

Alunos: Amanda Caetano Amorim e Maicon Cezar Azambuja da Cunha

Prova 1 de Orientação a Objetos Observações:

- A prova deverá ser feita individualmente ou em dupla;
- O exercício que for plagiado de colegas ou internet receberá nota ZERO;
- A entrega deverá conter um arquivo para as respostas teóricas (1 a 5) e as classes solicitadas (.java). Ela pode ser feita em qualquer formato compactado (zip, rar, 7zip, tar, tar.gz...).

1) Explique com suas palavras a necessidade de criarmos nossas próprias Classes na orientação a objetos. (1.0)

R: A necessidade de fazermos nossas próprias classes consiste no pensamento de tornar-se algo mais explícito para a(s) pessoa(s) que a criam, sendo algo mais prático, conceitual e subjetivo. Sua finalidade, da classe, é de armazenar as variáveis/características de um determinado programa.

2) Conforme visto em sala há três tipos de modificadores de acesso, cite e explique cada um deles. (1.0)

R: Public: Um tipo de acesso que todas as classes/métodos do projeto podem acessar de forma livre sem necessidade de criação de um outro método/objeto.

Protegido: Um tipo de acesso em que as outras classes/métodos podem acessar de forma controlada, ou seja, através de um método/objeto.

Private: Apenas a classe pode acessar, ou seja, é necessário o acesso da própria classe para utilizar as variáveis que existem sob controle privado através de encapsuladores.

3) Quando criamos um objeto utilizamos a palavra “new”, o que essa palavra indica? (1.0)

R: A palavra new, com o significado de “nova”, em programação java, significa que está havendo a criação de um novo objeto.

4) Qual a finalidade dos gets e sets? (1.0)

R: Denominados Encapsuladores, possuem como finalidade permitir o acesso de variáveis privadas de um classe por outras classes do mesmo projeto.

5) Dê um exemplo de Classe com 5 atributos e pelo menos 1 método. (1.0)

R:

```
package oo;
```

```
public class Estudante
{
    private String nome;
    private int RA;
    private String curso;
    private String cpf;
```

```

private int idade;

public void imprimirDados()
{
    System.out.println("Nome: "+getNome());
    System.out.println("RA: "+getRA());
    System.out.println("Curso: "+getCurso());
    System.out.println("Cpf: "+getCpf());
    System.out.println("Idade: "+getIdade());
}

public Estudante()
{
}

public Estudante(String nome, int RA, String curso, String cpf, int idade)
{
    this.nome = nome;
    this.RA = RA;
    this.curso = curso;
    this.cpf = cpf;
    this.idade = idade;
}

public static void main(String[] args)
{
    Estudante e = new Estudante("Amanda",1745,"Informática","000.000.000-00",40);
    e.imprimirDados();
}

/**
 * @return the nome
 */
public String getNome() {
    return nome;
}

/**
 * @return the RA
 */
public int getRA() {
    return RA;
}

/**
 * @return the curso
 */

```

```

public String getCurso() {
    return curso;
}

/**
 * @return the cpf
 */
public String getCpf() {
    return cpf;
}

/**
 * @return the idade
 */
public int getIdade() {
    return idade;
}
}

```

6) Crie uma classe Produto em JAVA, em seguida coloque os atributos: id, nome e preço. Além disso, crie os encapsuladores e o método imprimirDados() responsável por mostrar todos os dados de um produto. (2.5)

R:

```

package oo;

public class Produto
{
    private int id;
    private String nome;
    private double preco;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {

```

```

        this.nome = nome;
    }

    public double getPreco() {
        return preco;
    }

    public void setPreco(double preco) {
        this.preco = preco;
    }

    public Produto(int id, String nome, double preco)
    {
        this.id = id;
        this.nome = nome;
        this.preco = preco;
    }

    public void imprimirDados()
    {
        System.out.println("Id: "+getId());
        System.out.println("Nome: "+getNome());
        System.out.println("Preço: "+getPreco());
    }
}

```

7) Crie uma classe TesteProduto, em seguida utilize o trecho de código a seguir: Produto sabao = new Produto(1, “sabao”, 5); sabao.imprimirDados(); Qual foi a saída impressa no console? (Cole na resposta do exercício) (2.5)

R:

```

package prova1;

public class TesteProduto {

    public static void main(String[] args) {

        Produto sabao = new Produto(1, "sabao",5);
        sabao.imprimirDados();

    }
}

```

}

O resultado foi

“run:

Id: 1

Nome: sabao

Preço: 5.0

CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)

”

