



Instituto Infnet

Fundamentos de Desenvolvimento com Java

Desenvolvimento Back-end

2025T2

Lohran Fellipe Mendes de Souza

TP3 - Escrever programas em Java que utilizem classes e objetos

Observação:

O tp e os enunciados das questões parecem ter sido criados por uma inteligência artificial, pedindo para que a gente entregasse as respostas de uma maneira pouco natural, mesmo assim, eu tentei fazer tudo o que foi pedido ao pé da letra, por isso verá que no código há partes que não há nenhuma saída no console, pois a questão não pedia, e outras partes que não estão sendo instanciadas no método Main, fiz isso para facilitar o entendimento e porque a maneira que a questão pediu dificulta a compreensão do resto do código, qualquer comentário pode me retornar.

Exercício 1

A linguagem de programação Java usa um paradigma de programação orientada a objetos, objetos que são criados a partir de classes.

Tá bom, mas o que são Classes?

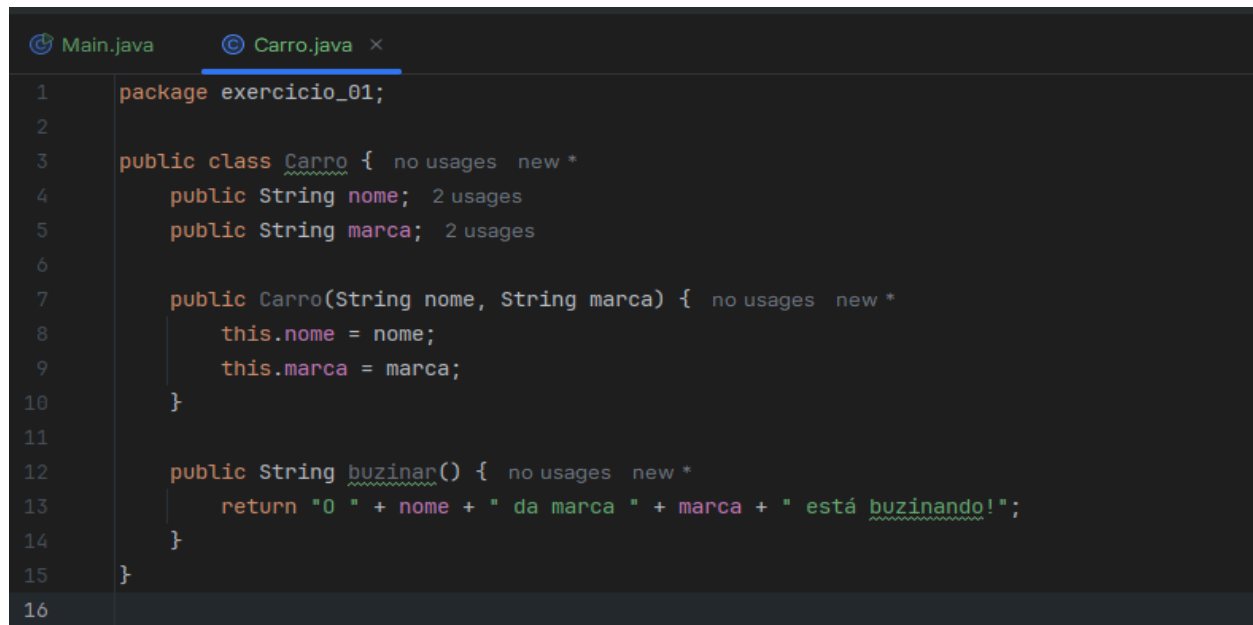
Classes são criadas em Java para definir a estrutura ou um padrão de objetos com que iremos trabalhar, dentro dessas classes nós definimos a regra de negócio, seus métodos e atributos, feito isso, nós podemos instâncias desses objetos e trabalhar em cima de suas funcionalidades.

Atributos são as características desse objeto, como a cor dele, o peso, altura, estas são propriedades necessárias que podem ou devem ser

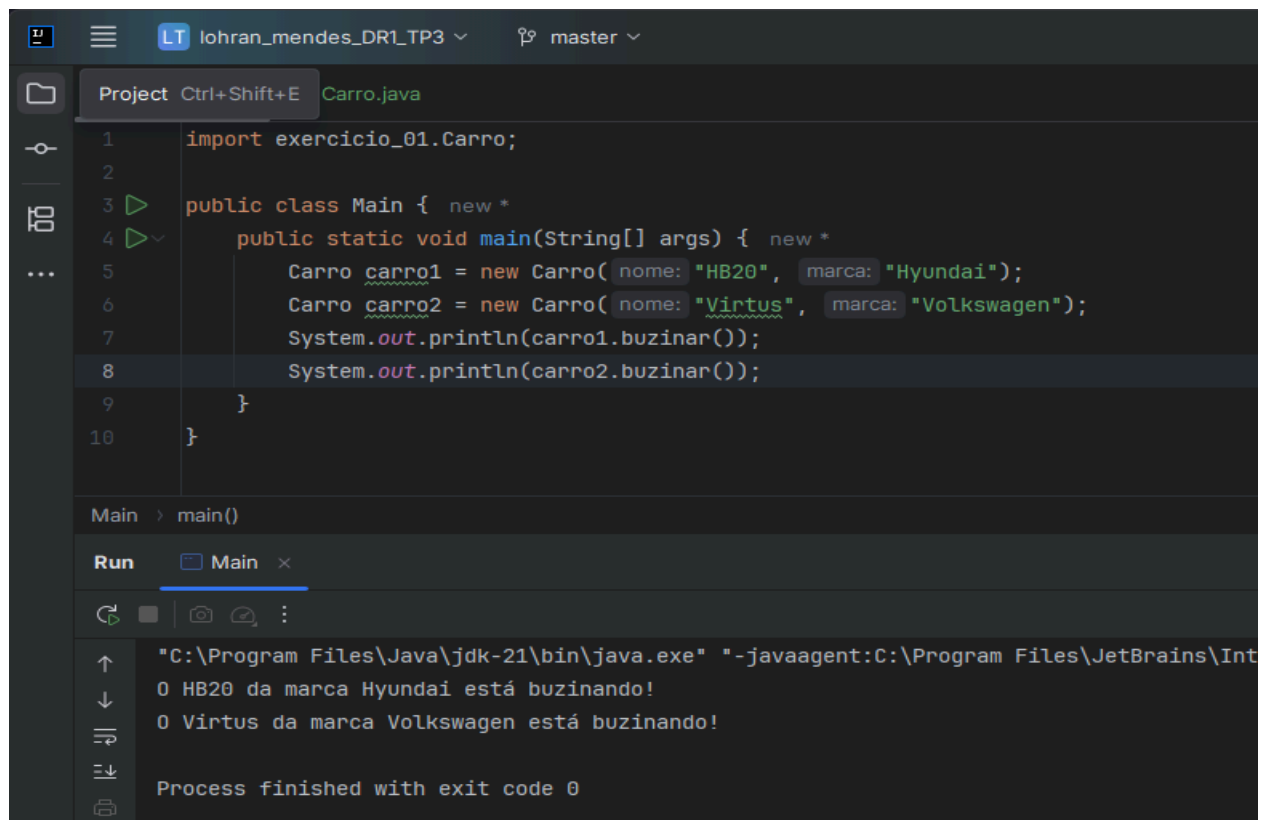
3

usadas dentro do código, enquanto os métodos são ações que esse objeto criado a partir de uma classe deve fazer.

Como exemplo, criarei uma classe de carros, e irei instanciar dois diferentes:



```
1 package exercicio_01;
2
3 public class Carro { no usages new *
4     public String nome; 2 usages
5     public String marca; 2 usages
6
7     public Carro(String nome, String marca) { no usages new *
8         this.nome = nome;
9         this.marca = marca;
10    }
11
12    public String buzinar() { no usages new *
13        return "0 " + nome + " da marca " + marca + " está buzinando!";
14    }
15 }
16
```



```
Project Ctrl+Shift+E Carro.java
1 import exercicio_01.Carro;
2
3 public class Main { new *
4     public static void main(String[] args) { new *
5         Carro carro1 = new Carro( nome: "HB20", marca: "Hyundai");
6         Carro carro2 = new Carro( nome: "Virtus", marca: "Volkswagen");
7         System.out.println(carro1.buzinar());
8         System.out.println(carro2.buzinar());
9     }
10 }
```

Main > main()

Run Main x

"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea_rt.jar=5033:C:\Program Files\Java\jdk-21\bin" -Dfile.encoding=UTF-8

0 HB20 da marca Hyundai está buzinando!

0 Virtus da marca Volkswagen está buzinando!

Process finished with exit code 0

Como no exemplo mostrado, os objetos `carro1` e `carro2` foram criados, cada um com suas próprias propriedades de nome e marca, e os dois tinham um método que corresponde a uma ação que todo objeto da classe `carro` deve ter, o de buzinar.

Exercício 2

Cada atributo é importante pois são essenciais no contexto apresentado de um supermercado, todo produto precisa ter um nome para identificá-lo, um preço para que esse seja vendido, e a sua quantidade disponível no estoque para que o vendedor consiga fazer o controle.

```
Main.java  Produto.java  Carro.java
1 package exercicio_2;
2
3 public class Produto { no usages new *
4     public String nome; no usages
5     public double preco; no usages
6     public int quantidadeEmEstoque; no usages
7 }
8
```

Exercício 3

```
Main.java  Produto.java
1 package exercicio_2;
2
3 public class Produto { no usages new *
4     public String nome; 1 usage
5     public double preco; 2 usages
6     public int quantidadeEmEstoque; 3 usages
7
8     public void alterarPreco(double novoPreco){ no usages new *
9         this.preco = novoPreco;
10    }
11    public void alterarQuantidade(int novaQuantidade){ no usages new *
12        this.quantidadeEmEstoque = novaQuantidade;
13    }
14    public String exibirInformacoes(){ no usages new *
15        return "0 produto " + this.nome + " custa R$" + this.preco + ", e temos o total de " + this.quantidadeEmEstoque + " " + (this.quantidadeEmEstoque == 1 ?
16            "disponível no estoque": " disponiveis no estoque");
17    }
18 }
```

Exercício 4

```
1 package exercicio_2;
2
3 public class Produto { 4 usages new *
4     public String nome; 2 usages
5     public double preco; 3 usages
6     public int quantidadeEmEstoque; 4 usages
7
8     public void alterarPreco(double novoPreco){ 1 usage new *
9         this.preco = novoPreco;
10    }
11    public void alterarQuantidade(int novaQuantidade){ 1 usage new *
12        this.quantidadeEmEstoque = novaQuantidade;
13    }
14    public String exibirInformacoes(){ 1 usage new *
15        return "O produto " + this.nome + " custa R$" + this.preco + ", e temos o total de " + this.quantidadeEmEstoque + (this.quantidadeEmEstoque == 1 ? "
16        disponivel no estoque.": " disponiveis no estoque.");
17    }
18    public static void exercicio_4() { 1 usage new *
19        Produto pastaDeDente = new Produto();
20        pastaDeDente.nome = "Pasta de Dente";
21        pastaDeDente.preco = 3.50;
22        pastaDeDente.quantidadeEmEstoque = 20;
23        pastaDeDente.alterarQuantidade( novaQuantidade: 19);
24        pastaDeDente.alterarPreco( novoPreco: 4.50);
25        System.out.println(pastaDeDente.exibirInformacoes());
26    }
```



```

3 public class Produto { 6 usages new *
30 public static void exercicio_5() { 1 usage new *
36     System.out.println("preço do produto através do 'getPreco': " + kitkat.getPreco());
37     System.out.println("quantidade do produto através do 'getQuantidadeEmEstoque': " + kitkat.getQuantidadeEmEstoque());
38 }
39
40 public String getNome() { 1 usage new *
41     return nome;
42 }
43
44 public void setNome(String nome) { 1 usage new *
45     this.nome = nome;
46 }
47
48 public double getPreco() { 1 usage new *
49     return preco;
50 }
51
52 public void setPreco(double preco) { 1 usage new *
53     this.preco = preco;
54 }
55
56 public int getQuantidadeEmEstoque() { 1 usage new *
57     return quantidadeEmEstoque;
58 }
59
60 public void setQuantidadeEmEstoque(int quantidadeEmEstoque) { 1 usage new *
61     this.quantidadeEmEstoque = quantidadeEmEstoque;
62 }
63
64 }

```

```
27
30     public static void exercicio_5() { 1 usage new *
31         Produto kitkat = new Produto();
32         kitkat.setNome("KitKat");
33         kitkat.setPreco(4);
34         kitkat.setQuantidadeEmEstoque(30);
35         System.out.println("nome do produto através do 'getNome': " + kitkat.getNome());
36         System.out.println("preço do produto através do 'getPreco': " + kitkat.getPreco());
37         System.out.println("quantidade do produto através do 'getQuantidadeEmEstoque': " + kitkat.getQuantidadeEmEstoque());
38     }
39 }
40
41 Produto > exercicio_5()
42
43 Run Main x
44
45 C:\Program Files\Java\jdk-21\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.5\lib\idea_rt.jar=65396" -Df
46 nome do produto através do 'getNome': KitKat
47 preço do produto através do 'getPreco': 4.0
48 quantidade do produto através do 'getQuantidadeEmEstoque': 30
49
50 Process finished with exit code 0
```


Exercício 6

O uso de construtores facilita a criação do objeto, fazendo a utilização de menos código para instanciá-lo, além de nos obrigar a instanciar o objeto com certas propriedades, no caso eu criei dois construtores para que o meu código dos outros exercícios continuasse funcionando, mas caso eu criasse apenas um construtor com propriedades obrigatórias, os meus outros objetos dariam erro até eu atualizar o meu construtor.

```
3      public class Produto { 8 usages new *
4          public String nome; 5 usages
5          public double preco; 6 usages
6          public int quantidadeEmEstoque; 7 usages
7
8          public Produto() {} 2 usages new *
9          public Produto(String nome, double preco, int quantidadeEmEstoque) {} 1 usage new *
10             this.nome = nome;
11             this.preco = preco;
12             this.quantidadeEmEstoque = quantidadeEmEstoque;
13     }
```

```
46
47     public static void exercicio_6() { 1 usage new *
48         Produto sabonete = new Produto( nome: "sabonete", preco: 2.50, quantidadeEmEstoque: 50);
49         System.out.println(sabonete.exibirInformacoes());
50     }
```

Produto

Run Main x

↑ ↓ ↺ ⌂

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.5\lib\idea_
0 produto sabonete custa R$2.5, e temos o total de 50 disponiveis no estoque.
Process finished with exit code 0
```

Exercício 7

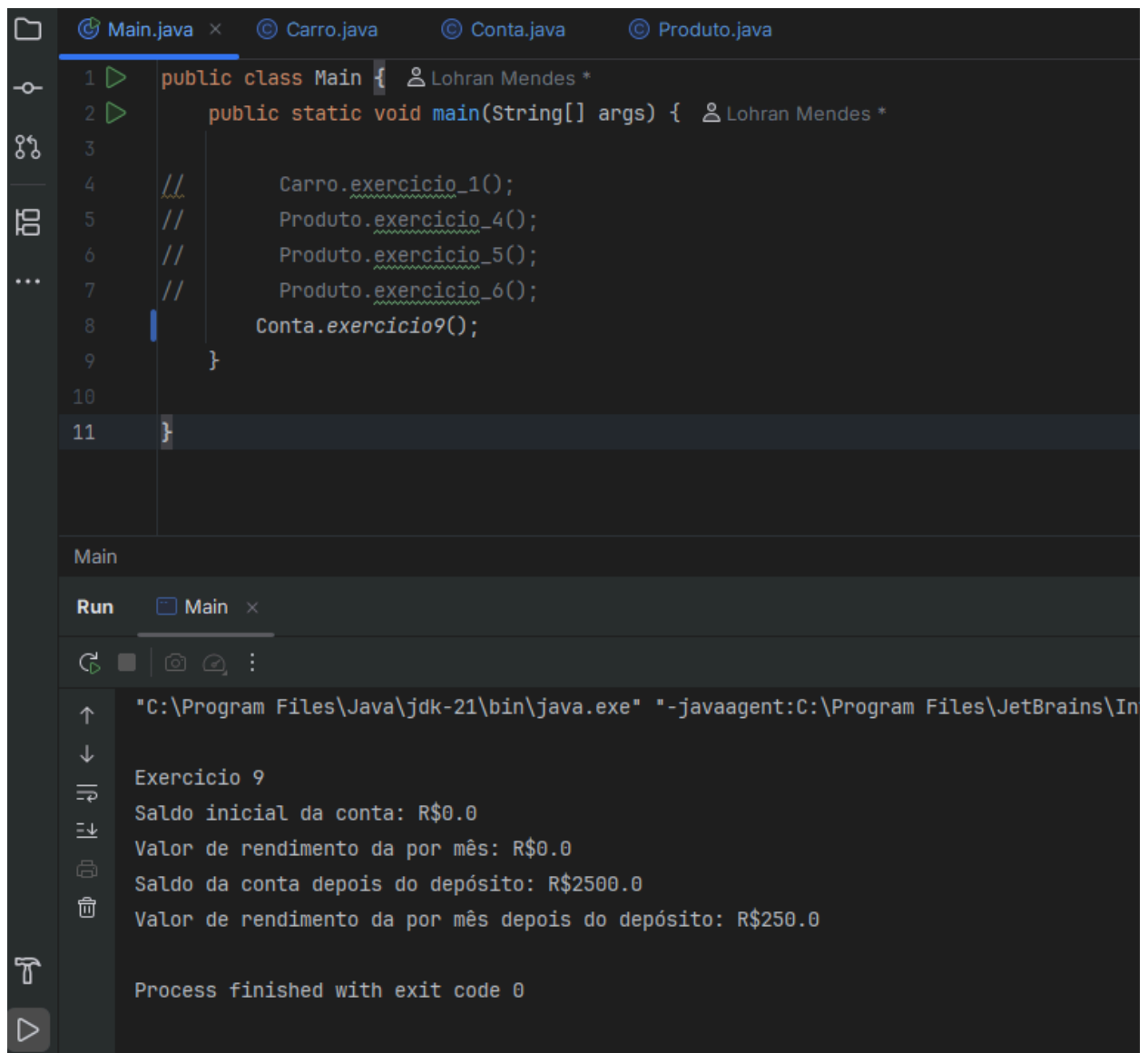
```
© Main.java  © Conta.java ×  © Produto.java
1  public class Conta { no usages new *
2      String titular; no usages
3      String agencia; no usages
4      int numero; no usages
5      double saldo; no usages
6      String dataAbertura; no usages
7
8  }
9  |
```

Exercício 8

```
© Main.java  © Conta.java ×  © Produto.java
1  public class Conta { no usages new *
2      String titular; no usages
3      String agencia; no usages
4      int numero; no usages
5      double saldo; 3 usages
6      String dataAbertura; no usages
7
8      public void saca(double valor){ no usages new *
9          | this.saldo -= valor;
10         | }
11         public void deposita(double valor){ no usages new *
12             | this.saldo += valor;
13             | }
14         public double calculaRendimento(){ no usages new *
15             | return (this.saldo * 0.1);
16             | }
17     }
18     |
```

Exercício 9

```
© Main.java  © Carro.java  © Conta.java  ×  © Produto.java
1  public class Conta { 3 usages  Lohran Mendes *
2      String titular; 1 usage
3      String agencia; 1 usage
4      int numero; 1 usage
5      double saldo; 6 usages
6      String dataAbertura; 1 usage
7
8      public void saca(double valor){ no usages  Lohran Mendes
9          this.saldo -= valor;
10     }
11     public void deposita(double valor){ 1 usage  Lohran Mendes
12         this.saldo += valor;
13     }
14     public double calculaRendimento(){ 2 usages  Lohran Mendes
15         return (this.saldo * 0.1);
16     }
17     public static void exercicio9(){ 1 usage  Lohran Mendes *
18         System.out.println("\nExercicio 9");
19         Conta contaPropria = new Conta();
20         contaPropria.titular = "Maria";
21         contaPropria.saldo = 0;
22         contaPropria.agencia = "agencia 01";
23         contaPropria.numero = 0001;
24         contaPropria.dataAbertura = "08/06/2025";
25         System.out.println("Saldo inicial da conta: R$" + contaPropria.saldo);
26         System.out.println("Valor de rendimento da por mês: R$" + contaPropria.calculaRendimento());
27         contaPropria.deposita(valor: 2500);
28         System.out.println("Saldo da conta depois do depósito: R$" + contaPropria.saldo);
29         System.out.println("Valor de rendimento da por mês depois do depósito: R$" + contaPropria.calculaRendimento());
30     }
31 }
32
```



The image shows a screenshot of an IDE with four tabs: Main.java, Carro.java, Conta.java, and Produto.java. The Main.java tab is active, showing the following code:

```
1 public class Main { Lohran Mendes *
2     public static void main(String[] args) { Lohran Mendes *
3
4         // Carro.exercicio_1();
5         // Produto.exercicio_4();
6         // Produto.exercicio_5();
7         // Produto.exercicio_6();
8         Conta.exercicio9();
9     }
10
11 }
```

Below the code editor, the Run tab is active, showing the execution output:

```
Run Main x
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\In
Exercicio 9
Saldo inicial da conta: R$0.0
Valor de rendimento da por mês: R$0.0
Saldo da conta depois do depósito: R$2500.0
Valor de rendimento da por mês depois do depósito: R$250.0

Process finished with exit code 0
```

Exercício 10

O atributo raio é importante tanto para a classe circulo quanto a esfera porque é a propriedade que define essas formas, sem a qual elas não podem existir matematicamente. Além disso, qualquer tipo de método ou calculo adicionado posteriormente precisa do raio.

```
© Main.java  © Circulo.java ×  © Esfera.java
1  public class Circulo { no usages new *
2      public double raio; no usages
3  }
4  |
```

```
© Main.java  © Circulo.java  © Esfera.java ×
1  public class Esfera { no usages new *
2      public double raio; no usages
3  }
```

Exercício 11

```
© Main.java  © Circulo.java ×  © Esfera.java
1  public class Circulo { no usages new *
2      public double raio; 2 usages
3
4      double calcularArea(){ no usages new *
5          return (raio*raio) * Math.PI;
6      }
7  }
```

```
© Main.java  © Circulo.java  © Esfera.java ×
1  public class Esfera { no usages new *
2      public double raio; 1 usage
3
4      double calcularVolume(){ no usages new *
5          return (4.0 / 3.0) * Math.PI * Math.pow(raio, 3);
6      }
7  }
8
```

Exercício 12

```
11
12 public static void exercicio12() { 1 usage new *
13     System.out.println("\nExercício 12");
14     Circulo circulo = new Circulo();
15     circulo.raio = 3.0;
16     Esfera esfera = new Esfera();
17     esfera.raio = 5.0;
18
19     System.out.println("Área do círculo com " + circulo.raio + " de raio: " + circulo.calcularArea());
20     System.out.println("Volume da esfera com " + esfera.raio + " de raio: " + esfera.calcularVolume());
21 }
22 }
```

Main > exercicio12()

Run Main x

"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.5\lib\idea_

Exercício 12

Área do círculo com 3.0 de raio: 28.274333882308138

Volume da esfera com 5.0 de raio: 523.5987755982989

Process finished with exit code 0