Professor: José Antonio Gonçalves

Aluno: Pedro de Oliveira Machado - RA: 2417855

1º Questionário - POO

1º Por que falamos que Java é totalmente aderente às técnicas de Orientação a Objetos?

A linguagem java faz uso de conceitos como classe, atributos, objetos e abstração.

2º Explique o que é e como se utiliza o processo de “abstração”.

Na abstração, utiliza-se das caracteristicas de algo real, seus aspectos, para defini-lo.

3º Quais são os artefatos produzidos na programação orientada a objetos?

Objetos, Métodos, Atributos e Classes.

4º O que é um “tipo primitivo” de dados?

Ele é proprio da linguagem de programação, não apresenta métodos, nem possui representação em forma de um objeto.

5º O que é um “tipo abstrato” de dados?

Chamado de TAD, serve como um modo de de encapsular certos dados e agrupar métodos que vão trabalhar com eles.

6º Explique o que é o “Garbage Collector”. Como este recurso pode dinamizar o funcionamento do sistema?

Ele evita que objetos já sem referência, ou seja, que já não estão mais sendo utilizados, continuem utilizando memória, limpando-os. Fazendo com que o sistema tenha uma probabilidade menor de que a memória acabe.

7º Considerando o modo Shell (linhas de comando) do sistema operacional Windows, como se faz para:

a) Compilar um codigo fonte Java;

javac Classe.java

b) Fazer com que a J.V.M. (Máquina Virtual Java) execute uma aplicação Java.

java Classe

8º O que é o “ByteCode”?

Ele serve como uma ponte entre o codigo da maquina e o do usuário. Conhecido como Código intermediário.

9º Explique o que é a característica “Portabilidade”. Como isto é possível com aplicações Java? Para esta resposta relacione 4 “personagens” deste cenário: o código fonte (arquivo .java), o byteCode (arquivo .class), o Sistema Operacional e a JVM (Java Virtual Machine).

O código fonte, utiliza-se do byteCode que é passivel de portabilidade, para que seja interpretado pela JVM do sistema operacional da máquina.

10º Justifique a afirmação que diz que “a segurança em Java se dá em dois níveis: proteção de hardware e proteção de software”.

Proteção de Hardware

Java não faz uso de ponteiros, o que impede a alocação cruzada de memória com outro programa, com isso, garantindo a integridade da memória.

Proteção de Software

Se dá na alta confiabilidade das API’s fornecidas nativamente, que possuem baixa taxa de erro baseada em uma alta reusabilidade.

11º Explique como aplicamos o conceito de “Modularidade” em Java. Na resposta desta questão deve-se tratar dos conceitos sobre “Acoplagem” e “Coesão”.

Na Programação Orientada a Objetos, a modularidade é um importante recurso, pois faz com que um objeto possa ser reutilizado de forma independente,

fazendo o uso de um objeto.

a) Como esta característica pode ajudar na questão da “Manutenibilidade”?

Facilita a manutenibilidade pois quando alterações forem feitas, o resto do projeto se manterá inalterado, podendo-se fazer modificações pontuais.

Para que servem os objetos:

a) this;

Cria uma referência a um atributo pertencente a esta mesma classe.

b) super.

Cria referencia a um atributo da classe mãe, caso haja.

13º Usando Java, dê um exemplo que contemple as respostas das questões 12.a e 12.b.

```
this.objeto = objeto;
super.impDados();
```

14º Dentre os conceitos de sustentar a Orientação a Objetos, explique:

a) Encapsulamento:

a.i) Seus níveis (explique cada um dos três níveis);

Public

Pode ser acessado e alterado por qualquer classe.

Private

Só pode ser acessado dentro de sua classe.

Protected

Pode acessar quem herda-lo ou fizer parte do mesmo package.

a.ii) Como o Encapsulamento pode nos ajudar na padronização,segurança e “manutenibilidade” no desenvolvimento de sistemas;

Deixa a cargo do programador o controle total do acesso de seus objetos e métodos no momento em que cria-los.

b) Herança:

b.i) Explique os conceitos que “Generalização” e “Especialização”;

Generalização

Quando uma classe filha da acesso aos seus métodos e atributos a outra classe.

Especialização

Demonstra o maior nível de detalhes que uma classe dá a um objeto.

b.ii) Como o mecanismo de Herança pode nos ajudar na padronização, segurança e “manutenibilidade” no desenvolvimento de sistemas;

Dá a possibilidade de que determinadas classe e atributos possam ser acessadas somente por classes especificas, sendo mais um meio de manter caracteristicas do código seguras.

b.iii) Explique o conceito de “Reusabilidade”. Como este é aplicado no mecanismo de Herança e, ainda, como esta possibilidade nos ajuda no dinamismo da codificação.

Através da Reusabilidade, usamos e reutilizamos a utilização de uma classe, tornando o código mais dinamico e livre.

c) Polimorfismo:

c.i) Sobrecarga;

Métodos com mesmo nome em uma classe, porém, com assinatura diferente.

c.ii) Sobrescrita;

Métodos com mesma assinaturaem diferentes classes.

c.iii) Coerção;

Relação de uma classe e seus métodos.

15º Construa um programa para exemplificar as respostas das questões 14.a, 14.b e14.c

```
public abstract class Produto implements Desconto{


  private float preco;
  private String marca, modelo;


  public Produto(){
    preco = 0;
    marca = " ";
    modelo = " ";
  }


  public Produto(float preco, String marca, String modelo){
    this.preco = preco;
    this.marca = marca;
    this.modelo = modelo;
    }


  public void calcDesconto(float valor_desc, float precox){
    setPreco(precox - valor_desc);
    System.out.println("\nNovo preço: "+getPreco());
  }


  public float getPreco(){
    return preco;
  }
  public String getMarca(){
    return marca;
  }
  public String getModelo(){
    return modelo;
  }
```

```
  public void setPreco(float preco){
    this.preco = preco;
  }
  public void setMarca(String marca){
    this.marca = marca;
  }
  public void setModelo(String modelo){
    this.modelo = modelo;
  }
}


public interface Desconto{

  public void calcDesconto(float valor_desc, float precox);
}


public class FoneOuvido extends Produto{


  private String p_mic;
  private String conexao;


  public FoneOuvido(){
    p_mic = "nao declarado";
    conexao = "nao declarado";
  }


  public String getP_mic(){
    return p_mic;
  }
  public String getConexao(){
    return conexao;
  }


  public void setP_mic(String p_mic){
```

```
    this.p_mic = p_mic;
  }
  public void setConexao(String conexao){
    this.conexao = conexao;
  }
}


public class Principal{
  public static void main(String arg[]){


    FoneOuvido f = new Produto();
    Leitura l = new Leitura();


    f.setMarca(l.entDados("\nMarca: "));
    f.setModelo(l.entDados("\nModelo: "));
    f.setPreco(Float.parseFloat(l.entDados("\nPreço em R$: ")));


    f.setP_mic(l.entDados("\nPresença de Microfone: "));
    f.setConexao(l.entDados("\nConectividade - [Wireless] | [Cabeado]: "));


    f.calcDesconto(Float.parseFloat(l.entDados("\nInforme o valor de desconto:
")),f.getPreco());


    System.out.println("\n---- PRODUTO ----");
    System.out.println("Tipo: Fone de Ouvido");
    System.out.println("Marca: "+f.getMarca());
    System.out.println("Modelo: "+f.getModelo());
    System.out.println("Microfone: "+f.getP_mic());
    System.out.println("Conexão: "+f.getConexao());
    System.out.println("\nPreco: R$"+f.getPreco());
  }
}
```

16º Explique o que são trocas de mensagens? Como isso acontece?

Se dá na utilização de métodos através de objetos dos tipos das classes, que conversam entre si.

17º O que é um “método construtor”? Qual sua importância? Faça um código que demonstra sua explicação.

Informa as caracteristicas dos objetos de um metodo para que possam ser alocadas os seus devidos espaços de memória.

```
public class Teste{
    public void Teste(){
        System.out.println(“teste”);
    }

public class Principal{
    public static void main(String arg[]){
    Teste t = new Teste();
    }
}
```

18º Explique o que são como e quando utilizamos:

a) Classe abstrata;

Uma classe abstrata é uma classe que não pode ser instanciada, ou seja, ela é dependente de outra para que funcione. Por exemplo, uma classe mãe, onde suas classes filhas a referenciam indiretamente ao serem instanciadas na classe Principal.

b) Método abstrato;

É um método que para ser utilizado deve ser referenciado dentro de outra claase, para que possa ser implementado, como no caso de uma classe mãe abstrata que a possui e uma classe filha (onde ela ela deve ser implementada).

c) Classe final;

Quando uma classe não pode ser herdada.

d) Atributo final;

Quando seu valor não pode ser alterado, ou seja, é uma constante.

e) Método final;

Método que não pode ter seus valores sobrescritos.

19º Dentro da tecnologia Java, explique o que é a estrutura de dados “Interface”. Quando a utilizamos?

Uma classse em que todos os seus metodos são concebidos fora dela, seus atributos são constantes e ela serve pois linguagens aderentes a POO não possuem herança multipla.