



UAST

Unidade Acadêmica
de Serra Talhada - PE

Desde 2006



BACHARELADO EM
SISTEMAS DE INFORMAÇÃO

MPOO

Site: <https://sigs.ufrpe.br/sigaa/ava/index.jsf>

Site: <https://sites.google.com/site/profricodemery/mpoo>

Disciplina: Modelagem e Programação Orientada a Objetos (MPOO)

Profº: Richarlyson D'Emery

Aluno: _____

LISTA DE EXERCÍCIOS I

Atenção: Ao responder as perguntas desta lista informe, em cada questão, se você baseou sua resposta em alguma pesquisa ou se você respondeu a partir de seus próprios conhecimentos. Sendo assim use: "REFERÊNCIA: Elaboração própria" ou "REFERÊNCIA: citar local da pesquisa".

Você sabe como referenciar uma fonte? Saiba mais sobre as normas da ABNT em

- <https://www.normasabnt.org/referencias-bibliograficas/>
- ABNT NBR 6023 (<https://www.ufpe.br/documents/40070/1837975/ABNT+NBR+6023+2018+%281%29.pdf/3021f721-5be8-4e6d-951b-fa354dc490ed>)

Responda:

1) Explique:

- 1.1) Os porquês de o paradigma da orientação a objetos (POO) ser uma alternativa para o antigo paradigma clássico (imperativo ou estruturado).
- 1.2) Como deu o surgimento do POO?
- 1.3) Quais as principais diferenças entre o paradigma clássico do POO?

2) Analise a frase:

"Quando um produto grande é formado por um único bloco monolítico de código, sua manutenção se torna um pesadelo. Mesmo para o autor de tal monstruosidade, tentar depurar o código é extremamente difícil; para outro programador, compreendê-lo é praticamente impossível" (Schach, 2010 p. 177¹).
Como o paradigma OO dirime essa problemática?

3) No intuito de relacionarmos o paradigma clássico a disciplina MPOO, responda como fazemos na linguagem de programação C para criar uma solução de um problema de forma que possamos reutilizar o código em outras situações?

4) Qual a relação entre struct da linguagem de programação C e o conceito de classe?

5) Em uma demanda de desenvolvimento de software, uma das principais dúvidas surge na modelagem para diferenciar o que se será código ou que apenas trate de uma interação do usuário com o sistema. Sendo assim:

- 5.1) Apresente uma estratégia para diferenciar o que será código e o que não será código, ou seja, apenas uma interação com o sistema.
- 5.2) Como uma demanda no desenvolvimento de software que será transformada em código pode ser apresentada para uma equipe de desenvolvimento de software?
- 5.3) Apresente uma situação em que há pelo menos duas demandas: uma relacionada a interação com o sistema e outra que se trata de uma funcionalidade codificada. Utilize trechos de um documento de requisitos, diagrama de classes e *use case*.

¹ SCHACH, S. R. Engenharia de Software: os paradigmas clássico e orientado a objetos. Porto Alegre: AMGH, 7. ed., 2010. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788563308443/>. Acesso em: 09 nov. 2022.

6) A partir da descrição do problema abaixo, apresente o devido diagrama de use case:

[RF01] - Devolver produto	
Ator Principal	Cliente
Atores Secundários	Funcionário e Caixa
Resumo	Este caso de uso descreve as etapas necessárias para que um cliente devolva um produto comprado.
Pré-condições	É necessário existir um produto.
Pós-condições	Escolher opção de devolução
Fluxo Principal	
Ações do Ator	Ações do sistema
1. Solicitar a devolução de um produto apresentando o produto.	
	2. Executar a Devolução de um produto
	3. Executar Caso de uso Atualizar estoque
	4. Efetuar troca ou ressarcir o valor do produto
Restrições /validações	
1. O produto só poderá ser devolvido pelo cliente que realizou a compra	
2. O produto só poderá ser devolvido se não apresentar avaria.	
Fluxo Alternativo I – devolução	
Ações do Ator	Ações do sistema
	1. Executar caso de uso Efetuar Troca
Fluxo Alternativo II – ressarcimento	
Ações do Ator	Ações do sistema
	1. Ressarcir ao cliente o valor pago pelo produto (não é considerado taxas de envio)

[RF02] – Atualizar estoque	
Ator Principal	Vendedor
Resumo	Este caso de uso descreve as etapas necessárias para atualizar o estoque
Pré-condições	É necessário haver uma demanda de atualização
Pós-condições	Verificar situação do produto
Fluxo Principal	
Ações do Ator	Ações do sistema
1. Atualizar a base com um produto	
	2. Executar a atualização de estoque
Restrições /validações	
1. Existência de produto requisitado para troca	

[RF03] – Efetuar Troca	
Ator Principal	Vendedor
Resumo	Este caso de uso descreve as etapas necessárias para efetuar a troca de um produto
Pré-condições	É necessário existir uma solicitação de troca de produto
Pós-condições	É necessário dar baixa do produto trocado no estoque
Fluxo Principal	
Ações do Ator	Ações do sistema
1. Efetuar a troca de um produto por outro	
	2. Executar o caso de uso Baixar Estoque
Restrições /validações	
1. Existência de produto requisitado para troca	

[RF04] – Ressarcir cliente	
Ator Principal	Caixa
Resumo	Este caso de uso descreve o ressarcimento de um cliente
Pré-condições	É necessário existir uma solicitação de ressarcimento
Fluxo Principal	
Ações do Ator	Ações do sistema
1. Realizar a devolução do ressarcimento de valores a cliente.	
	2. Executar o ressarcimento de valor a cliente
Restrições /validações	
1. Existência de uma conta bancária	
Fluxo Alternativo I – ressarcimento em conta	
Ações do Ator	Ações do sistema
	1. Executar a transferência de valor em conta bancária de um cliente

7) A partir da codificação abaixo

```
//arquivo Classe.java
public class Classe {
    int atributo;
}

//arquivo App.java
public class App {
    public static void main(String[] args) {
        int variavel;
        Classe objeto = new Classe();
    }
}
```

7.1) Podemos dizer que o tamanho da área de memória de `variavel` é o mesmo de `objeto`, uma vez que `objeto` possui apenas um atributo do tipo inteiro?

Antes de responder, pesquise sobre o tamanho de memória para tipos primitivos e para objetos e:

7.2) Explane sobre como é realizada a alocação de memória em JAVA para uma variável do tipo primitiva e para uma instância (objeto).

8) Qual a diferença entre acesso direto e indireto a memória? Como fazemos na Linguagem de Programação C para acessar a memória direta e indiretamente? E como se dá alocação de memória em Java?

9) A partir da codificação abaixo

```
public class Classe {
    int valor1;

    public static void main(String[] args) {
        int valor2;
        Classe instancia1;
        Classe instancia2 = new Classe();
        //?
    }
}
```

Qual seria a saída para `//?` assumindo as instruções abaixo:

Instrução <code>//?</code>	Saída
<code>System.out.println(valor1);</code>	
<code>System.out.println(valor2);</code>	
<code>System.out.println(instancia1);</code>	
<code>System.out.println(instancia2);</code>	
<code>System.out.println(instancia1.valor1);</code>	
<code>System.out.println(instancia2.valor1);</code>	

10) **Desafio:** A partir da problemática de autenticação de usuário apresentada na aula da Semana 1, proponha uma solução representada em diagrama de classes e codificada em Java.