



---

## Exercícios usando Exceções com Java

**From:** "Lucas Idm" <lucasldm2444@gmx.com>

**To:** josivaldo@gmx.com

**Date:** Mar 24, 2023 6:06:35 PM

---

**Integrantes:** Lucas Lucena, Gustavo Gouveia e Lucas Andrade

**1. O B.uni, banco da Uninassau quer que você faça um programa para ler os dados de uma conta bancária e depois realizar um saque nesta conta bancária, mostrando o novo saldo. Um saque não pode ocorrer se não houver saldo na conta, ou se o valor do saque for superior ao limite de saque da conta.**

```
import java.util.InputMismatchException;
import java.util.Scanner;

public class ContaBancaria {

    public static float saldo;
    public static float saque;

    public static void sacarDinheiro(float saq, float sal){

        if(sal > 0 && saq < sal){
            saldo -= saque;
            System.out.println("Sacou R$" + String.format("%.2f", saque));
            System.out.println("Seu saldo atual é de R$" + String.format("%.2f", saldo));
        }

        if(sal == 0) {
            System.out.println("Saldo zerado. Não pode sacar.");
        }

        if(saq > sal){
            System.out.println("Não pode sacar mais que o limite do saldo.");
        }
    }

    public static void main(String[] args) {

        try{

            Scanner sc = new Scanner(System.in);

            System.out.printf("Qual o seu saldo? R$");
            saldo = sc.nextFloat();

            System.out.printf("Quanto deseja sacar? R$");
            saque = sc.nextFloat();

            sacarDinheiro(saque, saldo);

        }catch(InputMismatchException e){

            System.out.println("Por favor insira apenas números.");

        }finally {
```

```
        System.out.println("Finalizando programa...");  
    }  
}
```

**2. Crie uma classe Carro com os atributos marca, modelo e ano, e um método getInformacoes() que retorne uma string com as informações do carro. Crie um objeto dessa classe e chame o método getInformacoes().**

### **Classe Carros**

```
public class Carro {  
  
    private String marca;  
    private String modelo;  
    private int ano;  
  
    public String getMarca() {  
        return marca;  
    }  
  
    public void setMarca(String marca) {  
        this.marca = marca;  
    }  
  
    public String getModelo() {  
        return modelo;  
    }  
  
    public void setModelo(String modelo) {  
        this.modelo = modelo;  
    }  
  
    public int getAno() {  
        return ano;  
    }  
  
    public void setAno(int ano) {  
        this.ano = ano;  
    }  
  
    public void getInformacoes(){  
  
        System.out.println("| -- Informações do carro -- |");  
        System.out.println("A marca do carro é: " + this.getMarca());  
        System.out.println("O seu modelo é: " + this.getModelo());  
        System.out.printf("O carro '%s' é do ano %d", this.getModelo(), this.getAno());  
    }  
}
```

### **Classe Main**

```
import java.util.InputMismatchException;  
import java.util.Scanner;  
  
public class Main {  
  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
        Carro carro1 = new Carro();  
  
        // 1500 ... 1970  
  
        System.out.printf("Marca do carro: ");  
        String marcaCarro = sc.nextLine();  
    }  
}
```

```

        carro1.setMarca(marcaCarro);

        System.out.printf("Modelo do carro: ");
        String modeloCarro = sc.nextLine();
        carro1.setModelo(modeloCarro);

        try{
            System.out.printf("Ano do carro: ");
            int anoCarro = sc.nextInt();
            carro1.setAno(anoCarro);
            carro1.getInformacoes();

        }catch(InputMismatchException e){

            System.out.println("O ano deve ser um número inteiro.");

        }

        sc.close();
    }
}

```

**3. Crie uma classe Banco com os atributos nome e cnpj, e um método abrirConta(String nomeTitular, double saldoInicial) que cria uma conta bancária com o titular e o saldo inicial especificados, e retorna o número da conta. Caso o saldo inicial seja negativo, lance uma exceção SaldoInicialInvalidoException. Crie um objeto dessa classe e teste o método abrirConta().**

### Classe Banco

```

public class Banco extends Exception {

    public static String nomeBanco;
    public static String CNPJ;

    public void abrirConta(String nomeBanco, String CNPJ, String nomeTitular, double saldoInicial) throws SaldoInicialInvalidoException{

        if(saldoInicial < 0){
            throw new SaldoInicialInvalidoException("O saldo inicial não pode ser negativo.");
        }else{
            System.out.println("| -- Conta criada! -- |");
            System.out.println("Banco: " + nomeBanco);
            System.out.println("CNPJ: " + CNPJ);
            System.out.println("Titular: " + nomeTitular);
            System.out.println("Número da conta: 1234 013 123456789-2");
        }
    }
}

```

### Classe Cliente

```

import java.util.InputMismatchException;
import java.util.Scanner;

public class Cliente {

    public static void main(String[] args) throws SaldoInicialInvalidoException {

        Scanner sc = new Scanner(System.in);
        Banco conta1 = new Banco();

        try{

            System.out.printf("Nome do banco: ");
            String nomeBanco = sc.nextLine();

            System.out.printf("CNPJ: ");
            String CNPJ = sc.nextLine();

```

```

        System.out.printf("Nome do titular: ");
        String nomeTitular = sc.nextLine();

        System.out.printf("Saldo inicial: ");
        double saldoInicial = sc.nextDouble();

        conta1.abrirConta(nomeBanco, CNPJ, nomeTitular, saldoInicial);
    } catch (InputMismatchException e) {
        System.out.println("O saldo inicial deve ser um número decimal ou inteiro");
    }

    sc.close();
}
}

```

### **Classe da Exceção**

```

public class SaldoInicialInvalidoException extends Exception{

    public SaldoInicialInvalidoException(String mensagem){
        super(mensagem);
    }
}

```

**4. Crie uma classe Animal com os atributos nome e especie, e um método emitirSom() que exibe o som que o animal faz. Crie duas subclasses de Animal: Cachorro e Gato, cada uma com seu próprio som. Crie um método brincar(Animal animal) que exibe uma mensagem de brincadeira entre o animal que chama o método e o animal passado como parâmetro. Crie um objeto de cada uma das subclasses e chame o método brincar().**

### **Classe Animal**

```

public abstract class Animal {

    String nome;
    String especie;

    public void emitirSom(){}

    public void brincar(Animal animal, Animal animal2){
        System.out.printf("%s está brincando com %s", animal.nome, animal2.nome);
    }

}

```

### **Classe Cachorro**

```

public class Cachorro extends Animal{

    @Override
    public void emitirSom() {
        System.out.printf("Au au!!!");
    }

    public static void main(String[] args) {

        Cachorro cachorro1 = new Cachorro();
        cachorro1.nome = "Golias";
        cachorro1.especie = "Pinscher Austríaco";

        Cachorro cachorro2 = new Cachorro();
        cachorro2.nome = "Caramelo";
        cachorro2.especie = "Vira lata";

        cachorro1.brincar(cachorro1, cachorro2);
    }
}

```

**Classe Gato**

```
public class Gato extends Animal{

    @Override
    public void emitirSom() {
        System.out.printf("Miau miau!!!");
    }

    public static void main(String[] args) {

        Gato gato1 = new Gato();
        gato1.nome = "Junior";
        gato1.especie = "Siamês";

        Gato gato2 = new Gato();
        gato2.nome = "Tigresa";
        gato2.especie = "Vira Lata";

        gato1.brincar(gato1, gato2);
    }
}
```

**5. Crie uma classe Retangulo com os atributos base e altura, e os métodos getBase() e getAltura() que retornem os respectivos atributos. Adicione um método calcularArea() que calcula e retorna a área do retângulo. No construtor da classe, lance uma exceção DimensaoInvalidaException caso a base ou a altura seja negativa. Crie um objeto dessa classe e teste o método calcularArea().**

**Classe Retângulo**

```
import java.util.Scanner;

public class Retangulo {
    private double base;
    private double altura;

    public Retangulo(double base, double altura) throws DimensaoInvalidaException {
        if (base < 0 || altura < 0) {
            throw new DimensaoInvalidaException("As dimensões devem ser maiores ou iguais a zero.");
        }
        this.base = base;
        this.altura = altura;
    }

    public double getBase() {
        return this.base;
    }

    public double getAltura() {
        return this.altura;
    }

    public double calcularArea() {
        return this.base * this.altura;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        double base = 0;
        double altura = 0;

        while (base <= 0) {
            System.out.print("Digite a base do retângulo: ");
            base = scanner.nextDouble();
            if (base <= 0) {
                System.out.println("A base deve ser maior que zero.");
            }
        }
    }
}
```

```
while (altura <= 0) {
    System.out.print("Digite a altura do retângulo: ");
    altura = scanner.nextDouble();
    if (altura <= 0) {
        System.out.println("A altura deve ser maior que zero.");
    }
}

try {
    Retangulo retangulo = new Retangulo(base, altura);
    System.out.println("Área do retângulo: " + retangulo.calcularArea());
} catch (DimensaoInvalidaException e) {
    System.out.println(e.getMessage());
}

scanner.close();
}
```

### **Classe da Exceção**

```
public class DimensaoInvalidaException extends Exception {
    public DimensaoInvalidaException(String message) {
        super(message);
    }
}
```