porta;

public class TestaCasa {

    public static void main(String[] args) {
        Porta porta1 = new Porta(false, "azul", 1.3, 2, 3);
        Porta porta2 = new Porta(true, "vermelho", 1.3, 2, 3);
        Porta porta3 = new Porta(true, "verde", 1.3, 2, 3);

        Casa minha_casa = new Casa(porta1, porta2, porta3, "azul");
        System.out.println("Cor da casa: " + minha_casa.getCor());
        System.out.println("Quantidade de portas abertas: " +
minha_casa.quantasPortasEstaoAberta());
        minha_casa.pinta("amarelo");
        System.out.println("Cor da casa pintada: " + minha_casa.getCor());
        System.out.println(minha_casa.getPorta1().toString());
        System.out.println(minha_casa.getPorta2().toString());
        System.out.println(minha_casa.getPorta3().toString());
    }
}

```

5.

```

// Calculadora.java
package ex5;

public class Calculadora {
    public double multiplicar(double x, double y)
    {
        return x*y;
    }

    public double dividir(double x, double y) {
        try {
            return x/y;
        } catch (ArithmeticException e) {
            System.out.println("ERRO. Divisão por zero.");
            return Double.NaN;
        }
    }

    public double somar (double x, double y)
    {

```

```

        return x+y;
    }

    public double subtrair (double x, double y)
    {
        return x-y;
    }
}

// CalculadoraCientifica.java
package ex5;

public class CalculadoraCientifica extends Calculadora{
    double potencia(double x, double y)
    {
        if (x == 0 && y == 0)
            return Double.NaN;
        return Math.pow(x, y);
    }
}

// TestarCalculadora.java
package ex5;

public class TestarCalculadora {
    public static void main(String[] args) {
        Calculadora calc = new Calculadora();
        CalculadoraCientifica calccien = new CalculadoraCientifica();

        System.out.println("Resultado de somar(2, 3): " + calc.somar(2, 3));
        System.out.println("Resultado de dividir(0, 0): " + calc.dividir(0, 0));
        System.out.println("Resultado de multiplicar(13, 22): " +
calc.multiplicar(13, 22));
        System.out.println("Resultado de subtrair(11, 22): " + calc.subtrair(11,
22));
        System.out.println("Resultado de potencia(0, 0): " +
calccien.potencia(0, 0));
    }
}

```

6.

```

import java.time.LocalDate;

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

    public Data(int dia, int mes, int ano) {
        if (dataValida(dia, mes, ano)) {
            this.dia = dia;

```

```

        this.mes = mes;
        this.ano = ano;
    }
    else {
        System.out.println("Data inválida");
    }
}

public Data() {
    LocalDate dataAtual = LocalDate.now();
    this.dia = dataAtual.getDayOfMonth();
    this.mes = dataAtual.getMonthValue();
    this.ano = dataAtual.getYear();
}

// Método para ser utilizado na verificação da data
private boolean isAnoBissexto(int ano) {
    return (ano % 4 == 0 && ano % 100 != 0) || (ano % 400 == 0);
}

private boolean dataValida(int dia, int mes, int ano) {
    if (ano < 1 || mes < 1 || mes > 12 || dia < 1) {
        return false;
    }
    int[] diasNoMes = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

    // Verifica se é ano bissexto
    if (isAnoBissexto(ano) && mes == 2) {
        diasNoMes[1] = 29;
    }

    return dia <= diasNoMes[mes - 1];
}

public void imprimirData() {
    System.out.println(this.dia + "/" + this.mes + "/" + this.ano);
}

public void proximoDia() {
    if (dataValida(this.dia+1, this.mes, this.ano)) {
        this.dia++;
    }
    else if (dataValida(this.dia, this.mes+1, this.ano)) {
        this.mes++;
        this.dia = 1;
    }
    else {
        this.ano++;
        this.mes = 1;
        this.dia = 1;
    }
}
}

```

```
public static void main(String[] args) {  
    // Teste do construtor com parâmetros  
    Data data1 = new Data(31, 12, 1900);  
    data1.imprimirData(); // Output: 31/12/1900  
  
    // Teste do método proximoDia()  
    data1.proximoDia();  
    data1.imprimirData(); // Output: 1/1/1901  
  
    // Teste do construtor sem parâmetros  
    Data dataAtual = new Data();  
    dataAtual.imprimirData();  
}
```

7.

8. A condição é verdadeira porque as variáveis lamp1 e lamp2 apontam para o mesmo objeto na memória, será exibido o 10, porque o acréscimo só é feito após a execução da linha.
9. Nada, pois as duas lâmpadas, por mais que tenham o mesmo estado das variáveis, são dois objetos separados na memória, portanto o código dentro do if não será executado.

10. 11

11. 10

12. No código da esquerda o tipo das duas lâmpadas será modificado portanto a execução da segunda linha mostrará o tipo da lâmpada como "Halógena". No código da direita a primeira linha trocará o tipo apenas de lamp1 e nas informações gerais da lamp2 o tipo permanecerá LED.