



MAPA – Material de Avaliação Prática da Aprendizagem

Nome: Francisco Willians Da Vera Cruz Silva	R.A 23102201-5
Curso: Superior De Tecnologia Em Análise E Desenvolvimento De Sistemas	
Disciplina: Programação Avançada	

Instruções para Realização da Atividade

1. Revise seu arquivo antes do envio. Certifique-se de que é o arquivo correto, formato correto, se contempla todas as demandas da atividade etc.
2. Após o envio não serão permitidas alterações.
3. Durante a disciplina, procure sanar suas dúvidas pontuais em relação ao conteúdo relacionado à atividade. Porém, não são permitidas correções parciais, ou seja, enviar para que o professor possa fazer uma avaliação prévia e retornar para que o aluno possa ajustar e enviar novamente. Isso não é permitido, pois descharacteriza o processo de avaliação.
4. Ao utilizar quaisquer materiais de pesquisa référencia conforme as normas da ABNT;

Em caso de dúvidas, entre em contato com seu Professor Mediador.

Bons estudos!



AGORA, É COM VOCÊ!

A Programação Orientada a Objetos (POO) é um dos paradigmas mais utilizados no desenvolvimento de software, permitindo a criação de sistemas modulares, reutilizáveis e de fácil manutenção. Em Java, a Programação Orientada a Objetos é baseada em classes, objetos, encapsulamento, herança e polimorfismo, conceitos essenciais para a modelagem de soluções eficientes e escaláveis.

Neste conjunto de atividades, os estudantes praticarão e consolidarão esses fundamentos, avançando progressivamente desde a criação de classes e objetos até a implementação de herança, polimorfismo e interfaces. O objetivo é não apenas compreender a teoria, mas também aplicar os conceitos em situações reais, desenvolvendo habilidades que são fundamentais para o mercado de trabalho.

Referência: OLIVEIRA JUNIOR, E. A.; PEREIRA, R. de L. **Programação Avançada**. Maringá: UniCesumar, 2016.

Classes e Objetos

1 Criando a primeira classe

Crie uma classe chamada Pessoa com os seguintes atributos:

nome (String)

idade (int)

cidade (String)

No método main, instancie um objeto da classe Pessoa e exiba os dados no console.

Métodos e Construtores

2 Classe "Retângulo" com métodos

Crie uma classe Retângulo com os atributos largura e altura e um método:

calcularArea() → Retorna a área do retângulo.

No main, instancie um Retângulo, defina os valores e exiba a área.

3 Conta Bancária (Construtor e Métodos)

Crie a classe ContaBancaria com os atributos:

titular

saldo

Implemente os métodos:

depositar(double valor)

sacar(double valor)

exibirSaldo()

No main, crie um objeto e teste as operações de saque e depósito.

Encapsulamento (Getters e Setters)

4 Classe "Produto" (Encapsulamento)

Crie uma classe Produto com os atributos privados:

nome (String)

preco (double)

Implemente os métodos **getters e setters** para acessar e modificar os atributos.



No main, instancie um objeto Produto, defina os valores e exiba as informações.
Herança e Polimorfismo

5 Criando a classe "Funcionário"

Crie uma classe Funcionário com os atributos:

nome
salario

Crie o método calcularBonus(), que retorna **10% do salário**.

Agora, crie a classe Gerente, que **herda** de Funcionário, e **sobrescreva** calcularBonus() para retornar **20% do salário**.

Teste o polimorfismo no main:

Crie um Funcionário e um Gerente, e exiba o bônus de cada um.

6 Sistema de Veículos (Herança e Polimorfismo)

Crie uma **classe base** chamada Veículo com um método mover().

Agora, crie duas classes-filhas:

Carro → Sobrescreva mover() para exibir "**O carro está se movendo rapidamente**".

Bicicleta → Sobrescreva mover() para exibir "**A bicicleta está se movendo lentamente**".

No main, teste o polimorfismo chamando mover() em objetos Carro e Bicicleta.

Como entregar a atividade:

A atividade deverá ser produzida em um arquivo do tipo TEXTO, conforme TEMPLATE anexado no MATERIAL DA DISCIPLINA, disponibilizado no Studeo, e DEVE ser entregue com a extensão (.PDF). Depois, deve ser anexado no ambiente da Atividade no STUDEO.

Obs.: inserir o print com o código e a execução do projeto.

Dicas para realizar a atividade:

1. Durante as aulas, o professor fornecerá dicas que podem ser utilizadas para a confecção das suas atividades. Assim, é de suma importância participar das aulas ao vivo ou assisti-las posteriormente.
2. Assista às aulas conceituais da disciplina.

Orientações:

Plágios e cópias indevidas serão penalizadas com descontos na nota, podendo chegar a zero.

Não são permitidas correções parciais no decorrer do módulo, pois a interpretação da atividade também faz parte da avaliação.

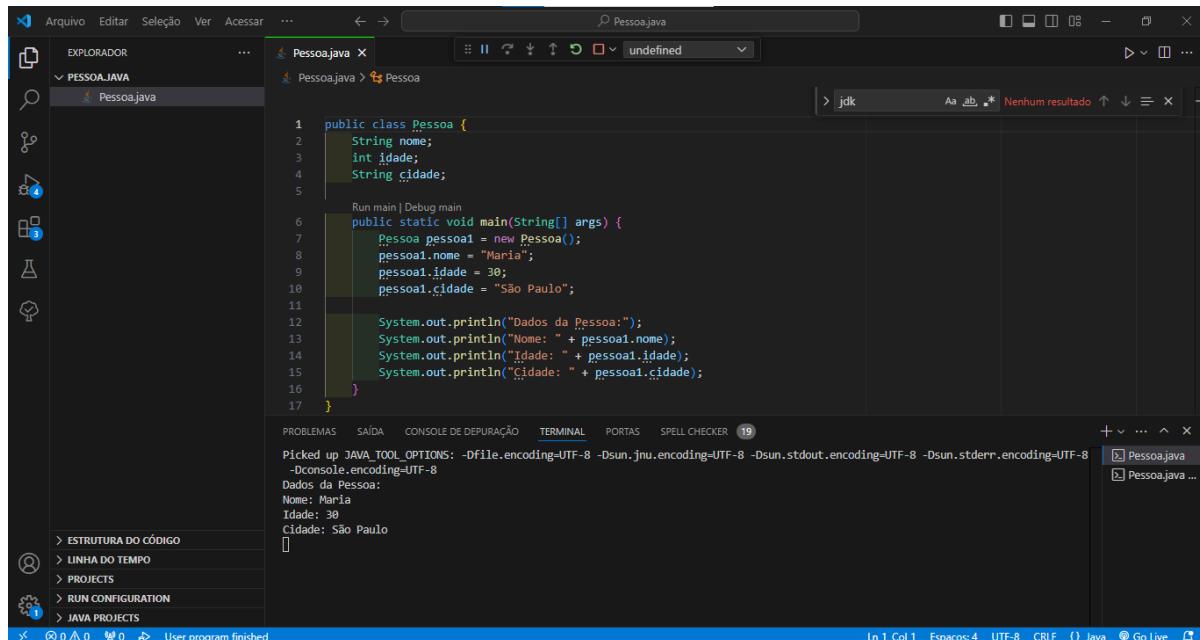
Atenção ao prazo de entrega da atividade. Sugerimos que envie sua atividade antes do prazo final para evitar transtornos e lentidão nos servidores. Evite o envio de atividade em cima do prazo.

Boa atividade!

Coloque sua resposta a continuação:

Print do código e execução:

Resposta 1 – Classe Pessoa

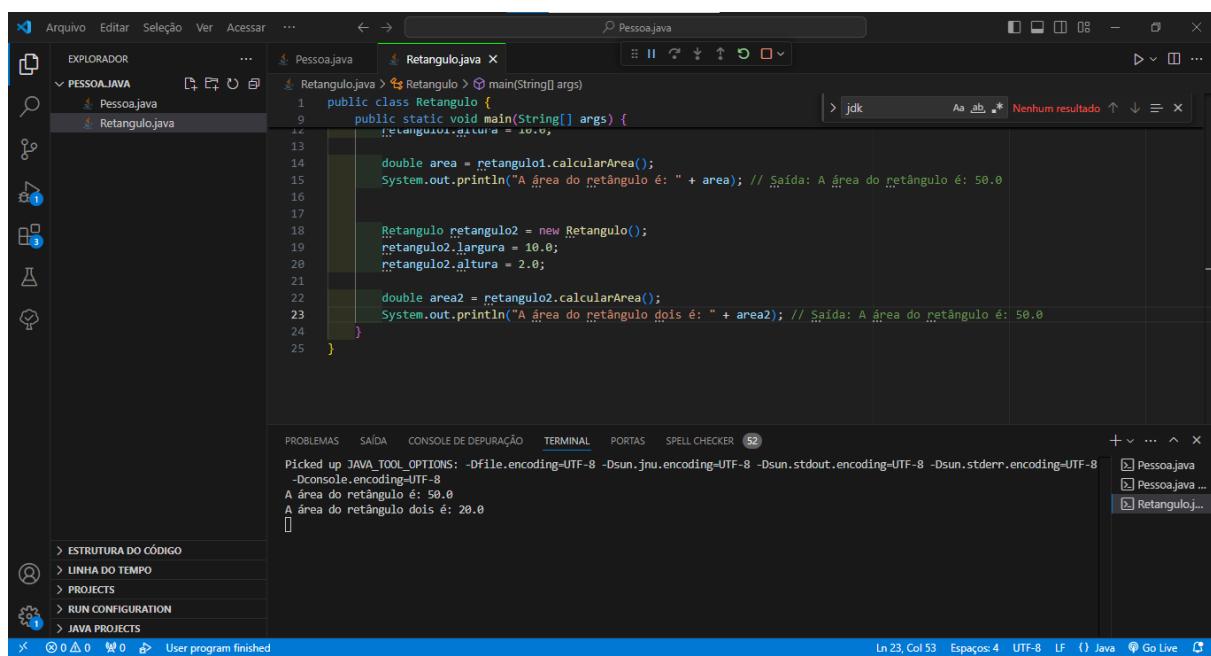


```

Arquivo Editar Seleção Ver Acessar ...
Explorador PESSOA.JAVA Pessoajava
Pessoajava > Pessoajava undefined
1 public class Pessoa {
2     String nome;
3     int idade;
4     String cidade;
5
6     Run main | Debug main
7     public static void main(String[] args) {
8         Pessoa pessoa1 = new Pessoa();
9         pessoa1.nome = "Maria";
10        pessoa1.idade = 30;
11        pessoa1.cidade = "São Paulo";
12
13        System.out.println("Dados da Pessoa:");
14        System.out.println("Nome: " + pessoa1.nome);
15        System.out.println("Idade: " + pessoa1.idade);
16        System.out.println("Cidade: " + pessoa1.cidade);
17    }
}
PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL PORTAS SPELL CHECKER 19
Picked up JAVA_TOOL_OPTIONS: -Dfile.encoding=UTF-8 -Dsun.jnu.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8
-Dconsole.encoding=UTF-8
Dados da Pessoa:
Nome: Maria
Idade: 30
Cidade: São Paulo
+ ... ^ x
Pessoajava.java Pessoajava ...
Ln 1, Col 1 Espaços: 4 UTF-8 CRLF () Java Go Live

```

Resposta 2 – Classe Retângulo



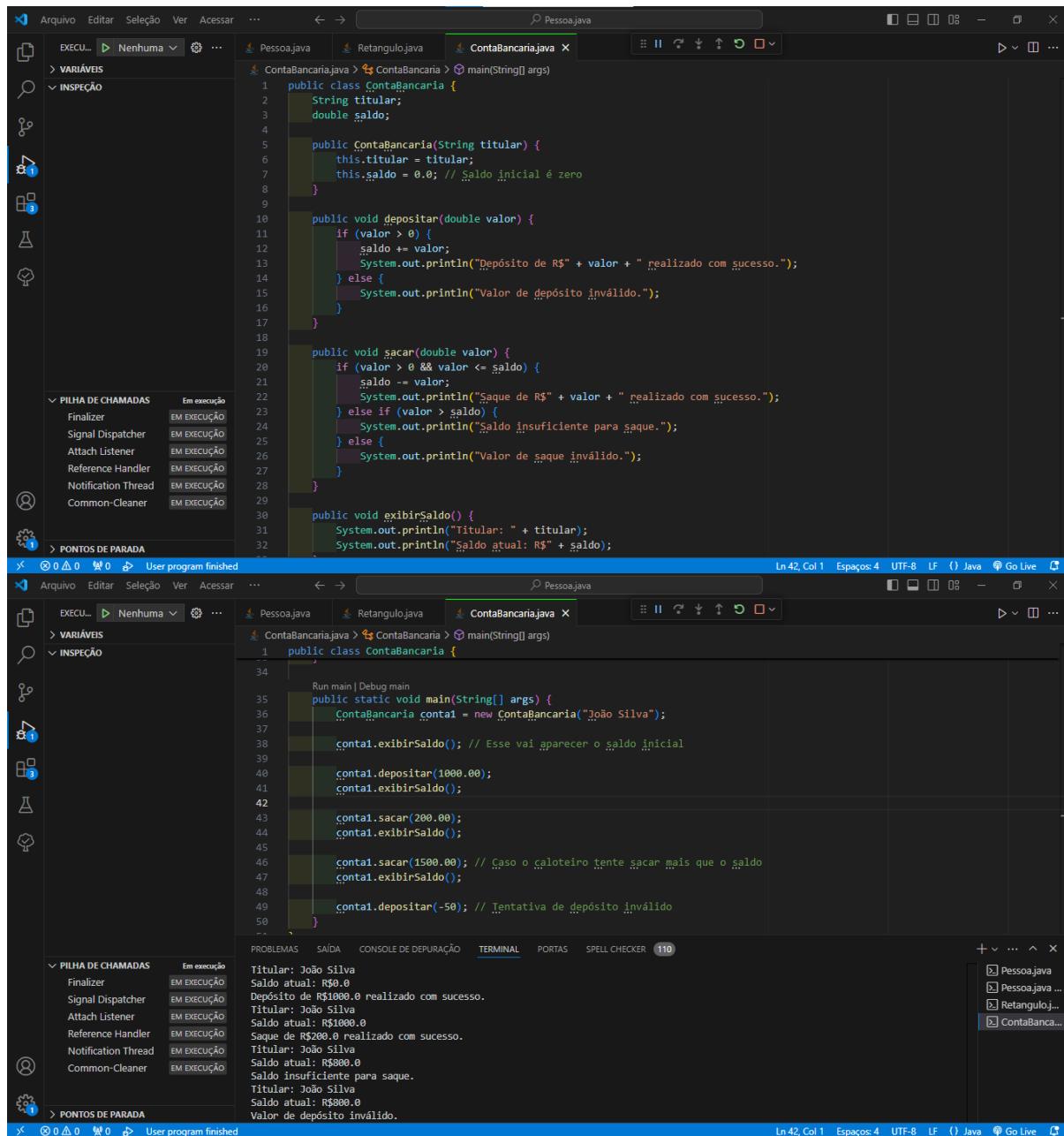
```

Arquivo Editar Seleção Ver Acessar ...
Explorador PESSOA.JAVA Retangulo.java Pessoajava
Retangulo.java > Retângulo main(String[] args)
1 public class Retângulo {
2     public static void main(String[] args) {
3         Retângulo retângulo1 = new Retângulo();
4         double area = retângulo1.calcularArea();
5         System.out.println("A área do retângulo é: " + area); // Saída: A área do retângulo é: 50.0
6
7         Retângulo retângulo2 = new Retângulo();
8         retângulo2.largura = 10.0;
9         retângulo2.altura = 2.0;
10
11        double area2 = retângulo2.calcularArea();
12        System.out.println("A área do retângulo dois é: " + area2); // Saída: A área do retângulo dois é: 20.0
13    }
}
PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL PORTAS SPELL CHECKER 52
Picked up JAVA_TOOL_OPTIONS: -Dfile.encoding=UTF-8 -Dsun.jnu.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8
-Dconsole.encoding=UTF-8
A área do retângulo é: 50.0
A área do retângulo dois é: 20.0
+ ... ^ x
Pessoajava.java Pessoajava ... Retângulo.java ...
Ln 23, Col 53 Espaços: 4 UTF-8 LF () Java Go Live

```



Resposta 3 – Conta Bancária



The screenshot shows two instances of an IDE displaying Java code related to a bank account class.

Top Window (Screenshot 1):

```

public class ContaBancaria {
    String titular;
    double saldo;

    public ContaBancaria(String titular) {
        this.titular = titular;
        this.saldo = 0.0; // Saldo inicial é zero
    }

    public void depositar(double valor) {
        if (valor > 0) {
            saldo += valor;
            System.out.println("Depósito de R$" + valor + " realizado com sucesso.");
        } else {
            System.out.println("Valor de depósito inválido.");
        }
    }

    public void sacar(double valor) {
        if (valor > 0 && valor <= saldo) {
            saldo -= valor;
            System.out.println("Saque de R$" + valor + " realizado com sucesso.");
        } else if (valor > saldo) {
            System.out.println("Saldo insuficiente para saque.");
        } else {
            System.out.println("Valor de saque inválido.");
        }
    }

    public void exibirSaldo() {
        System.out.println("Titular: " + titular);
        System.out.println("Saldo atual: R$" + saldo);
    }
}

```

Bottom Window (Screenshot 2):

```

public static void main(String[] args) {
    ContaBancaria conta1 = new ContaBancaria("João Silva");

    conta1.exibirSaldo(); // Esse vai aparecer o saldo inicial

    conta1.depositar(1000.00);
    conta1.exibirSaldo();

    conta1.sacar(200.00);
    conta1.exibirSaldo();

    conta1.sacar(1500.00); // Caso o caloteiro tente sacar mais que o saldo
    conta1.exibirSaldo();

    conta1.depositar(-50); // Tentativa de depósito inválido
}

```

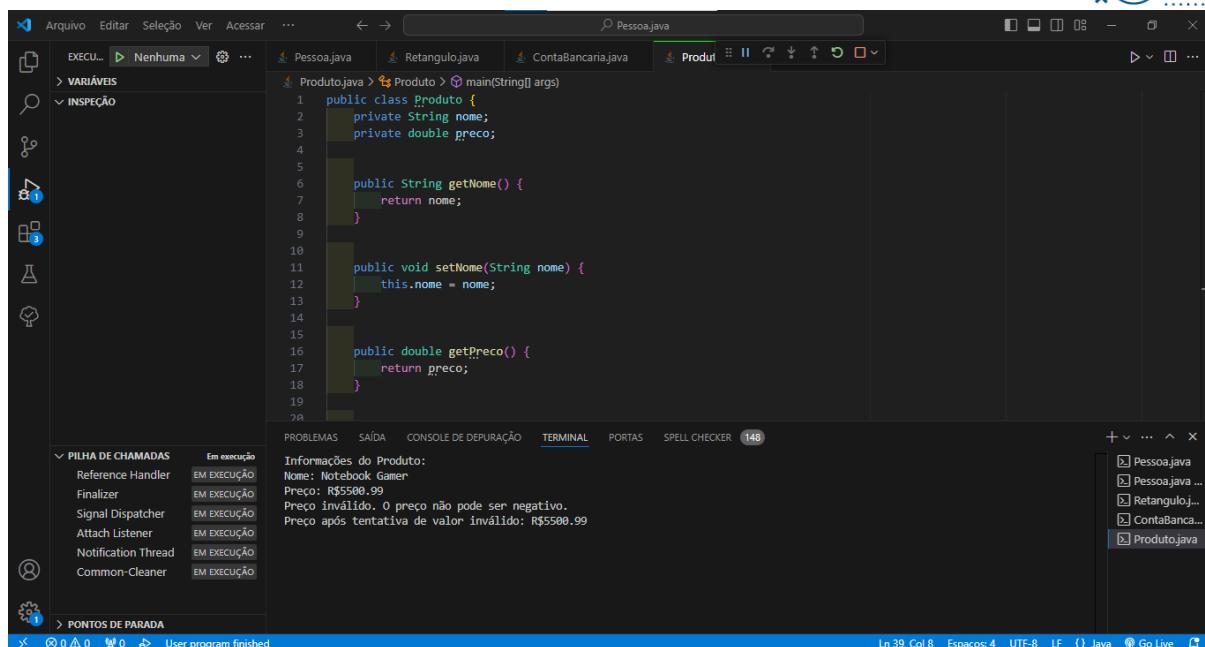
The bottom window shows the execution results:

```

Titular: João Silva
Saldo atual: R$0.0
Depósito de R$1000.00 realizado com sucesso.
Titular: João Silva
Saldo atual: R$1000.0
Saque de R$200.0 realizado com sucesso.
Titular: João Silva
Saldo atual: R$800.0
Saldo insuficiente para saque.
Titular: João Silva
Saldo atual: R$800.0
Valor de depósito inválido.

```

Resposta 4 – Classe Produto com Encapsulamento



Arquivo Editar Seleção Ver Acessar ... ← → 🔍 Pessoajava Retângulo.java ContaBancaria.java Produto.java

VARIÁVEIS INSPEÇÃO

```

1 public class Produto {
2     private String nome;
3     private double preco;
4
5     public String getName() {
6         return nome;
7     }
8
9
10    public void setName(String nome) {
11        this.nome = nome;
12    }
13
14    public double getPreco() {
15        return preco;
16    }
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

```

PILHA DE CHAMADAS Em execução

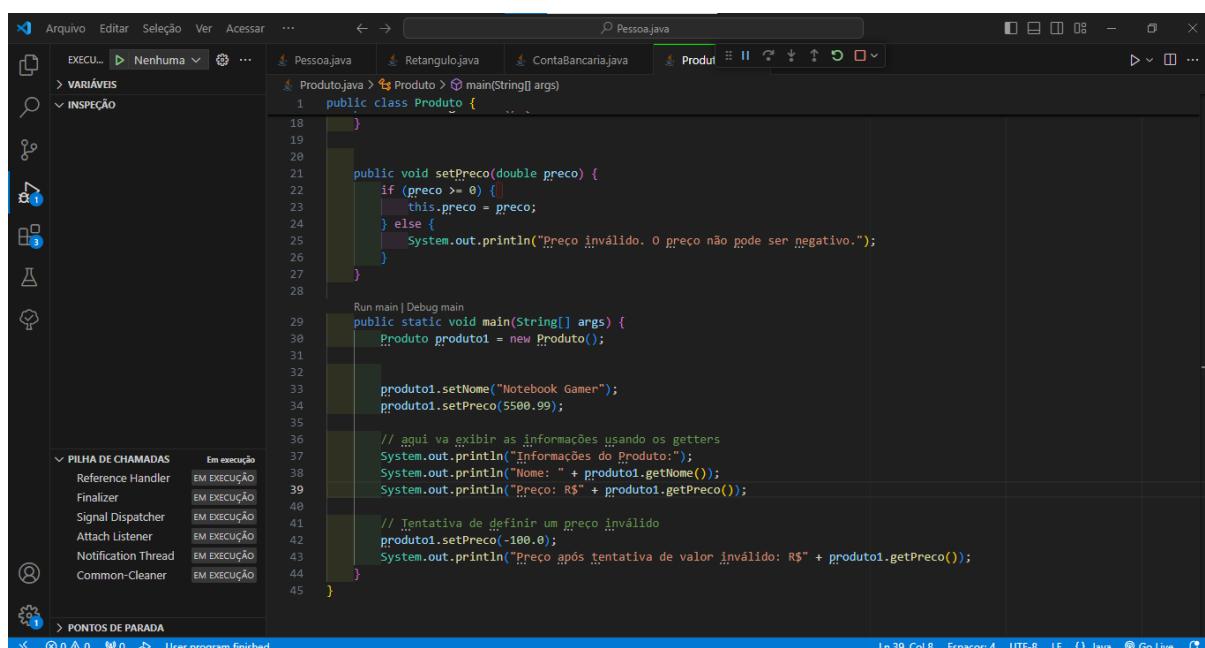
- Reference Handler EM EXECUÇÃO
- Finalizer EM EXECUÇÃO
- Signal Dispatcher EM EXECUÇÃO
- Attach Listener EM EXECUÇÃO
- Notification Thread EM EXECUÇÃO
- Common-Cleaner EM EXECUÇÃO

PONTOS DE PARADA

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL PORTAS SPELL CHECKER 148

Informações do Produto:
 Nome: Notebook Gamer
 Preço: R\$5500.99
 Preço inválido. O preço não pode ser negativo.
 Preço após tentativa de valor inválido: R\$5500.99

Ln 39, Col 8 Espaços: 4 UTF-8 LF { } Java Go Live



Arquivo Editar Seleção Ver Acessar ... ← → 🔍 Pessoajava Retângulo.java ContaBancaria.java Produto.java

VARIÁVEIS INSPEÇÃO

```

1 public class Produto {
2     ...
3
4     public void setPreco(double preco) {
5         if (preco >= 0) {
6             this.preco = preco;
7         } else {
8             System.out.println("Preço inválido. O preço não pode ser negativo.");
9         }
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

```

PILHA DE CHAMADAS Em execução

- Reference Handler EM EXECUÇÃO
- Finalizer EM EXECUÇÃO
- Signal Dispatcher EM EXECUÇÃO
- Attach Listener EM EXECUÇÃO
- Notification Thread EM EXECUÇÃO
- Common-Cleaner EM EXECUÇÃO

PONTOS DE PARADA

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL PORTAS SPELL CHECKER 148

Run main | Debug main

```

public static void main(String[] args) {
    Produto produto1 = new Produto();

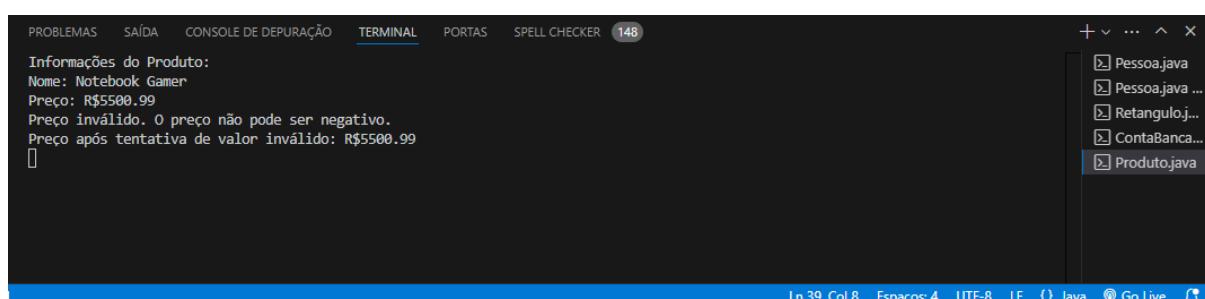
    produto1.setName("Notebook Gamer");
    produto1.setPreco(5500.99);

    // aqui vai exibir as informações usando os getters
    System.out.println("Informações do Produto:");
    System.out.println("Nome: " + produto1.getName());
    System.out.println("Preço: R$" + produto1.getPreco());

    // Tentativa de definir um preço inválido
    produto1.setPreco(-100.0);
    System.out.println("Preço após tentativa de valor inválido: R$" + produto1.getPreco());
}

```

Ln 39, Col 8 Espaços: 4 UTF-8 LF { } Java Go Live

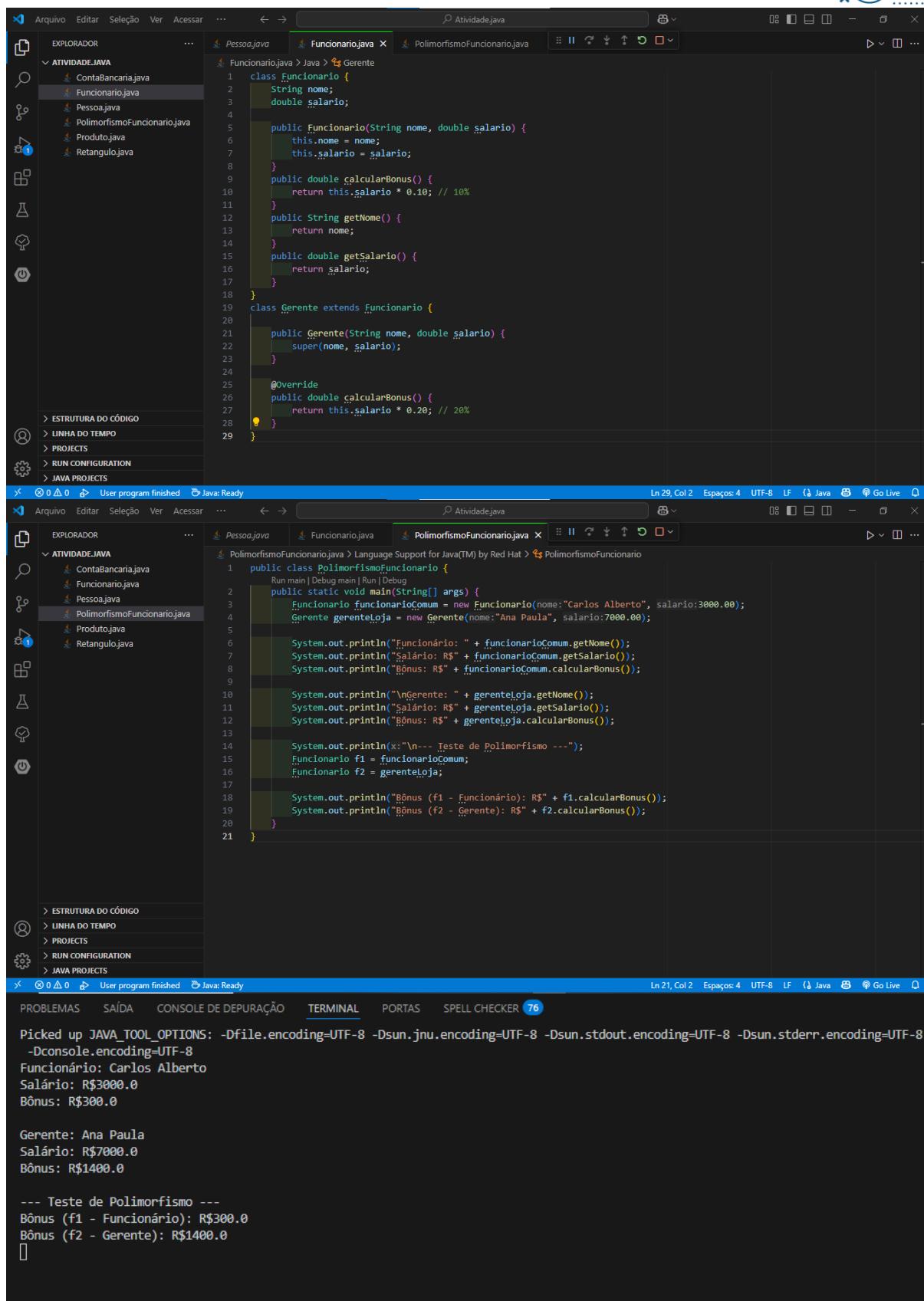


PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL PORTAS SPELL CHECKER 148

Informações do Produto:
 Nome: Notebook Gamer
 Preço: R\$5500.99
 Preço inválido. O preço não pode ser negativo.
 Preço após tentativa de valor inválido: R\$5500.99

Ln 39, Col 8 Espaços: 4 UTF-8 LF { } Java Go Live

Resposta 5 – Funcionário e Gerente (Herança e Polimorfismo)



The screenshot shows three stacked windows of an IDE (likely Eclipse or IntelliJ IDEA) demonstrating Java polymorphism.

Top Window: Shows the code for `Funcionario.java` and `Gerente.java`. The `Funcionario` class has methods `calcularBonus()` and `getNome()`, and a constructor. The `Gerente` class extends `Funcionario` and overrides the `calcularBonus()` method to return 20%.

```
1 class Funcionario {
2     String nome;
3     double salario;
4
5     public Funcionario(String nome, double salario) {
6         this.nome = nome;
7         this.salario = salario;
8     }
9     public double calcularBonus() {
10        return this.salario * 0.10; // 10%
11    }
12    public String getNome() {
13        return nome;
14    }
15    public double getSalario() {
16        return salario;
17    }
18}
19 class Gerente extends Funcionario {
20
21    public Gerente(String nome, double salario) {
22        super(nome, salario);
23    }
24
25    @Override
26    public double calcularBonus() {
27        return this.salario * 0.20; // 20%
28    }
29}
```

Middle Window: Shows the code for `PolimorfismoFuncionario.java`. It creates instances of `FuncionarioComum` and `GerenteLoja`, prints their names, salaries, and bonuses, and then prints the bonuses of `f1` and `f2`.

```
1 public class PolimorfismoFuncionario {
2     public static void main(String[] args) {
3         Funcionario funcionarioComum = new Funcionario(nome:"Carlos Alberto", salario:3000.00);
4         Gerente gerenteLoja = new Gerente(nome:"Ana Paula", salario:7000.00);
5
6         System.out.println("Funcionário: " + funcionarioComum.getNome());
7         System.out.println("Salário: R$" + funcionarioComum.getSalario());
8         System.out.println("Bônus: R$" + funcionarioComum.calcularBonus());
9
10        System.out.println("\nGerente: " + gerenteLoja.getNome());
11        System.out.println("Salário: R$" + gerenteLoja.getSalario());
12        System.out.println("Bônus: R$" + gerenteLoja.calcularBonus());
13
14        System.out.println("\n--- Teste de Polimorfismo ---");
15        Funcionario f1 = funcionarioComum;
16        Funcionario f2 = gerenteLoja;
17
18        System.out.println("Bônus (f1 - Funcionário): R$" + f1.calcularBonus());
19        System.out.println("Bônus (f2 - Gerente): R$" + f2.calcularBonus());
20    }
21}
```

Bottom Window: Shows the terminal output of the program execution. It prints the names, salaries, and bonuses for both employees, followed by the polymorphic test results.

```
Picked up JAVA_TOOL_OPTIONS: -Dfile.encoding=UTF-8 -Dsun.jnu.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8
-Dconsole.encoding=UTF-8
Funcionário: Carlos Alberto
Salário: R$3000.0
Bônus: R$300.0

Gerente: Ana Paula
Salário: R$7000.0
Bônus: R$1400.0

--- Teste de Polimorfismo ---
Bônus (f1 - Funcionário): R$300.0
Bônus (f2 - Gerente): R$1400.0
```

Resposta 6 – Sistema de Veículos (Herança e Polimorfismo)

Atividade.java

```

EXECUT... Nenhuma ... Pessoajava Funcionarijava PolimorfismoFuncionario.java ... Atividadejava
  VARIÁVEIS
  INSPEÇÃO
  PILHA DE CHAMADAS Em execução
    Finalizer EM EXECUÇÃO
    Signal Dispatcher EM EXECUÇÃO
    Reference Handler EM EXECUÇÃO
    Attach Listener EM EXECUÇÃO
    Notification Thread EM EXECUÇÃO
    Common-Cleaner EM EXECUÇÃO
  PONTOS DE PARADA
    Uncaught Exceptions
    Caught Exceptions
  User program finished Java: Ready
Arquivo Editar Seleção Ver Acessar ... Atividadejava
  VARIÁVEIS
  INSPEÇÃO
  PILHA DE CHAMADAS Em execução
    Finalizer EM EXECUÇÃO
    Signal Dispatcher EM EXECUÇÃO
    Reference Handler EM EXECUÇÃO
    Attach Listener EM EXECUÇÃO
    Notification Thread EM EXECUÇÃO
    Common-Cleaner EM EXECUÇÃO
  PONTOS DE PARADA
    Uncaught Exceptions
    Caught Exceptions
  User program finished Java: Ready
  
```

PolimorfismoVeiculo.java

```

1  class Veiculo {
2      public void mover() {
3          System.out.println("O veículo está se movendo.");
4      }
5  }
6  class Carro extends Veiculo {
7      @Override
8      public void mover() {
9          System.out.println("O carro está se movendo rapidamente.");
10 }
11 }
12 class Bicicleta extends Veiculo {
13     @Override
14     public void mover() {
15         System.out.println("A bicicleta está se movendo lentamente.");
16     }
17 }
18 public class PolimorfismoVeiculo {
19     public static void main(String[] args) {
20         Veiculo meuVeiculoGenerico = new Veiculo();
21         Veiculo meuCarro = new Carro();
22         Veiculo minhaBicicleta = new Bicicleta();
23
24         System.out.println("---Veículo Gênero---");
25         meuVeiculoGenerico.mover();
26
27         System.out.println("\n---Carro---");
28         meuCarro.mover();
29
30         System.out.println("\n---Bicicleta---");
31         minhaBicicleta.mover();
32     }
33
34     System.out.println("\n--- Demonstração Array de Veículos ---");
35     Veiculo[] veiculos = new Veiculo[3];
36     veiculos[0] = new Veiculo();
37     veiculos[1] = new Carro();
38     veiculos[2] = new Bicicleta();
39
40     for (Veiculo v : veiculos) {
41         v.mover();
42     }
43
44 }
  
```

PolimorfismoVeiculo.java

```

18  public class PolimorfismoVeiculo {
19      public static void main(String[] args) {
31
32
33     System.out.println("\n--- Demonstração Array de Veículos ---");
34     Veiculo[] veiculos = new Veiculo[3];
35     veiculos[0] = new Veiculo();
36     veiculos[1] = new Carro();
37     veiculos[2] = new Bicicleta();
38
39     for (Veiculo v : veiculos) {
40         v.mover();
41     }
42
43
44 }
  
```

Ln 23, Col 1 Espaços: 4 UTF-8 LF à Java Go Live



```
PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL PORTAS SPELL CHECKER 126

Picked up JAVA_TOOL_OPTIONS: -Dfile.encoding=UTF-8 -Dsun.jnu.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8
-Dconsole.encoding=UTF-8
---Veículo Genérico ---
O veículo está se movendo.

---Carro ---
O carro está se movendo rapidamente.

---Bicicleta ---
A bicicleta está se movendo lentamente.

--- Demonstração Array de Veículos ---
O veículo está se movendo.
O carro está se movendo rapidamente.
A bicicleta está se movendo lentamente.
[]

Java: Ready
```

Ln 23, Col 1 Espaços: 4 UTF-8 LF { Java

Importante:

- Certifique-se de que todos os prints estejam legíveis e mostrem a execução correta do projeto.
- O arquivo final deve ser exportado em formato **.PDF**, conforme orientações da disciplina.