UAST
Unidade Acadêmica de Serra Talhada - PE
UFRPE
Desde 2006

Bacharelado em Sistemas de Informação

MPOO

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

http://ava.ufrpe.br/

https://sigs.ufrpe.br/sigaa/ava/index.jsf

Disciplina: Modelagem e Programação Orientada a Objetos (MPOO)

Profº: Richarlyson D'Emery

LISTA DE EXERCÍCIOS III

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 se 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) O que é a assinatura de um método?
2) Para que serve os argumentos do `main`?
3) Sabendo-se da existência de argumentos no método `main`:
   3.1) Ilustre uma aplicação Main.java que utiliza os argumentos do `main`. A codificação deverá ser apresentada.
   3.2) Ilustre um diagrama de *use case* em que um SistemaA se comunica com o SistemaB
   3.3) Implemente uma aplicação Java para o diagrama de 3.2)
4) Em diversas situações é preciso fazer o uso de informações **`static`** e **`final`**. Qual a diferença entre essas palavras-chaves em um sistema desenvolvido em Java?
5) Para que serve o **`this`**?
6) O que acontece quando é colocado **`static`** na assinatura de um método de uma classe Java? E em um atributo?
7) A partir da codificação abaixo:

```
1  package util;
2  import javax.swing.JOptionPane;
3
4  public class Mensagem {
5
6      public static final String MENSAGEM_FALHA = "Falha no sistema";
7      protected static final String MENSAGEM_SUCESSO= "Operação realizada com sucesso";
8      private static final String MENSAGEM_ERRO = "O sistema será finalizado";
9      static final String MENSAGEM = "Bem vindo ao sistema";
10
11     public static String exibirMensagemFalha(){
12         return "Falha";
13     }
14
15     public static void exibirMensagem(String mensagem){
16         JOptionPane.showMessageDialog(null, mensagem);
17     }
18 }
```

   7.1) Apresente uma aplicação Java que utiliza `exibirMensagemFalha()`, `exibirMensagem(String mensagem)` e as diferentes mensagens *MENSAGEM_FALHA, MENSAGEM_SUCESSO*, *MENSAGEM_ERRO* e *MENSAGEM*.
   7.2) Apresente o diagrama de classes de 7.1)

8) 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 `//?`	Saída
`System.out.println(valor1);`	
`System.out.println(valor2);`	
`System.out.println(valor3);`	

Instrução `//??`	Saída
`System.out.println(valor1);`	
`System.out.println(valor2);`	
`System.out.println(valor3);`	
`System.out.println(instancia1);`	
`System.out.println(instancia2);`	
`System.out.println(instancia1.valor1);`	
`System.out.println(instancia2.valor1);`	

9) Na Lista de Exercícios II vimos que o "método construtor" realiza a alocação de uma área de memória. Mas:

9.1) Todos os atributos sempre devem ser atualizados por parâmetros do método construtor? Explique.

9.2) Devem-se colocar parâmetros para atualizar os atributos **`static`**?

9.3) O que seria um "método destrutor"?

9.4) Para que serve o método `finalize()`?

10) Dado o código abaixo:

```
//Usuario.java
public class Usuario {
    private String login;
    private int senha;

    public Usuario(String login, int senha) {
        this.login = login;
        this.senha = senha;
    }

    public void destroyed(Usuario usuario){
        usuario=null;
        System.gc();
    }
}
//App.java
public class App {
    public static void main(String[] args) {
        Usuario user = new Usuario("Godofredo", 1234);
        user.destroyed(user);
    }
}
```

Responda:

10.1) Houve liberação de memória do usuário "Godofredo"? Explique.

10.2) No método `destroyed` é possível adicionar o comando **`this=null;`**? Explique.

11) Sobre Garbage Collector:

11.1) Ainda que o programador não "chame" diretamente `System.gc()`, quando ele é executado?

11.2) Faça uma aplicação Java ilustrando a alocação de MUITA memória e a ilustração do coletor de lixo em funcionamento.

Desafio

Você, aluno de MPOO, está experienciando situações-problemas do universo de desenvolvimento de software e começará a ser desafiado a solucionar problemas a partir de conhecimentos de Programação e Orientação a Objetos.

12) Crie um diagrama de classes, use case e a codificação Java para cada um dos seguintes problemas:

12.1) Geralmente as frutas contêm casca e caroços. Crie os métodos retirarCaroco( ) - que retira os caroços um a um da fruta, caso haja caroços; retirarCasca () - que retira a casca de uma fruta, caso haja casca; e o método comerFruta( ) que retira a casca e os caroços e elimina uma fruta. Faça o devido uso de coletor de lixo para liberar a memória da fruta. Faça o devido uso de construtores e parâmetros para os métodos. Crie uma aplicação que ilustra diversos tipos de frutas: com caroço(s) e casca; sem caroço(s) e com casca; com caroço(s) e sem casca. Analise e trate em sua solução: como diferenciar uma melancia de um abacate?

12.2) A classe Robot possui os atributos privados nome, posição e direção. Possui métodos para inicializar um robô com um nome indicado e supondo que esteja na posição (0,0) é direcionado para o Norte. Possui dois métodos, sendo um para o robô andar 1 passo e outro para vários passos. Possui um método para mudar a posição do robô. Possui um método que retorna a configuração do robô. Possui um método chamado retornaPosZero( ) que leva o robô a sua posição inicial. Crie uma aplicação com um Robô que utiliza as informações da questão.

12.3) Um aluno na UFRPE possui os atributos privados para nome, matricula, quatro VAs (Verificações de Aprendizagem), sendo uma desta a VA Final, e uma média. A média é tida como a média aritmética das duas maiores VAs, se a média for maior ou igual 7.0 o aluno está aprovado por média, caso contrário, precisará fazer a prova Final. Se a nota da VA Final for superior a 5.0 então o aluno está aprovado, caso contrário, reprovado. O método construtor da classe Aluno inicia o nome e a matrícula de um aluno criado. Possui um método que retorna as duas maiores notas de um aluno. Possui um método que retorna a situação do aluno. Crie uma aplicação que ilustra a média e a situação de um aluno. Como um desenvolvedor saberá qual a situação de um aluno? Será preciso sempre rodar uma aplicação para conhecer sua situação?

13) Escolha um dos problemas da questão anterior (questão 12) e ilustre a partir de capturas de telas (*print screen*) as alterações de dados do(s) objeto(s) durante a execução da aplicação. *Observação: não colocar toda a tela, apenas recortes para a(s) linha(s) que está(ão) sendo executada(s) e para o(s) valor(es) do(s) dado(s).*