

Projeto Da Disciplina

Anápolis, 04 de novembro de 2024

Nome da disciplina: Programação Orientada a Objetos

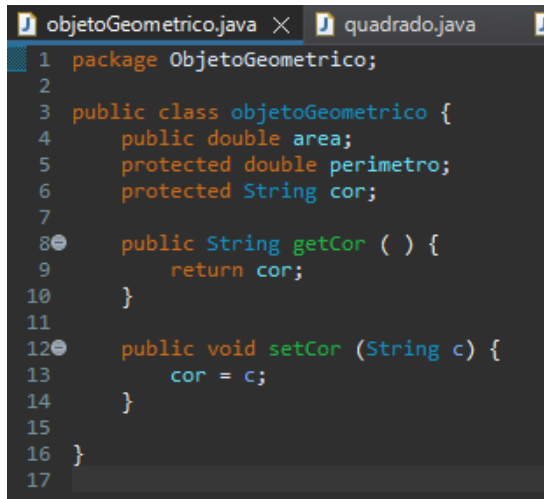
Alunos: Matheus Marques Portela

A seguir, apresento os dois projetos desenvolvidos durante nossas aulas, ambos implementados em Java e focados nos conceitos do paradigma de programação orientada a objetos. O primeiro projeto, denominado objetosGeometricos, é voltado para a modelagem e manipulação de diferentes formas geométricas, permitindo calcular suas áreas, perímetros e setar as cores de cada objeto. O segundo projeto, chamado banco, foi criado com o intuito de simular um sistema bancário, onde diversas classes representam tipos de contas, operações financeiras e funcionalidades associadas, como crédito, débito. Esses projetos visam não apenas aplicar os princípios da programação orientada a objetos, como encapsulamento, herança e polimorfismo, mas também desenvolver habilidades práticas em design e implementação de software.

- **objetosGeometricos:**

- **Link do GitHub:** <https://github.com/oPortela/ObjetoGeometrico/tree/main>

1. Classe objetoGeometrico:



```
1 package ObjetoGeometrico;
2
3 public class objetoGeometrico {
4     public double area;
5     protected double perimetro;
6     protected String cor;
7
8     public String getCor ( ) {
9         return cor;
10    }
11
12    public void setCor (String c) {
13        cor = c;
14    }
15
16 }
17
```

2. Classe quadrado:

```
objetoGeometrico.java | quadrado.java | Retangulo.java
1 package ObjetoGeometrico;
2
3 public class quadrado extends objetoGeometrico {
4     private double lado;
5     quadrado(double l) {
6         lado = l;
7     }
8
9     public double getArea ( ) {
10         area = lado * lado;
11         return area;
12     }
13
14     public double getPerimetro ( ) {
15         perimetro = 2 * lado;
16         return perimetro;
17     }
18
19 }
```

3. Classe retângulo:

```
objetoGeometrico.java | quadrado.java | Retangulo.java | Triangulo.java
1 package ObjetoGeometrico;
2
3 public class Retangulo extends objetoGeometrico {
4     private double comprimento;
5     private double altura;
6
7     Retangulo(double c, double a) {
8         comprimento = c;
9         altura = a;
10    }
11
12    public double getArea ( ) {
13        area = comprimento * altura;
14        return area;
15    }
16
17    public double getPerimetro ( ) {
18        perimetro = comprimento + altura;
19        return perimetro;
20    }
21
22 }
```

4. Classe Triângulo:

```
objetoGeometrico.java  quadrado.java  Retangulo.java  Triangulo.java X
1 package ObjetoGeometrico;
2
3 public class Triangulo extends objetoGeometrico {
4     public double base;
5     public double altura;
6     Triangulo(double b, double a) {
7         base = b;
8         altura = a;
9     }
10
11     public double getArea ( ) {
12         area = base * altura /2;
13         return area;
14     }
15
16     public double getPerimetro ( ) {
17         perimetro = 3 * base;
18         return perimetro;
19     }
20
21 }
```

5. Classe circunferência:

```
objetoGeometrico.java  quadrado.java  Retangulo.java  Triangulo.java  Circunferencia.java X
1 package ObjetoGeometrico;
2
3 public class Circunferencia extends objetoGeometrico {
4     private double pi = 3.141618;
5     private double raio;
6
7     Circunferencia(double r) {
8         raio = r;
9     }
10    public double getArea ( ) {
11        area = pi * (raio * raio);
12        return area;
13    }
14
15    public double getPerimetro ( ) {
16        perimetro = 2 * pi * raio;
17        return perimetro;
18    }
19
20 }
```

6. Classe QuadroNegro:

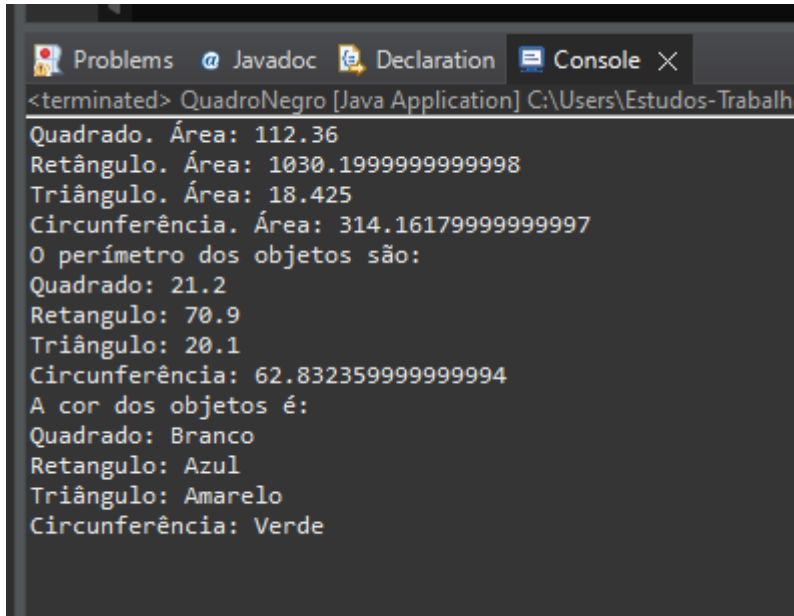
Classe voltado a setar os valores de área e perímetro dos objetos criados, além de definir uma cor para cada um deles e por fim imprimir o resultado na tela ao usuário.

```

1 package ObjetoGeometrico;
2
3 public class QuadroNegro {
4     private quadrado quadrado;
5     private Retangulo retangulo;
6     private Circunferencia circunferencia;
7     private Triangulo triangulo;
8
9     public void limpaQuadro ( ) {
10
11     }
12
13     public void criaQuadrado ( double l, String c ) {
14         quadrado = new quadrado(l);
15         quadrado.setCor(c);
16     }
17
18     public void criaRetangulo ( double c, double a, String cor ) {
19         retangulo = new Retangulo(c, a);
20         retangulo.setCor(cor);
21     }
22
23     public void criaCircunferencia ( double r, String c ) {
24         circunferencia = new Circunferencia(r);
25         circunferencia.setCor(c);
26     }
27
28     public void criaTriangulo ( double b, double a, String c ) {
29         triangulo = new Triangulo(b, a);
30         triangulo.setCor(c);
31     }
32
33
34     public void mostraAreaObjetos ( ) {
35         System.out.println("Quadrado. Área: " + quadrado.getArea());
36         System.out.println("Retângulo. Área: " + retangulo.getArea());
37         System.out.println("Triângulo. Área: " + (triangulo.base * triangulo.altura)/2);
38         System.out.println("Circunferência. Área: " + circunferencia.getArea());
39     }
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57     public static void main ( String args[] ) {
58
59
60         QuadroNegro quadroNegro = new QuadroNegro();
61         quadroNegro.criaQuadrado(10.6, "Branco");
62         quadroNegro.criaRetangulo(50.5,20.4, "Azul");
63         quadroNegro.criaTriangulo(6.7,5.5, "Amarelo");
64         quadroNegro.criaCircunferencia(10, "Verde");
65         quadroNegro.mostraAreaObjetos();
66         quadroNegro.mostraPerimetroObjetos();
67         quadroNegro.mostraCorObjetos();
68     }
69
70
71 }

```

7. Resultado da compilação do código acima:

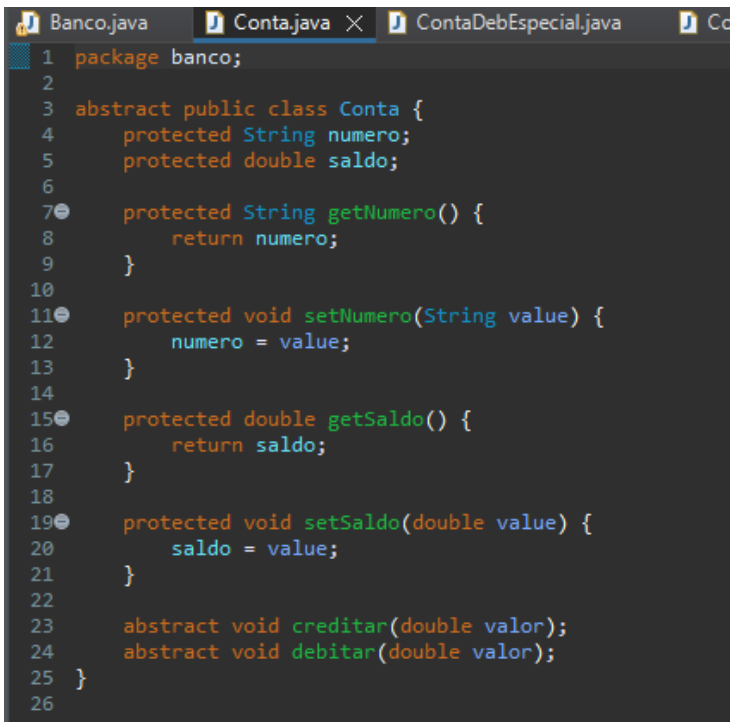


```
<terminated> QuadroNegro [Java Application] C:\Users\Estudos-Trabalho\
Quadrado. Área: 112.36
Retângulo. Área: 1030.1999999999998
Triângulo. Área: 18.425
Circunferência. Área: 314.16179999999997
O perímetro dos objetos são:
Quadrado: 21.2
Retangulo: 70.9
Triângulo: 20.1
Circunferência: 62.832359999999994
A cor dos objetos é:
Quadrado: Branco
Retangulo: Azul
Triângulo: Amarelo
Circunferência: Verde
```

- Banco:

- Link do GitHub: <https://github.com/oPortela/banco>

1) Classe conta:



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

2) Classe ContaDebEspecial:

```
Banco.java  Conta.java  ContaDebEspecial.java  ContaEspecial.java
1 package banco;
2
3 public class ContaDebEspecial extends ContaEspecial{
4     ContaDebEspecial(){
5         super();
6     }
7
8     ContaDebEspecial(String n, double s, double l){
9         numero = n;
10        saldo = s;
11        limite = l;
12    }
13
14    void debitar(double valor) {
15        if((limite + saldo - valor) >= 0) {
16            saldo = saldo - valor;
17        }
18    }
19 }
20
```

3) Classe ContaEspecial:

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

4) Classe ContaEstudantil:

```
Banco.java  Conta.java  ContaDebEspecial.java  ContaEs
1  package banco;
2
3  public class ContaEstudantil extends ContaNormal{
4      private double emprestimo;
5
6      public void setEmprestimo(double e) {
7          emprestimo = e;
8      }
9
10     /*public void getEmprestimo() {
11         return emprestimo;
12     }*/
13
14     public void creditar(double valor) {
15         saldo = saldo + valor;
16     }
17
18     public void debitar(double valor) {
19         if((saldo - valor + emprestimo) >= 0) {
20             saldo = saldo - valor;
21         }
22     }
23 }
24
```

5) Classe ContaNormal:

```
Banco.java  Conta.java  ContaDebEspecial.java  ContaEspecial.java  ContaEstud
1  package banco;
2
3  public class ContaNormal extends Conta {
4      public void creditar(double valor) {
5          if (valor > 0) {
6              saldo += valor;
7          } else {
8              System.out.println("Valor para crédito deve ser positivo.");
9          }
10     }
11
12     public void debitar(double valor) {
13         if (valor > saldo) {
14             System.out.println("Saldo insuficiente para realizar a operação.");
15         } else {
16             saldo -= valor;
17         }
18     }
19 }
20
```

6) Classe Banco:

Na classe banco usamos ela para setar os valores e chamar as função que criamos em cada uma das classes acima e dentro da classe Banco fazemos ela imprimir ao usuário os resultados requisitados.

```
1 package banco;
2
3 public class Banco {
4     public static void main(String args[]) {
5         Conta c1;
6         c1 = new ContaNormal();
7         c1.setNumero(new String("1654-3"));
8         c1.setSaldo(500);
9         ContaDebEspecial c2 = new ContaDebEspecial();
10        ContaDebEspecial c3 = new ContaDebEspecial(new String("4067-6"),2500,5050);
11
12        ContaEstudantil c4 = new ContaEstudantil();
13
14        c2.setNumero(new String("4067-6"));
15        c2.setSaldo(2500);
16        c2.setLimite(1000.67);
17
18        c4.setNumero(new String("1234-5"));
19        c4.setSaldo(1800);
20        c4.setEmprestimo(1000.67);
21
22
23
24        System.out.println("A conta número " + c1.getNumero() + " possui saldo " + c1.getSaldo());
25        c1.creditar(1000);
26        System.out.println("Após o crédito de R$ 1000,00, a conta número " + c1.getNumero() +
27        " passou a ter saldo " + c1.getSaldo());
28        c1.debitar(100);
29        System.out.println("Após o débito de R$ 100,00 a conta número " + c1.getNumero() +
30        " passou a ter o saldo de " + c1.getSaldo());
31
32        System.out.println("");
33
34        System.out.println("A conta número " + c2.getNumero() + " possui saldo " + c2.getSaldo());
35        c2.debitar(500);
36        System.out.println("A conta número " + c2.getNumero() + " possui saldo " + c2.getSaldo() + " após o denito de R$500,00");
37        System.out.println("A conta número " + c2.getNumero() + " possui saldo " + c2.getSaldo() + " e limite de " +
38        c2.getLimite());
39        c2.setLimite(10000);
40        System.out.println("A conta número " + c2.getNumero() + " possui saldo " + c2.getSaldo() + " e o novo limite de" +
41        c2.getLimite());
42    }
43 }
44
```

7) Resultado da compilação do projeto Banco:

```
Problems Javadoc Declaration Console
<terminated> Banco [Java Application] C:\Users\Estudios-Trabalho\p2\pool\plugins\org.eclipse.justj.openjdk.hot
A conta número 1654-3 possui saldo 500.0
Após o crédito de R$ 1000,00, a conta número 1654-3 passou a ter saldo 1500.0
Após o débito de R$ 100,00 a conta número 1654-3 passou a ter o saldo de 1400.0

A conta número 4067-6 possui saldo 2500.0
A conta número 4067-6 possui saldo 2000.0 após o denito de R$500,00
A conta número 4067-6 possui saldo 2000.0 e limite de 1000.67
A conta número 4067-6 possui saldo 2000.0 e o novo limite de10000.0
```