



UAST
Unidade Acadêmica
de Serra Talhada - PE
Desde 2006



MPOO

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

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

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".

Responda:

- 1) Quais as principais diferenças entre o paradigma imperativo (estruturado) do paradigma orientado a objetos.
- 2) No intuito de relacionarmos o aprendizado dos semestres anteriores 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?
- 3) 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:
 - 3.1) Apresente uma estratégia para diferenciar o que será código e o que não será código, ou seja, apenas uma interação com o sistema.
 - 3.2) Como uma demanda no desenvolvimento de software que será transformada em código pode ser apresentada para uma equipe de desenvolvimento de software?
 - 3.3) 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*.
- 4) O que é um método construtor? É possível haver mais de um método construtor? Explique.
- 5) Em Java, por que uma classe principal não pode ser *private*?
- 6) 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?
- 7) Para que serve os métodos *getters* e *setters*?
- 8) Explicação sobre como é realizada a alocação de memória em JAVA para uma variável do tipo primitiva e para uma instância (objeto).
- 9) A partir da codificação abaixo

```
public class Classe {  
    int valor1;  
  
    public static void metodo(){  
        int valor2;  
        //?  
    }  
  
    public static void main(String[] args) {  
        int valor3;  
        Classe instancia1;  
        Classe instancia2 = new Classe();  
        metodo();  
        ///?  
    }  
}
```

Explique o resultado para *//?* e *///??*:

Instrução <i>//?</i>	Saída
System.out.println(valor2);	

Instrução <i>///??</i>	Saída
System.out.println(valor1);	
System.out.println(valor3);	
System.out.println(instancia1);	
System.out.println(instancia2);	
System.out.println(instancia1.valor1);	
System.out.println(instancia2.valor1);	

10) Para que serve os argumentos do main?

11) Sabendo-se da existência de argumentos no método main:

11.1) Ilustre uma aplicação Main.java que utiliza os argumentos do main. A codificação deverá ser apresentada.

11.2) Ilustre um diagrama de *use case* em que um SistemaA se comunica com o SistemaB

11.3) Implemente uma aplicação Java para o diagrama de 11.2)

12) **Desafio:** A partir da problemática (“Questionamento 04.1”) apresentada na vídeo-aula assíncrona da Semana 1, proponha uma solução. *Esta não precisará ser apresentada como resposta desta lista.*