



MAPA – Material de Avaliação Prática da Aprendizagem

Acadêmico: Jonatha de Albuquerque Lopes	R.A. 24512555-5
Curso: SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA INTERNET	
Disciplina: PROGRAMAÇÃO ORIENTADA A OBJETOS	
Valor da atividade: 3.5	Prazo: 26/10/2025

Instruções para Realização da Atividade

1. Todos os campos acima deverão ser devidamente preenchidos;
2. É obrigatória a utilização deste formulário para a realização do MAPA.
3. Esta é uma atividade individual. Caso identificado cópia de colegas, o trabalho de ambos sofrerá decréscimo de nota.
4. Utilizando este formulário, realize sua atividade, salve em seu computador, renomeie e envie em forma de anexo.
5. Formatação exigida para esta atividade: documento Word, Fonte Arial ou Times New Roman tamanho 12, Espaçamento entre linhas 1,5, texto justificado.
6. Ao utilizar quaisquer materiais de pesquisa faça a referência conforme as normas da ABNT;
7. Critérios de avaliação: Utilização deste template (Formulário Padrão); Atendimento ao Tema; Constituição dos argumentos e organização das Ideias; Correção Gramatical e atendimento às normas ABNT.
8. Procure argumentar de forma clara e objetiva, de acordo com o conteúdo da disciplina.

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

Bons estudos!

AGORA É COM VOCÊ!

Você foi contratado como desenvolvedor para criar um sistema acadêmico para a Unicesumar, uma universidade fictícia que deseja digitalizar o registro de alunos e professores. Ao longo da atividade, você irá construir classes, criar objetos e aplicar conceitos de Programação Orientada a Objetos (POO) em Java. Para tornar a experiência mais real, você utilizará seus próprios dados para preencher os objetos e testar os métodos.

OBJETIVOS DA APRENDIZAGEM

Ao concluir esta atividade, você deverá ser capaz de:

- Criar **classes e objetos** em Java.
- Aplicar **encapsulamento** usando atributos privados e métodos *getters/setters*.
- Implementar **herança**, relacionando classes base e derivadas.
- Demonstrar **polimorfismo** por meio de sobrescrita de métodos.
- Implementar **interfaces** e utilizar métodos abstratos.
- Relacionar conceitos de POO com situações do mundo real, personalizando os exemplos com seus próprios dados.

INSTRUÇÕES DA ATIVIDADE

- Todas as questões **devem ser implementadas em Java, em um único arquivo** por questão.
- **Use seus próprios dados** (nome, matrícula, curso) ao criar objetos.
- **Documente o código com comentários** explicando o que cada trecho faz.
- Teste todas as funcionalidades no método *main* antes de entregar.
- A atividade **deve ser entregue com códigos funcionando e organizados**.

Questão 1 – Classe e Objeto

- Crie uma classe Aluno com os atributos nome e matrícula.
- No método *main*, crie um objeto Aluno com seus próprios dados e exiba no console:

Nome: seu nome

Matrícula: número de matrícula fictício

Questão 2 – Encapsulamento

- Modifique a classe Aluno para que os atributos sejam privados.
- Crie *getters* e *setters* para acessar e alterar os dados.

- No *main*, altere, por exemplo, seu **nome** usando o setter e exiba novamente usando o *getter*.

Questão 3 – Herança

- Crie uma classe Pessoa com o atributo nome.
- Faça Aluno herdar de Pessoa e adicione o atributo curso.
- Crie a classe Professor com o atributo disciplina.
- No *main*, crie:
 - Um objeto Aluno com seus dados
 - Um objeto Professor com dados fictícios
 - Exiba todos os atributos de ambos


Questão 4 – Polimorfismo


- Na classe Pessoa, crie o método apresentar().
- Sobrescreva esse método em Aluno e Professor, retornando apresentações personalizadas.
- No *main*, chame apresentar() para seu objeto Aluno e para o objeto Professor, demonstrando polimorfismo.

Questão 5 – Interface

- Crie uma interface Avaliado com o método avaliarDesempenho().
- Faça a classe Aluno implementar a interface, retornando uma mensagem personalizada, por exemplo: "*Aluno [seu nome] obteve ótimo desempenho!*"
- No *main*, invoque esse método para o seu objeto Aluno.



	Como devo entregar esta atividade
<p>Você deverá implementar o sistema em Java, atendendo as questões apresentadas no enunciado. Deve colar neste <i>template</i> o print dos códigos e o print do programa em execução. Após responder, você deverá salvar esse <i>template</i> em PDF e anexar no campo disponível na atividade. Clicando em "Responder", confira se o arquivo que está anexando é exatamente o arquivo correto, depois clique em "Finalizar" para enviar. Por fim, verifique se o arquivo aparece anexado na atividade.</p>	

	ATENÇÃO
<ul style="list-style-type: none">- Um vídeo explicativo da atividade MAPA está disponível para ajudá-lo nesse processo de criação e desenvolvimento. Você poderá acessar em Fórum Informativo intitulado "VÍDEOS EXPLICANDO AS ATIVIDADES MAPA E AE1" ou em "ARQUIVOS > Material da Disciplina".- Durante as aulas ao vivo, o professor fornecerá dicas que podem ser utilizadas para a confecção das suas atividades. Assim, é de suma importância participar da aula ao vivo, ou assistir às gravações posteriormente.- Plágios e cópias indevidas serão penalizados 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 com pelo menos uma semana de antecedência para evitar transtornos e lentidão nos servidores. Evite o envio de atividade em cima do prazo.	

Boa atividade!

RESPOSTA: <<insira sua resolução a partir deste ponto>>

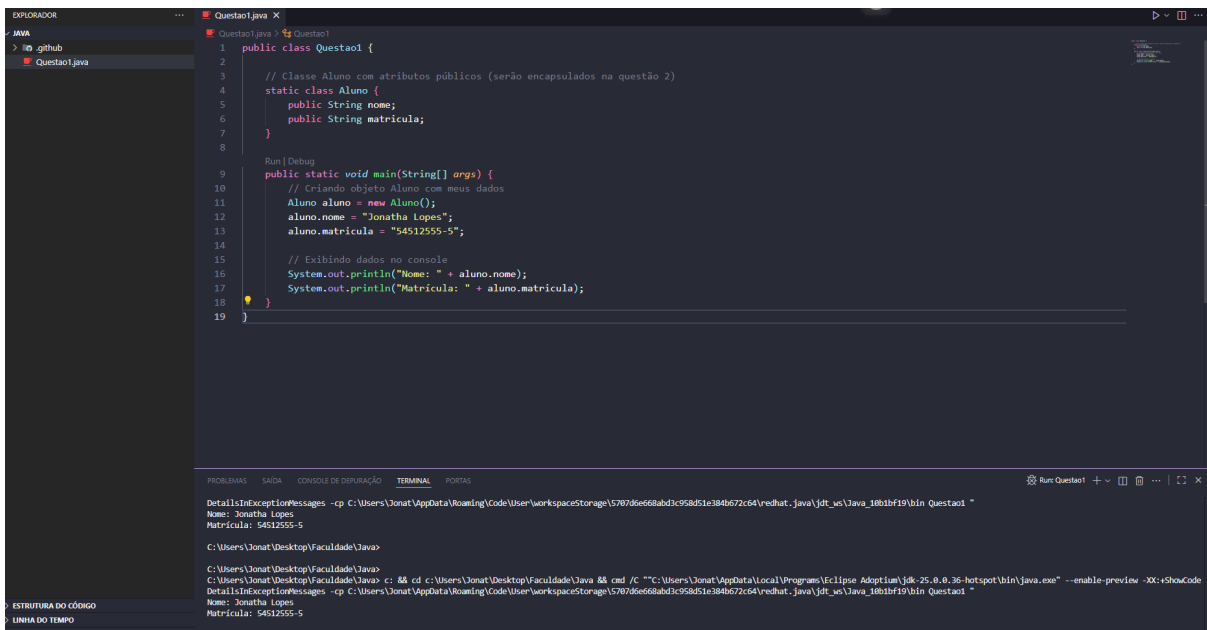
Questão 1

```
public class Questao1 {  
  
    // Classe Aluno  
    static class Aluno {  
        public String nome;  
        public String matricula;  
    }  
  
    public static void main(String[] args) {  
        // Criando objeto Aluno com meus dados
```



```
Aluno aluno = new Aluno();
aluno.nome = "Jonatha Lopes";
aluno.matricula = "54512555-5";

// Exibindo dados no console
System.out.println("Nome: " + aluno.nome);
System.out.println("Matrícula: " + aluno.matricula);
}
}
```



```
public class Questao1 {
    // Classe Aluno com atributos públicos (serão encapsulados na questão 2)
    static class Aluno {
        public String nome;
        public String matricula;
    }

    public static void main(String[] args) {
        // Criando objeto Aluno com meus dados
        Aluno aluno = new Aluno();
        aluno.nome = "Jonatha Lopes";
        aluno.matricula = "54512555-5";

        // Exibindo dados no console
        System.out.println("Nome: " + aluno.nome);
        System.out.println("Matrícula: " + aluno.matricula);
    }
}
```

Run Questao1

Details: NullPointerException - cp C:\Users\Jonat\AppData\Roaming\Code\User\workspaceStorage\5707d6e68abd3c958d51e384b672c64\vedhut_java\jdk_us\Java_10b1bf19\bin Questao1

Nome: Jonatha Lopes
Matrícula: 54512555-5

C:\Users\Jonat\Desktop\Faculdade\Java>

C:\Users\Jonat\Desktop\Faculdade\Java> cd c:\Users\Jonat\Desktop\Faculdade\Java && cmd /c "C:\Users\Jonat\AppData\Local\Programs\Eclipse Adoptium\jdk-25.0.0-hotspot\bin\java.exe" --enable-preview -XX:+ShowCodeDetails -Xmx1024m -cp C:\Users\Jonat\AppData\Roaming\Code\User\workspaceStorage\5707d6e68abd3c958d51e384b672c64\vedhut_java\jdk_us\Java_10b1bf19\bin Questao1

Nome: Jonatha Lopes
Matrícula: 54512555-5

Questao 2

// Questão 2 - Encapsulamento

```
public class Questao2 {
```

// Classe Aluno com atributos privados

```
static class Aluno {
```

```
    private String nome;
```

```
    private String matricula;
```

// Getter e Setter para nome



```
public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

// Getter e Setter para matrícula
public String getMatricula() {
    return matricula;
}

public void setMatricula(String matricula) {
    this.matricula = matricula;
}

}

public static void main(String[] args) {
    // Criando o objeto aluno
    Aluno aluno = new Aluno();

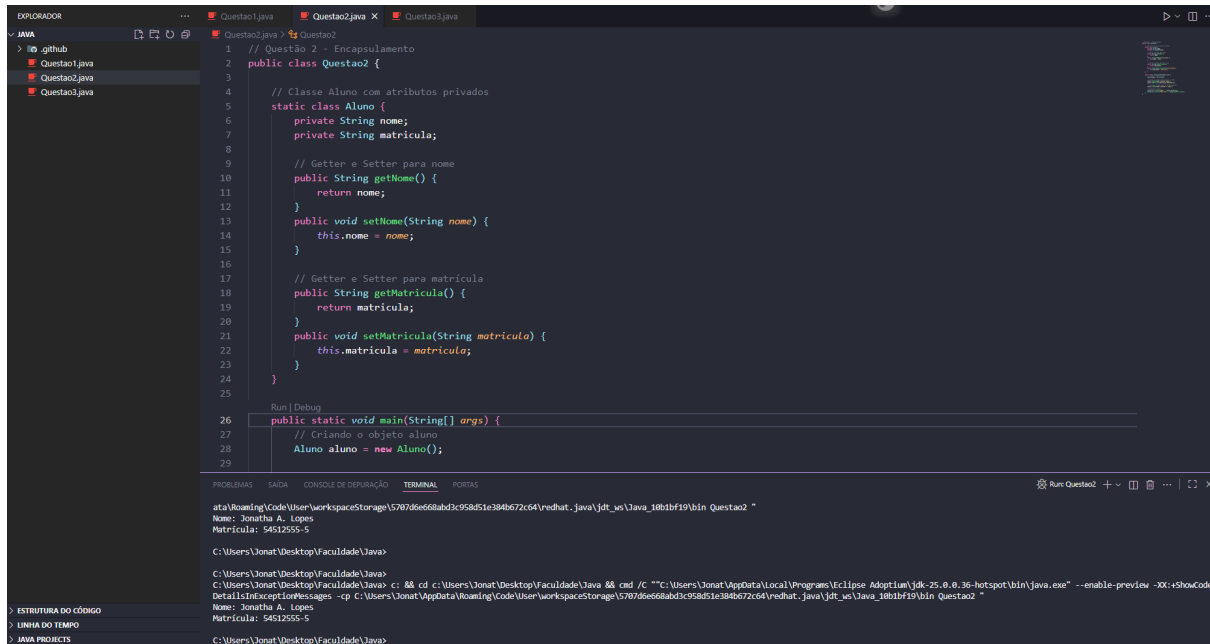
    // Usando os setters para inserir os dados
    aluno.setNome("Jonatha Lopes");
    aluno.setMatricula("54512555-5");

    // Alterando o nome (teste do encapsulamento)
    aluno.setNome("Jonatha A. Lopes");

    // Exibindo os dados
    System.out.println("Nome: " + aluno.getNome());
    System.out.println("Matrícula: " + aluno.getMatricula());
}
}
```



Print questão 2



```

1 // Questão 2 - Encapsulamento
2 public class Questao2 {
3
4     // Classe Aluno com atributos privados
5     static class Aluno {
6         private String nome;
7         private String matricula;
8
9         // Getter e Setter para nome
10        public String getNome() {
11            return nome;
12        }
13        public void setNome(String nome) {
14            this.nome = nome;
15        }
16
17        // Getter e Setter para matricula
18        public String getMatricula() {
19            return matricula;
20        }
21        public void setMatricula(String matricula) {
22            this.matricula = matricula;
23        }
24    }
25
26    public static void main(String[] args) {
27        // Criando o objeto aluno
28        Aluno aluno = new Aluno();
29    }

```

Run | Debug

PROBLEMAS SAÍDA CONSOLE DE DEBURAÇÃO TERMINAL PORTAS

Run: Questao2

Nome: Jonatha A. Lopes
Matricula: 54512555-5

C:\Users\Jonat\Desktop\Faculdade\Java>

C:\Users\Jonat\Desktop\Faculdade\Java>

C:\Users\Jonat\Desktop\Faculdade\Java> c: && cd c:\Users\Jonat\Desktop\Faculdade\Java && cmd /c ""C:\Users\Jonat\AppData\Local\Programs\Eclipse Adoptium\jdk-25.0.0-hotspot\bin\java.exe" --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp C:\Users\Jonat\AppData\Roaming\Code\User\workspaceStorage\5707d6e668abd3c958d51e384672c64\redhat.java\jdt_ws\Java_10b1bf19\bin Questao2 "

Nome: Jonatha A. Lopes
Matricula: 54512555-5

C:\Users\Jonat\Desktop\Faculdade\Java>

Questao 3

// Questão 3 - Herança

```
public class Questao3 {
```

```
    // Classe base
```

```
    static class Pessoa {
        protected String nome; //
    }
```

```
    // Classe Aluno herda de Pessoa
```

```
    static class Aluno extends Pessoa {
        private String curso;

        public String getCurso() {
```



```
        return curso;
    }

    public void setCurso(String curso) {
        this.curso = curso;
    }
}

// Classe Professor
static class Professor extends Pessoa {
    private String disciplina;

    public String getDisciplina() {
        return disciplina;
    }

    public void setDisciplina(String disciplina) {
        this.disciplina = disciplina;
    }
}

public static void main(String[] args) {
    // Criando objeto Aluno
    Aluno aluno = new Aluno();
    aluno.nome = "Jonatha Lopes";
    aluno.setCurso("Análise e Desenvolvimento de Sistemas");

    // Criando objeto Professor
    Professor prof = new Professor();
    prof.nome = "Carlos Mendes";
    prof.setDisciplina("Programação Orientada a Objetos");
}
```




```
// Exibindo informações

System.out.println("=== Dados do Aluno ===");

System.out.println("Nome: " + aluno.nome);

System.out.println("Curso: " + aluno.getCurso());


System.out.println("\n=== Dados do Professor ===");

System.out.println("Nome: " + prof.nome);

System.out.println("Disciplina: " + prof.getDisciplina());

}

}
```

Print Questao 3

The screenshot shows an IDE with a project named 'Questao3'. The main editor displays the following Java code:

```
1 // Questão 3 - Herança
2 public class Questao3 {
3
4     // Classe base
5     static class Pessoa {
6         protected String nome; //
7     }
8
9     // Classe Aluno herda de Pessoa
10    static class Aluno extends Pessoa {
11        private String curso;
12
13        public String getCurso() {
14            return curso;
15        }
16
17        public void setCurso(String curso) {
18            this.curso = curso;
19        }
20    }
21
22    // Classe Professor
23    static class Professor extends Pessoa {
24        private String disciplina;
25
26        public String getDisciplina() {
27            return disciplina;
28        }
29    }
30 }
```

The terminal output shows the execution of the program:

```
C:\Users\Jonat\Desktop\Faculdade\Java>
C:\Users\Jonat\Desktop\Faculdade\Java> c: cd c:\Users\Jonat\Desktop\Faculdade\Java && cd /C "C:\Users\Jonat\AppData\Local\Programs\Eclipse\Adoptium\jdk-25.0.0-hotspot\bin\java.exe" --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp C:\Users\Jonat\AppData\Local\Programs\Eclipse\Adoptium\jdk-25.0.0-hotspot\bin\java\jdt_ws\Java_10b1bf19\bin Questao3
=== Dados do Aluno ===
Nome: Jonatha Lopes
Curso: Análise e Desenvolvimento de Sistemas

--- Dados do Professor ---
Nome: Carlos Mendes
Disciplina: Programação Orientada a Objetos

C:\Users\Jonat\Desktop\Faculdade\Java>
```

Questao 4

// Questão 4 - Polimorfismo



```
public class Questao4 {

    // Superclasse Pessoa
    static class Pessoa {
        protected String nome;

        // Método
        public String apresentar() {
            return "Olá, meu nome é " + nome;
        }
    }

    // Classe Aluno
    static class Aluno extends Pessoa {
        private String curso;

        public Aluno(String nome, String curso) {
            this.nome = nome;
            this.curso = curso;
        }

        @Override
        public String apresentar() {
            return "Sou o aluno " + nome + " do curso de " + curso + ".";
        }
    }

    // Classe Professor
    static class Professor extends Pessoa {
        private String disciplina;

        public Professor(String nome, String disciplina) {
```

```
this.nome = nome;
this.disciplina = disciplina;
}

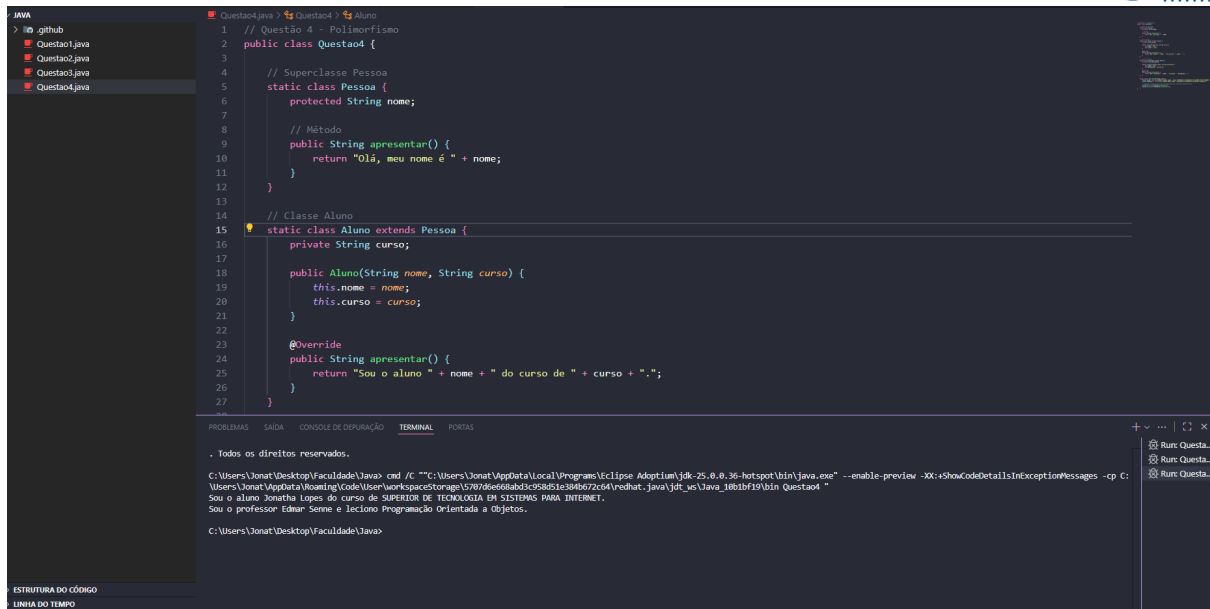
@Override
public String apresentar() {
    return "Sou o professor " + nome + " e leciono " + disciplina + ".";
}

}

public static void main(String[] args) {
    Pessoa aluno = new Aluno("Jonatha Lopes", "SUPERIOR DE TECNOLOGIA
EM SISTEMAS PARA INTERNET");
    Pessoa professor = new Professor("Edmar Senne", "Programação Orientada a
Objetos");

    // Chamando o mesmo método, mas com comportamentos diferentes
    (polimorfismo)
    System.out.println(aluno.apresentar());
    System.out.println(professor.apresentar());
}
}
```

Print Questao 4



```
1 // Questão 4 - Polimorfismo
2 public class Questao4 {
3
4     // Superclasse Pessoa
5     static class Pessoa {
6         protected String nome;
7
8         // Método
9         public String apresentar() {
10             return "Olá, meu nome é " + nome;
11         }
12     }
13
14     // Classe Aluno
15     static class Aluno extends Pessoa {
16         private String curso;
17
18         public Aluno(String nome, String curso) {
19             this.nome = nome;
20             this.curso = curso;
21         }
22
23         @Override
24         public String apresentar() {
25             return "Sou o aluno " + nome + " do curso de " + curso + ".";
26         }
27     }
28 }
```

PROBLEMAS SAÍDA CONSOLE DE DEBURAÇÃO TERMINAL PORTAS

. Todos os direitos reservados.

C:\Users\Jonat\Desktop\Faculdade\Java> cmd /C ""C:\Users\Jonat\AppData\Local\Programs\Eclipse Adoptium\jdk-25.0.0.36-hotspot\bin\java.exe" --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp C:\Users\Jonat\AppData\Roaming\Code\User\workspaceStorage\570796e668abd3c958d5e384b672c64\redhat.java\jdt_ws\Java_10b1bf19\bin Questao4

Sou o aluno Jonatha Lopes do curso de SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA INTERNET.

Sou o professor Edmar Senne e leciono programação Orientada a Objetos.

C:\Users\Jonat\Desktop\Faculdade\Java>

Questao 5

// Questão 5 - Interface

```
public class Questao5 {
```

```
    // Interface
```

```
    interface Avaliado {
```

```
        String avaliarDesempenho();
```

```
    }
```

```
    // Classe Aluno
```

```
    static class Aluno implements Avaliado {
```

```
        private String nome;
```

```
        public Aluno(String nome) {
```

```
            this.nome = nome;
```

```
        }
```



@Override

```
public String avaliarDesempenho() {
    return "Aluno " + nome + " obteve ótimas notas!";
}
}
```

```
public static void main(String[] args) {
    Aluno aluno = new Aluno("Jonatha Lopes");
    System.out.println(aluno.avaliarDesempenho());
}
}
```

The screenshot shows an IDE with the following content:

```
1 // Questão 5 - Interface
2 public class Questao5 {
3
4     // Interface
5     interface Avaliado {
6         String avaliarDesempenho();
7     }
8
9     // Classe Aluno
10    static class Aluno implements Avaliado {
11        private String nome;
12
13        public Aluno(String nome) {
14            this.nome = nome;
15        }
16
17        @Override
18        public String avaliarDesempenho() {
19            return "Aluno " + nome + " obteve ótimas notas!";
20        }
21    }
22
23    Run | Debug
24    public static void main(String[] args) {
25        Aluno aluno = new Aluno(nome:"Jonatha Lopes");
26        System.out.println(aluno.avaliarDesempenho());
27    }
28 }
```

The terminal output shows the command to run the program and the resulting output:

```
C:\Users\Jonat\Desktop\Faculdade\Java>
C:\Users\Jonat\Desktop\Faculdade\Java> cd c:\Users\Jonat\Desktop\Faculdade\Java && cd /C ""C:\Users\Jonat\AppData\Local\Programs\Eclipse Adoptium\jdk-25.0.6-hotspot\bin\java.exe" --enable-preview -XX:ShowCodeDetailsInExceptionMessages -cp C:\Users\Jonat\AppData\Roaming\Code\User\workspaceStorage\578796e668abd3c958d51e384672c64\redhat-java\jdk_w\Java_18b1bf19\bin Questao5
Aluno Jonatha Lopes obteve ótimas notas!
C:\Users\Jonat\Desktop\Faculdade\Java>
```