UAST
Unidade Acadêmica de Serra Talhada - PE
UFRPE
Desde 2006

BSi
Bacharelado em Sistemas de Informação

MPOO

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

Site: http://ava.ufrpe.br/

Disciplina: Modelagem e Programação Orientada a Objetos (MPOO)

Profº: Richarlyson D'Emery **Aluno:** João Paulo Santiago Cavalcante

LISTA DE EXERCÍCIOS I

Respostas:

1) No paradigma estruturado, eles dividem as suas funcionalidades em subdivisões, levando a criação de sistemas caracterizados por conter grandes blocos de código-fonte, ou subdivisões sem a organização dos dados. Enquanto, o paradigma orientado a objetos utiliza uma maneira mais apropriada de realizar a correta divisão de um programa em conceitos, permitindo que cada conceito centralize em si todos os dados relativos ao que ela representa.

REFERÊNCIA: Elaboração própria

2) Na intenção de ter um código reutilizável, é necessário quebrá-lo em partes menores. A modularização é o ato de dividir o sistema em subsistemas que funcionam independentemente, mas que ao serem conectados, funcionam como um único componente. E dessa forma, é possível reaproveitar componentes de sistemas distintos, acelerando o processo de desenvolvimento. É indispensável que cada função tenha uma única responsabilidade, e realize ela da melhor forma possível. Utilizamos a palavra modularização, para determinar a ação de produzir código reutilizável e consistente, que pode ser utilizado em mais de uma aplicação. As linguagens de programação costumam possuir um conjunto de implementações reutilizáveis desenvolvidas por vários programadores denominadas bibliotecas, que tornam o desenvolvimento de aplicações mais rápidos e menos custosos.

REFERÊNCIA: Elaboração própria

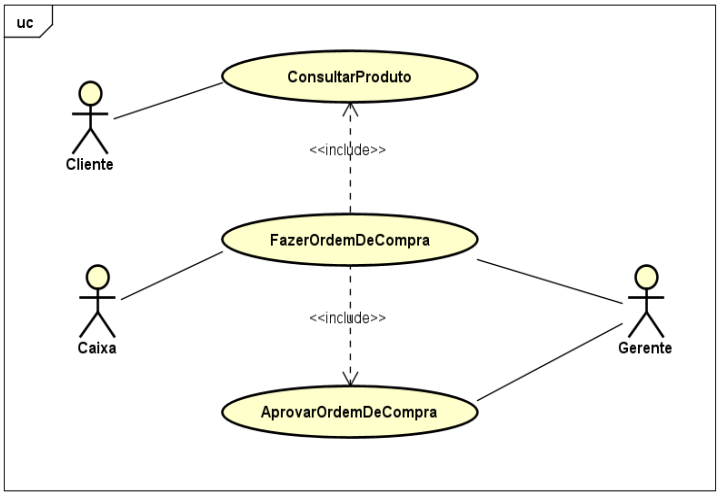
3)

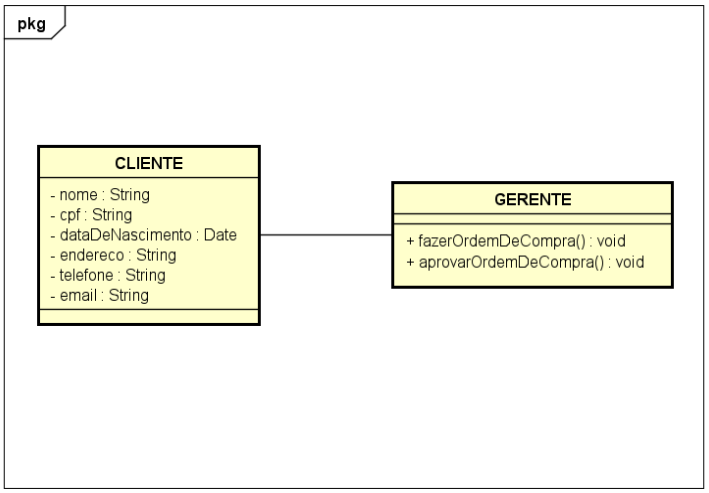
3.1) As interações com o sistema são funcionalidades intrínsecas com o funcionamento da máquina, no caso, do computador, então para classificá-las é necessário ter em mente que algumas coisas funcionam sem a necessidade de ser programadas, como a digitação de uma senha em um campo de senha de um navegador, a captura do evento de input deve ser programada, mas o mecanismo de digitação não precisa ser. Também é importante notar que essas funções podem ser alteradas sem influenciar no funcionamento do sistema.

3.2) Uma demanda de uma funcionalidade cuja implementação é necessária através de código pode ser documentada a partir comumente de um diagrama de casos de uso (UML), no qual o processo de funcionamento de tal funcionalidade é mais importante que os detalhes dessa implementação.

3.3) Documento de Requisitos, Diagrama de Use Case e Classes

Documento de Requisitos	
Nome:	Consultar Produto
Atores:	Cliente
Descrição:	O sistema deve inserir um novo produto
Pré-condições:	Ter efetuado o login no sistema
Pós-condições:	Produto cadastrado com sucesso
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Cliente informa seu nome completo	
2. Cliente informa seu CPF	
3. Cliente informa seu número celular	
4. Cliente informa seu e-mail	
	5. Verificar se o cliente está cadastrado
	6. Cadastro concluído
Fluxo Alternativo:	Cliente não cadastrado, cadastro é cancelado, realizar cadastro do cliente





REFERÊNCIA: Elaboração própria

4) Um método construtor é o responsável por definir o estado de um objeto quando ele é instanciado. Pode sim haver mais de um construtor, desde que a assinatura de cada um seja diferente. Isso entra dentro do assunto de sobrecarga. A identificação de um construtor em uma classe é sempre o mesmo nome da classe.

REFERÊNCIA: Elaboração própria

5) Simplesmente porque o JVM precisa encontrar uma classe para inicializar o programa. Se a classe é private, ela não pode ser enxergada fora do seu escopo. Então geralmente classes private são colocadas dentro de classes públicas para que possam ser utilizadas apenas dentro daquele escopo. Então se a classe for private, nenhuma classe externa poderá visualizá-la.

REFERÊNCIA:
Maniero. “Motivos para usar classe privada”
Disponível em: https://pt.stackoverflow.com/questions/290621/motivos-para-usar-classe-privada

6) Acesso direto é quando o programador de tem o controle e a forma como os dados são alocados. Acesso indireto é quando a própria linguagem gerencia esse controle deixando para o programador apenas as formas para alocar os dados. Na linguagem de C o controle de memória pode ser melhor efetuado graças à implementação de ponteiros que armazenam endereços de memória, permitindo uma gama de possibilidades de implementação. Outra característica do C intrínseca com o acesso direto à memória é o fato da linguagem não alocar dinamicamente espaço para variáveis automaticamente. Já no Java, é possível definir a visibilidade do objeto para private, respeitando sempre a lógica implementada pela classe.

REFERÊNCIA: Elaboração própria

7) Os métodos getters e setters servem para acessar atributos que estejam privados ou protegidos, mas lembrando que, atributos protegidos podem ser acessados no mesmo pacote. Utilizando getters e setters, é possível entender qual o comportamento do objeto, sempre que algum desses dados for acessado, são úteis para realizar validações e tornar a utilização dos dados mais consciente pelo programador. Isso serve para controlar o acesso aos atributos da classe e é uma boa prática.

REFERÊNCIA: Elaboração própria

8) Ao definir uma variável de um tipo, um espaço é alocado para armazenar os dados que por sua vez só serão guardados após a inicialização da variável. A própria JVM faz o controle da alocação de memória para você programador. Quando um objeto é criado, pela chamada de new <*construtor*>, os seguintes passos acontecem:

- um espaço de memória é alocado para o objeto criado e suas variáveis de instância são inicializadas com seus valores default;
- as inicializações explícitas programadas para as variáveis de instância são realizadas;
- o construtor chamado é executado;
- o endereço de memória no qual o objeto está alocado é retornado e atribuído para a variável que referência ao objeto.

A diferença é que quando for um tipo primitivo, a variável retorna um endereço para o espaço no qual o objeto está alocado na memória.

REFERÊNCIA:
Ludimila Monjardim Casagrande. “POO - 03 - Alocação de Memória em Java”
Disponível em: https://pt.slideshare.net/ludimila_monjardim/poo-04alocao-dememria

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();
        //??
    }
}
```

Instrução //?	Saída
System.out.println(valor2);	ERRO

Instrução //??	Saída
System.out.println(valor1);	ERRO
System.out.println(valor3);	ERRO
System.out.println(instancia1);	ERRO
System.out.println(instancia2);	Classe@4617c264
System.out.println(instancia1.valor1);	ERRO
System.out.println(instancia2.valor1);	0

REFERÊNCIA: Elaboração própria

10) Os parâmetros do método main de uma classe podem ser passados na hora de executar o programa. Cada argumento é separado por um espaço. São úteis para executar software de maneira parametrizada em sua inicialização, via linha de comando. Eles são uma lista de parâmetros passados durante a inicialização de uma JVM para determinar algo no fluxo de funcionamento do programa, algo que pode influenciar totalmente no funcionamento da aplicação.

REFERÊNCIA: Elaboração própria

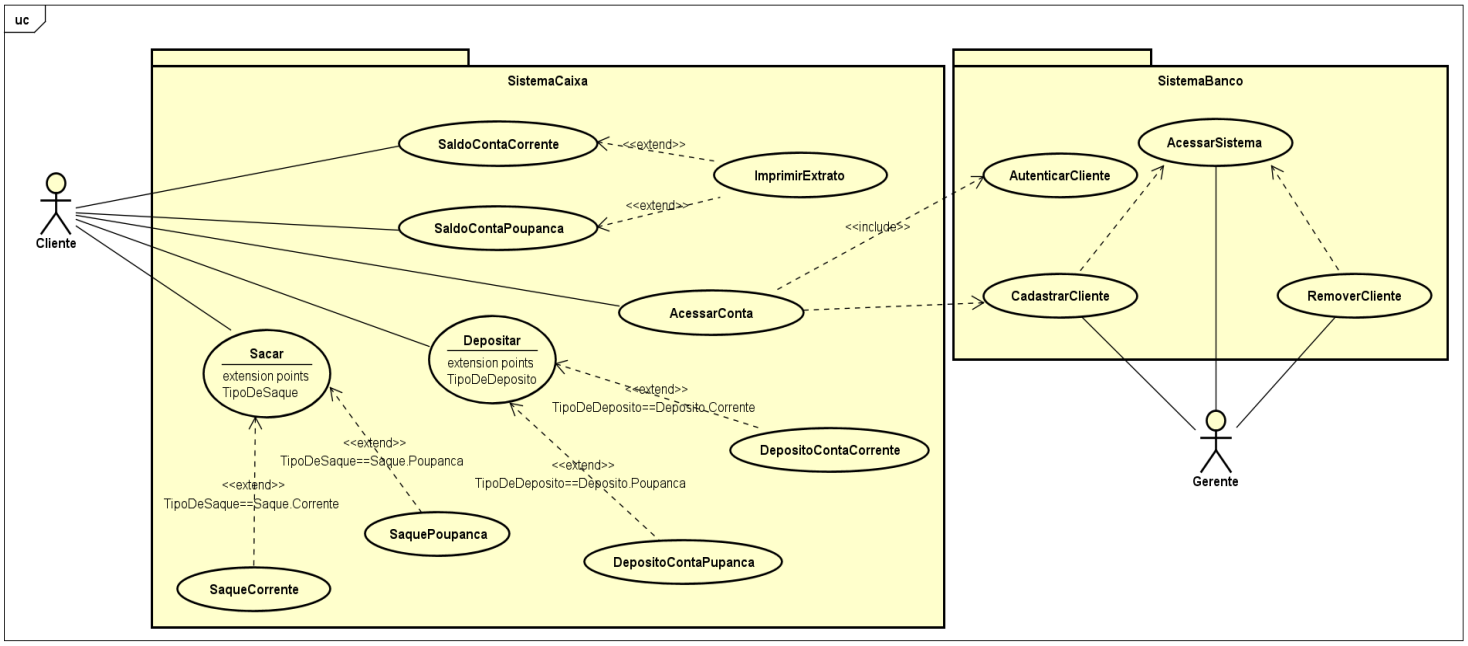
11)

11.1) Eclipse IDE (arquivo): ***Main.java***

```
1
2  public class Main {
3
4      public static void main(String[] args) {
5
6          System.out.println("Seja bem vindo, " + args[0]);
7
8          }
9
10 }
11
```

RUN deu certo: **Seja bem vindo, Paulo**

11.2) Diagrama de Casos de Uso (arquivo): ***diagrama.asta***



11.3) Programa (arquivo): ***SistemaBancoDados***

Observações:

- Por conta do tempo limitado, algumas funções são apenas ilustrativas.
- O sistema é focado na parte da autenticação, pois é nesta parte que se encontra a comunicação entre as classes de **SistemaCaixa** e **SistemaBanco** para validar um Cliente.
- Algumas funções não estão presentes, por não serem tão relevantes na resposta da pergunta.
- O sistema funciona via terminal, dicas de execução para cada classe são exibidas na ausência de comandos.

REFERÊNCIA: Elaboração própria