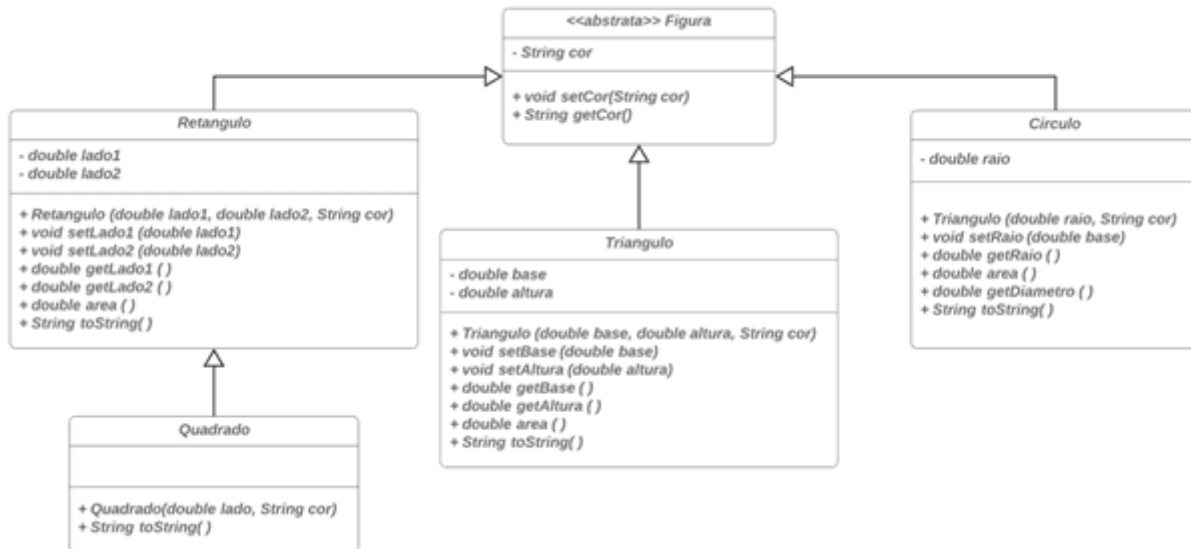


## Exercícios resolvidos de Orientação a Objetos em Java

Vou trazer alguns exercícios resolvidos de Java que tive na faculdade em POO, tratando de interfaces e classe abstrata.

### 1. Implemente o seguinte diagrama de classes em JAVA:



Para resolver este exercício, primeiro vamos criar uma classe abstrata (não pode ser instanciada) chamada **Figura**, que contém atributos e métodos em comum com todas as classes de **Figura**. Nós teremos um atributo **cor** do tipo **String** e dois métodos, um com retorno nulo setando a **Cor** e outro com retorno tipo **String** pegando a **Cor**.

```

1  package exAula08;
2
3  public abstract class Figura {
4      private String cor;
5
6      public Figura() {
7      }
8
9      public Figura(String cor) {
10         super();
11         this.cor = cor;
12     }
13
14     public String getCor() {
15         return cor;
16     }
17
18     public void setCor(String cor) {
19         this.cor = cor;
20     }
21
22     @Override
23     public String toString() {
24         return "Figura [cor=" + cor + "]";
25     }
26
27     public abstract double area();
28 }
  
```

Em seguida, vamos criar a classe **Retângulo** que vai estender a classe **Figura** e declarar dois atributos nela como **double** (**lado1** e **lado2**). Vamos também criar o construtor padrão, os getters/setters, o métodos **toString**

(para retornar um texto) e um método chamado `double área()`, onde colocaremos um `return` com a função `lado 1 * lado 2`, para obtermos a área.

```
1  package exAula08;
2
3  public class Retangulo extends Figura {
4      private double lado1;
5      private double lado2;
6
7      @Override
8      public double area() {
9          return this.lado1 * this.lado2;
10     }
11
12     public Retangulo(String cor, double lado1, double lado2) {
13         super(cor);
14         this.lado1 = lado1;
15         this.lado2 = lado2;
16     }
17
18     public Retangulo() {
19     }
20
21     public double getLado1() {
22         return lado1;
23     }
24
25     public void setLado1(double lado1) {
26         this.lado1 = lado1;
27     }
28
29     public double getLado2() {
30         return lado2;
31     }
32
33     public void setLado2(double lado2) {
34         this.lado2 = lado2;
35     }
36
37     @Override
38     public String toString() {
39         return "Retangulo [lado1= " + lado1 + ", lado2= " + lado2 + ", cor= " + super.getCor() + "];
40     }
41 }
```

Vamos criar uma classe `Triângulo` que vai estender a classe `Figura` com os atributos base e altura, criar o construtor padrão, getters/setters, método `toString` e método `double área` que retorna `base * altura`.

```
1  package exAula08;
2
3  public class Triangulo extends Figura {
4      private double base;
5      private double altura;
6
7      public Triangulo() {
8      }
9
10     public Triangulo(String cor, double base, double altura) {
11         super(cor);
12         this.base = base;
13         this.altura = altura;
14     }
15
16     public double getBase() {
17         return base;
18     }
19
20     public void setBase(double base) {
21         this.base = base;
22     }
23
24     public double getAltura() {
25         return altura;
26     }
27
28     public void setAltura(double altura) {
29         this.altura = altura;
30     }
31
32     @Override
33     public double area() {
34         return base * altura;
35     }
36
37     @Override
38     public String toString() {
39         return "Triangulo [base=" + base + ", altura=" + altura + ", cor= " + super.getCor() + "]\n";
40     }
41 }
```

Vamos criar uma classe Círculo que vai estender a classe Figura com os atributos raio, criar o construtor padrão, getters/setters, método toString e o método getDiametro() que retorna a fórmula  $\text{raio} * 3.14$  (valor de Pi) \* 2.

```
1  package exAula08;
2
3  public class Circulo extends Figura {
4      private double raio;
5
6      public Circulo(String cor, double raio) {
7          super(cor);
8          this.raio = raio;
9      }
10
11     public Circulo() {
12     }
13
14     public double getRaio() {
15         return raio;
16     }
17
18     public void setRaio(double raio) {
19         this.raio = raio;
20     }
21
22     @Override
23     public String toString() {
24         return "Circulo [raio=" + raio + ", cor= " + super.getCor() + "];";
25     }
26
27     @Override
28     public double area() {
29         return raio * 3.14 * 2;
30     }
31 }
```

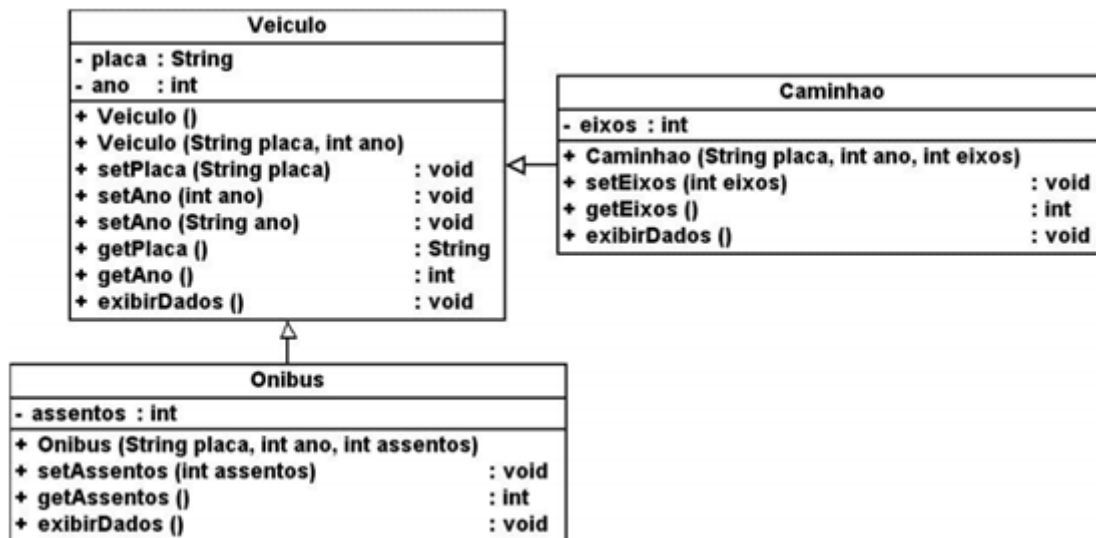
Vamos criar uma classe Quadrado que vai estender a classe Retângulo com um construtor super (de Retângulo) e o método toString.

```
1 public class Quadrado extends Retangulo {
2     public Quadrado() {
3         super();
4     }
5
6     public Quadrado(String cor, double lado1, double lado2) {
7         super(cor, lado1, lado2);
8     }
9
10    public Quadrado(String cor, double lado) {
11        lado = super.getLado1();
12    }
13
14    @Override
15    public double area() {
16        return this.getLado1() * this.getLado1();
17    }
18
19    @Override
20    public String toString() {
21        return "Quadrado [area()=" + area() + ", cor= " + super.getCor() + "]\n";
22    }
23 }
```

Para finalizar, vamos criar uma classe chamado Teste para criarmos os objetos de todas as classes que criamos e ver o resultado

```
1 package exAula08;
2
3 public class Teste {
4     public static void main(String[] args) {
5         Quadrado quadrado = new Quadrado();
6         Triangulo triangulo = new Triangulo("Azul", 5, 12);
7         Circulo circulo = new Circulo("Verde", 10.2);
8         Retangulo retangulo = new Retangulo("Rosa", 32, 12);
9
10        quadrado.setCor("Roxo");
11        quadrado.setLado1(4);
12
13        System.out.println(quadrado);
14        System.out.println(triangulo);
15        System.out.println(circulo);
16        System.out.println(retangulo);
17    }
18 }
```

**2. Criar um projeto Java no Eclipse chamado "Heranca\_02", que implemente todo o código necessário para a criação das classes abaixo:**



Vamos criar uma classe Veiculo, com os atributos placa e ano. Vamos criar o construtor padrão e o vazio, getters/setters e a função exibir dados, que vai apresentar um texto com os dados do veículo.

```

1  package herancaSegundo;
2
3  public class Veiculo {
4      protected String placa;
5      protected int ano;
6
7      public Veiculo(String placa, int ano) {
8          this.placa = placa;
9          this.ano = ano;
10     }
11
12     public Veiculo() {
13     }
14
15     public String getPlaca() {
16         return placa;
17     }
18
19     public void setPlaca(String placa) {
20         this.placa = placa;
21     }
22
23     public int getAno() {
24         return ano;
25     }
26
27     public void setAno(int ano) {
28         this.ano = ano;
29     }
30
31     public void exibirDados() {
32         System.out.println("A placa do veículo é " + placa + " e o ano de fabricação é " + ano);
33     }
34 }
  
```

Vamos criar uma classe Ônibus que estende Veiculo e terá o atributo assentos, vamos criar o construtor com superclasse, getters/setters e sobrescrever o método exibirDados(), que exibirá os dados do ônibus.



```
1 package herancaSegundo;
2
3 public class Onibus extends Veiculo {
4     private int assentos;
5
6     public Onibus(String placa, int ano, int assentos) {
7         super(placa, ano);
8         this.assentos = assentos;
9     }
10
11     public int getAssentos() {
12         return assentos;
13     }
14
15     public void setAssentos(int assentos) {
16         this.assentos = assentos;
17     }
18
19     @Override
20     public void exibirDados() {
21         System.out.println("A placa do veículo é " + placa + " e o ano de fabricação é " + ano
22             + ", o número de assentos é " + assentos);
23     }
24 }
```

Vamos criar a classe Caminhão que estende Veiculo e possui atributo de eixos, vamos criar o construtor vazio, construtor com super classe, getters/setters e o método dados para exibir os dados do Caminhão (sobrescrita do método).

```
1 package herancaSegundo;
2
3 public class Caminhao extends Veiculo {
4     private int eixos;
5
6     public Caminhao(String placa, int ano, int eixos) {
7         super(placa, ano);
8         this.eixos = eixos;
9     }
10
11     public Caminhao() {
12     }
13
14     public int getEixos() {
15         return eixos;
16     }
17
18     public void setEixos(int eixos) {
19         this.eixos = eixos;
20     }
21
22     public void exibirDados() {
23         System.out.println("A placa do veículo é " + placa + " e o ano de fabricação é " + ano
24             + ", a quantidade de eixos é " + eixos);
25     }
26 }
```

Para finalizar, vamos criar uma classe Main para testarmos as classes que criamos. Vamos criar um array de veículos, que armazenará todos os nossos objetos Veículos e Caminhões. E, por fim, vamos criar dois laços de repetição para exibirmos os dados dos objetos que criamos.

```
1  package herancaSegundo;
2
3  import java.util.ArrayList;
4
5  public class Main {
6      public static void main(String[] args) {
7          ArrayList<Veiculo> veiculos = new ArrayList<Veiculo>();
8          veiculos.add(new Veiculo("XPO-0987", 2010));
9          veiculos.add(new Veiculo("ABC-1234", 2012));
10         veiculos.add(new Veiculo("KYN-0987", 2014));
11         veiculos.add(new Caminhao("NLT-3028", 2014, 5));
12         veiculos.add(new Caminhao("HJU-3341", 2000, 7));
13
14         ArrayList<Caminhao> caminhoes = new ArrayList<Caminhao>();
15         veiculos.add(new Veiculo("XPO-0987", 2010));
16         veiculos.add(new Caminhao("HJU-3341", 2000, 7));
17
18         for (Veiculo veiculo : veiculos) {
19             veiculo.exibirDados();
20         }
21
22         for (Caminhao caminhao : caminhoes) {
23             caminhao.exibirDados();
24         }
25     }
26 }
```

Top comments (0) ↕