

# Geometria

Link GitHub: [https://github.com/KC2Dbk5/Programa-o-Orientada-a-Objetos---4-Perodo/tree/main/ciclo\\_2/geometry/src/geometry](https://github.com/KC2Dbk5/Programa-o-Orientada-a-Objetos---4-Perodo/tree/main/ciclo_2/geometry/src/geometry)

- Interface: GeometryInterface

```
GeometryInterface.java × GeometricObject.java Rectangle.java Main.java
1 package geometry;
2
3 public interface GeometryInterface {
4     double calculateArea();
5     double calculatePerimeter();
6     double getArea();
7     double getPerimeter();
8     String getColor();
9     void setColor(String color);
10 }
11
```

- Classe GeometricObject

```
GeometryInterface.java × GeometricObject.java × Rectangle.java Main.java
1 package geometry;
2
3 public abstract class GeometricObject implements GeometryInterface {
4
5     protected double area;
6     protected double perimeter;
7     protected String color;
8
9     public abstract double calculateArea();
10    public abstract double calculatePerimeter();
11
12    public String getColor() {
13        return color;
14    }
15
16    public void setColor(String color) {
17        this.color = color;
18    }
19
20    public double getArea() {
21        return area;
22    }
23
24    public double getPerimeter() {
25        return perimeter;
26    }
27 }
28
```

- Classe: Rectangle

```
1 package geometry;
2
3 public class Rectangle extends GeometricObject {
4
5     private double length;
6     private double height;
7
8     public Rectangle(double length, double height, String color) {
9         this.length = length;
10        this.height = height;
11        setColor(color);
12        calculateArea();
13        calculatePerimeter();
14    }
15
16    @Override
17    public double calculateArea() {
18        area = length * height;
19        return area;
20    }
21
22    @Override
23    public double calculatePerimeter() {
24        perimeter = 2 * (length + height);
25        return perimeter;
26    }
27
28    public double getLength() {
29        return length;
30    }
31
32    public void setLength(double length) {
33        this.length = length;
34        calculateArea();
35        calculatePerimeter();
36    }
37
38    public double getHeight() {
39        return height;
40    }
41
42    public void setHeight(double height) {
43        this.height = height;
44        calculateArea();
45        calculatePerimeter();
46    }
47
48    public boolean isSquare() {
49        return length == height;
50    }
51
52    public double getDiagonal() {
53        return Math.sqrt(Math.pow(length, 2) + Math.pow(height, 2));
54    }
55
56    public void resize(double scaleFactor) {
57        this.length *= scaleFactor;
58        this.height *= scaleFactor;
59        calculateArea();
60        calculatePerimeter();
61    }
62 }
```

- Classe: Main

```
1 import geometry.Rectangle;
2
3 public class Main {
4     public static void main(String[] args) {
5         Rectangle rect = new Rectangle(5, 10, "blue");
6         System.out.println("Área: " + rect.getArea());
7         System.out.println("Perímetro: " + rect.getPerimeter());
8         System.out.println("Cor: " + rect.getColor());
9         System.out.println("É um quadrado? " + rect.isSquare());
10        System.out.println("Diagonal: " + rect.getDiagonal());
11    }
12 }
13 }
```

- Saída

```
Área: 50.0
Perímetro: 30.0
Cor: blue
É um quadrado? false
Diagonal: 11.180339887498949
```

# Banco

Link GitHub: [https://github.com/KC2Dbk5/Programa-o-Orientada-a-Objetos---4-Perodo/tree/main/ciclo\\_2/banco/src/banco](https://github.com/KC2Dbk5/Programa-o-Orientada-a-Objetos---4-Perodo/tree/main/ciclo_2/banco/src/banco)

- Classe: Conta

```
1 package banco;
2
3 abstract public class Conta {
4
5     protected String numero;
6     protected double saldo;
7
8     public String getNumero() {
9         return numero;
10    }
11
12    public void setNumero(String numero) {
13        this.numero = numero;
14    }
15
16    public double getSaldo() {
17        return saldo;
18    }
19
20    protected void setSaldo(double saldo) {
21        this.saldo = saldo;
22    }
23
24    abstract void creditar(double valor);
25
26    abstract void debitar(double valor);
27 }
```

- Classe: ContaNormal

```
1 package banco;
2
3 public class ContaNormal extends Conta {
4
5     @Override
6     public void creditar(double valor) {
7         saldo += valor;
8     }
9
10    @Override
11    public void debitar(double valor) {
12        if (saldo >= valor) {
13            saldo -= valor;
14        } else {
15            System.out.println("Saldo insuficiente para débito.");
16        }
17    }
18 }
```

- Classe: ContaEspecial

```
1 package banco;
2
3 abstract public class ContaEspecial extends Conta {
4
5     protected double limite;
6
7     public double getLimite() {
8         return limite;
9     }
10
11    public void setLimite(double limite) {
12        this.limite = limite;
13    }
14
15    @Override
16    public void creditar(double valor) {
17        saldo += valor;
18    }
19 }
```

- Classe: ContaDebEspecial

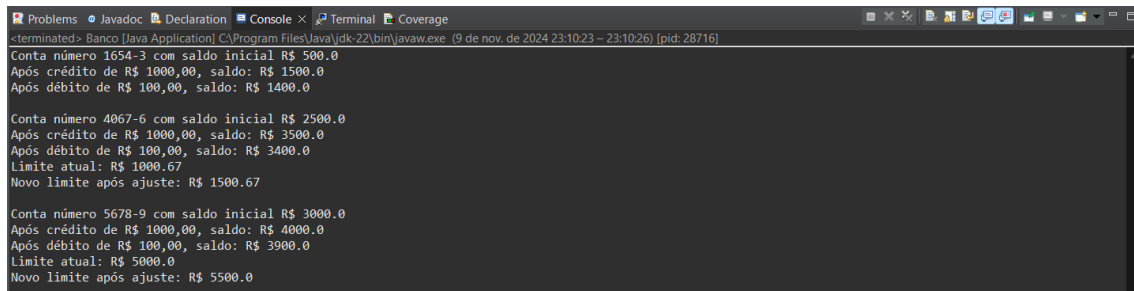
```
1 package banco;
2
3 public class ContaDebEspecial extends ContaEspecial {
4
5     public ContaDebEspecial() {}
6
7     public ContaDebEspecial(String numero, double saldo, double limite) {
8         this.numero = numero;
9         this.saldo = saldo;
10        this.limite = limite;
11    }
12
13    @Override
14    public void debitar(double valor) {
15        if ((saldo + limite) >= valor) {
16            saldo -= valor;
17        } else {
18            System.out.println("Limite insuficiente para débito.");
19        }
20    }
21 }
22
```

- Classe com o método Main: Banco

```
1 package banco;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Banco {
7
8     public static void main(String[] args) {
9
10        List<Conta> contas = new ArrayList<>();
11
12        ContaNormal conta1 = new ContaNormal();
13        conta1.setNumero("1654-3");
14        conta1.setSaldo(500);
15
16        ContaDebEspecial conta2 = new ContaDebEspecial("4067-6", 2500, 1000.67);
17        ContaDebEspecial conta3 = new ContaDebEspecial("5678-9", 3000, 5000);
18
19        contas.add(conta1);
20        contas.add(conta2);
21        contas.add(conta3);
22
23        realizarOperacoes(contas);
24    }
25
26    private static void realizarOperacoes(List<Conta> contas) {
27        for (Conta conta : contas) {
28            System.out.println("Conta número " + conta.getNumero() + " com saldo inicial R$ " + conta.getSaldo());
29
30            conta.creditar(1000);
31            System.out.println("Após crédito de R$ 1000,00, saldo: R$ " + conta.getSaldo());
32
33            conta.debitar(100);
34            System.out.println("Após débito de R$ 100,00, saldo: R$ " + conta.getSaldo());
35
36            if (conta instanceof ContaDebEspecial) {
37                ContaDebEspecial contaEspecial = (ContaDebEspecial) conta;
38                System.out.println("Limite atual: R$ " + contaEspecial.getLimite());
39
40                contaEspecial.setLimite(contaEspecial.getLimite() + 500);
41                System.out.println("Novo limite após ajuste: R$ " + contaEspecial.getLimite());
42            }
43            System.out.println();
44        }
45    }
46 }
```

## Aluno: Kayran Henrique Reis Teixeira - 2310370

- Saída



```
<terminated> Banco [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (9 de nov. de 2024 23:10:23 - 23:10:26) [pid: 28716]
Conta número 1654-3 com saldo inicial R$ 500.0
Após crédito de R$ 1000,00, saldo: R$ 1500.0
Após débito de R$ 100,00, saldo: R$ 1400.0

Conta número 4067-6 com saldo inicial R$ 2500.0
Após crédito de R$ 1000,00, saldo: R$ 3500.0
Após débito de R$ 100,00, saldo: R$ 3400.0
Limite atual: R$ 1000.67
Novo limite após ajuste: R$ 1500.67

Conta número 5678-9 com saldo inicial R$ 3000.0
Após crédito de R$ 1000,00, saldo: R$ 4000.0
Após débito de R$ 100,00, saldo: R$ 3900.0
Limite atual: R$ 5000.0
Novo limite após ajuste: R$ 5500.0
```