



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

Você deverá **implementar o sistema em Java**, atendendo as questões apresentadas no enunciado. **Deve colar neste template o print dos códigos e o print do programa em execução**. Após responder, você deverá salvar esse *template* 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.



ATENÇÃO

- 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
```



```

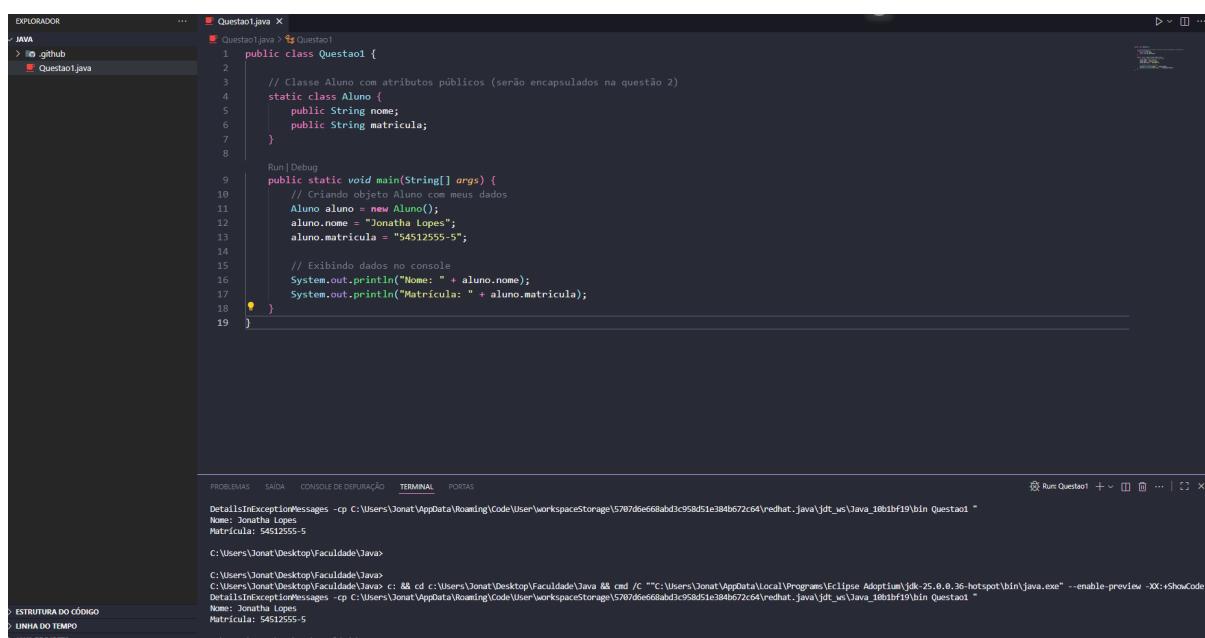
Aluno aluno = new Aluno();
aluno.nome = "Jonatha Lopes";
aluno.matricula = "54512555-5";

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

}

}

```



```

EXPLORADOR ... Questao1.java
IJAVA > github > Questao1
Questao1.java
1  public class Questao1 {
2
3      // Classe Aluno com atributos públicos (serão encapsulados na questão 2)
4      static class Aluno {
5          public String nome;
6          public String matricula;
7      }
8
9      Run | Debug
10     public static void main(String[] args) {
11         // Criando objeto Aluno com meus dados
12         Aluno aluno = new Aluno();
13         aluno.nome = "Jonatha Lopes";
14         aluno.matricula = "54512555-5";
15
16         // Exibindo dados no console
17         System.out.println("Nome: " + aluno.nome);
18         System.out.println("Matrícula: " + aluno.matricula);
19     }
}

```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL PORTAS

```

DetalhesInExceptionMessages -cp C:\Users\Jonat\AppData\Roaming\Code\User\workspaceStorage\57b7d5e68ab3c95bd51e384b672c64\redhat-.java\jdt_ws\Java_10\bf19\bin Questao1 "
Nome: Jonatha Lopes
Matrícula: 54512555-5

C:\Users\Jonat\Desktop\FaculdadeJava>
G:\Users\Jonat\Desktop\FaculdadeJava>C:\Users\Jonat\Desktop\FaculdadeJava> c: && cd c:/Users/Jonat/Desktop/FaculdadeJava && cmd /C ""C:\Users\Jonat\AppData\Local\Programs\Eclipse Adoptium\jdk-25.0.0-26-hotspot\bin\java.exe" --enable-preview -XX:+ShowCode
DetalhesInExceptionMessages -cp C:\Users\Jonat\AppData\Roaming\Code\User\workspaceStorage\57b7d5e68ab3c95bd51e384b672c64\redhat-.java\jdt_ws\Java_10\bf19\bin Questao1 "
Nome: Jonatha Lopes
Matrícula: 54512555-5

G:\Users\Jonat\Desktop\FaculdadeJava>

```

ESTRUTURA DO CÓDIGO LINHA DO TEMPO ÍNDICE DE PROJETO

Questão 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

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a Java project named "Questao2" with files Questao1.java, Questao2.java, and Questao3.java.
- Code Editor:** Displays the content of Questao2.java. The code defines a class "Aluno" with private attributes "name" and "matricula", and public getters and setters for them. It also contains a main method to create an "Aluno" object.
- Terminal:** Shows the command-line output of running the program, which prints the name and matricula of the created object.
- Bottom Bar:** Includes tabs for PROBLEMAS, SAÍDA, CONSOLE DE DEPURAÇÃO, TERMINAL, and PORTAS, along with various icons for running, stopping, and closing the application.

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() {
```



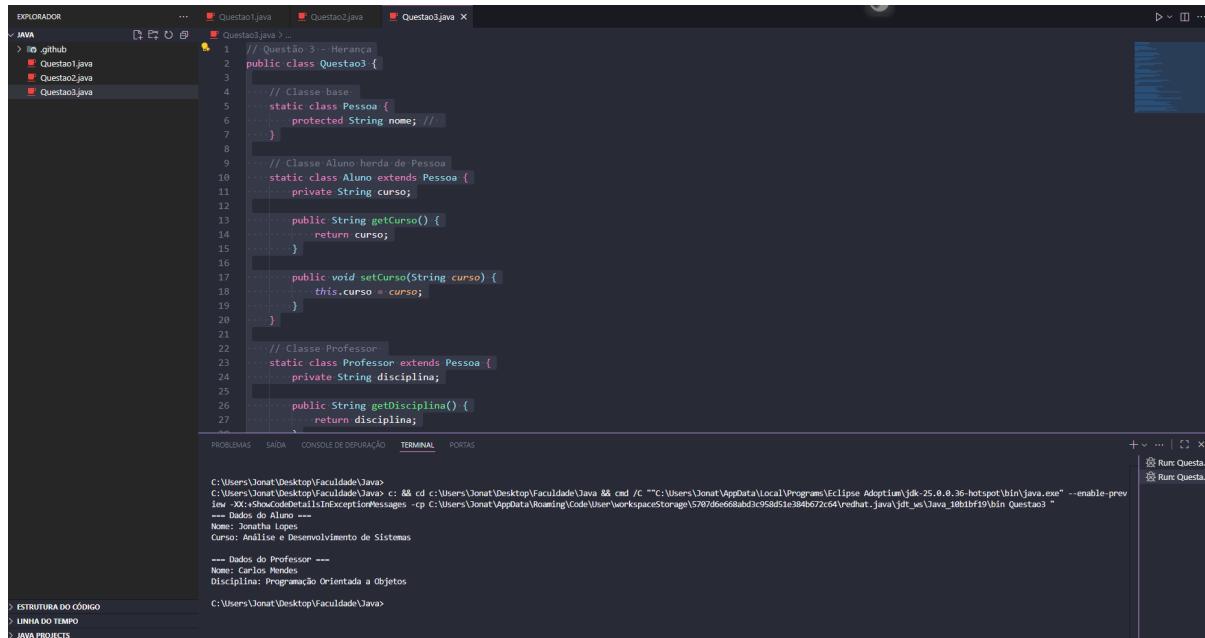
```
        return curso;  
    }  
  
    public void setCurso(String curso) {  
        this.curso = curso;  
    }  
}  
  
// Classe Professor  
static class Professor extends Pessoa {  
    private String disciplina;  
  
    public String getDisciplina() {  
        return disciplina;  
    }  
  
    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 the Eclipse IDE interface with the following details:

- Project Explorer:** Shows three Java files: Questao1.java, Questao2.java, and Questao3.java.
- Code Editor:** Displays the content of Questao3.java, which contains code for inheritance and polymorphism. It includes a base class Pesssoa with a protected attribute name, and two subclasses Aluno and Professor that inherit from it. The Aluno class has a private attribute curso and methods getCurso() and setCurso(). The Professor class has a private attribute disciplina and a method getDisciplina().
- Terminal:** Shows the output of running the code. It prints the details of a student named 'Carlos Mendes' from the course 'Análise e Desenvolvimento de Sistemas' and a professor named 'Carlos Mendes' from the discipline 'Programação Orientada a Objetos'.
- Status Bar:** Shows the path 'C:\Users\Jonat\Desktop\Faculdade\Java>' and other terminal-related information.

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;
}

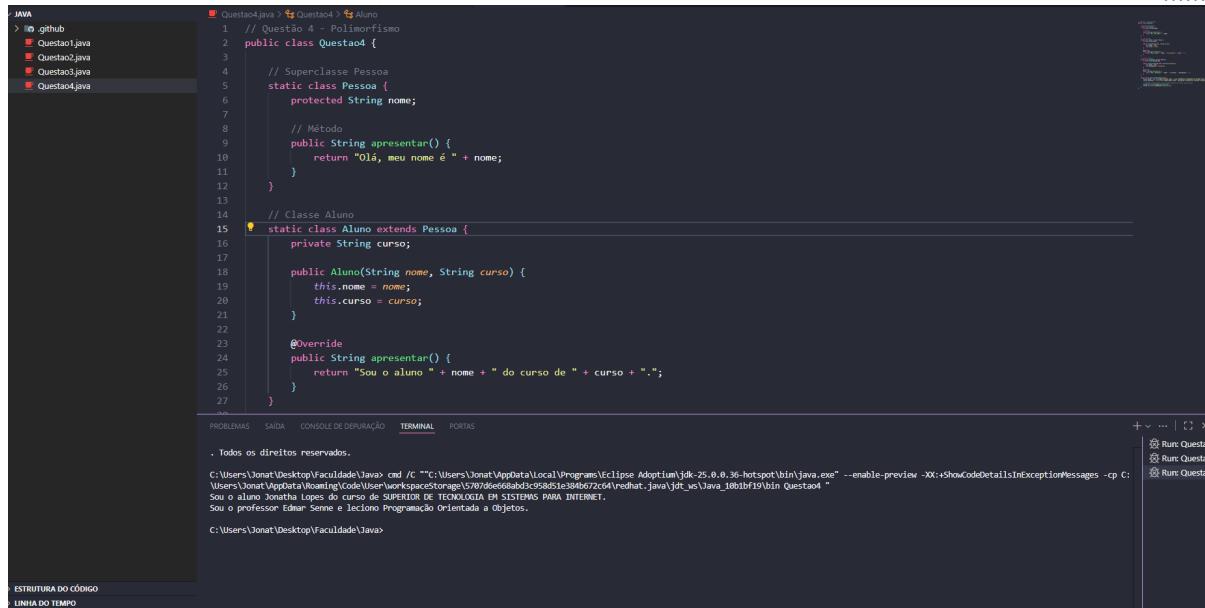
@Override
public String apresentar() {
    return "Sou o professor " + nome + " e leciono " + disciplina + ".";
}

}

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



Questao 5

```
// Questão 5 - Interface
```

```
// 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 the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a Java project named "QUESTAO". Inside, there are files: Questao1.java, Questao2.java, Questao3.java, Questao4.java, Questao5.java, and Questao5.java (selected). The selected file contains Java code for a student evaluation interface.
- Code Editor:** Displays the following Java code:

```
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 }
```

- Terminal:** Shows the command-line output of running the code:

```
C:\Users\Jonat\Desktop\Faculdade\Java>
C:\Users\Jonat\Desktop\Faculdade\Java> java c: && cd 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\CodeUser\workspaceStorage\5787dfe668abd3c958d51e384b672c64\redhat.java\jdt_vs\Java_10\bin\questao5
Aluno Jonatha Lopes obteve ótimas notas!
```

- Bottom Bar:** Includes tabs for PROBLEMAS, SAÍDA, CONSOLAS DE DEBAGURAÇÃO, TERMINAL, and PORTAS. It also shows icons for running configurations and a help icon.