

Atividade Avaliativa – POO

Aluno (a): Eric Henrique de Castro Chaves

Data: 18/09/2021

Prova 1 – Programação Orientada a Objetos

Professor – Daniel de Oliveira Capanema

Valor: 12 pontos.

O alunos poderá fazer a prova à mão ou no computador. Ao final o aluno deverá gerar um único arquivo pdf e entregar a atividade no portal da universidade. Códigos iguais serão avaliados como 0.

Questão 1 – 1 pontos - Quais dos identificadores abaixo podem ser usados como nomes de classes, atributos, métodos e variáveis em Java? Quais não podem, e por quê?

A. contador : O “ contador “ pode ser utilizado em classes porem não e o recomendado o ideal seria “ Contador ”, tambem pode ser utilizado nos metodos e em variáveis.

B. 1contador : Não pode ser utilizado em nenhum alternativa por começar com um numeral.

C. contador de linhas : Não pode ser utilizado em nenhum alternativa por haver espaços entre as palavras.

D. Contador : O “ Contador ” pode ser utilizado em metodos porem não e recomendado o ideal seria “ contador ”, ele pode ser usado perfeitamente nas classes tambem, pode ser utilizado em variaveis porem não e recomendado por começar com maiusculo o ideal seria “ contador ”

E. count : O “ count “ pode ser utilizado em classes porem não e o recomendado o ideal seria “ Count ”, tambem pode ser utilizado nos metodos e em variáveis.

Questão 2 – 2 pontos - Considere a classe A abaixo. Implemente o método "m" conforme as seguintes regras: o método

retorna true se o valor proposto no parâmetro de m for maior que zero. O método retorna false se o valor proposto no parâmetro de m for menor ou igual à zero. Se o método retornar true então ele deve, também, alterar o valor do atributo a para 10 vezes o valor proposto no parâmetro do método m.

```
public class A {  
    int a;  
    public boolean m(int valor) {  
        // ?  
    }  
}
```

```

/*Questão 2 – 2 pontos - Considere a classe A abaixo. Implemente o método "m" conforme as seguintes regras: o método
retorna true se o valor proposto no parâmetro de m for maior que zero. O método retorna false se o valor proposto no
parâmetro de m for menor ou igual à zero. Se o método retornar true então ele deve, também, alterar o valor do atributo a
para 10 vezes o valor proposto no parâmetro do método m.
*/

package Questao02;

public class A {
    int a;

    public boolean m(int valor) {
        if (valor > 0) {
            a = valor * 10;
            return true;
        } else {
            return false;
        }
    }
}

```

Questão 3 – 2 pontos -Reescreva as classes Conta e TestaConta aplicando o encapsulamento dos atributos da classe Conta, ou seja, modificando o acesso dos mesmos para apenas a própria classe e criando métodos de acesso.

Conta.java	TestaConta.java
<pre> class Conta { public String titular; public int numero; public double saldo; } </pre>	<pre> class TestaConta { public static void main(String [] args) { Conta conta = new Conta(); conta.titular="jose"; conta.numero=6; conta.saldo=30.50; System.out.println("Olá "+conta.titular+", seu saldo é de "+conta.saldo); } } </pre>

Novas classes:

Conta.java	TestaConta.java
<pre>class Conta { }</pre>	<pre>class TestaConta { public static void main(String [] args) { } }</pre>

/*Reescreva as classes Conta e TestaConta aplicando o encapsulamento dos atributos da classe Conta, ou seja, modificando o acesso dos mesmos para apenas a própria classe e criando métodos de acesso.*/

```
package Questao03;
```

```
public class TestaConta {
```

```
    Run | Debug
```

```
    public static void main(String[] args){
```

```
        Conta conta = new Conta();
```

```
        conta.setTitular("jose");
```

```
        conta.setNumero(6);
```

```
        conta.setSaldo(30.50);
```

```
        System.out.println("Olá " + conta.getTitular() + ", seu saldo é de " + conta.getSaldo() + "\nO numero da sua conta e " + conta.getNumero());
```

```
    }
```

```
}
```

/*Reescreva as classes Conta e TestaConta aplicando o encapsulamento dos atributos da classe Conta, ou seja, modificando o acesso dos mesmos para apenas a própria classe e criando métodos de acesso.*/

```
package Questao03;
```

```
public class Conta {
```

```
    private String titular;
```

```
    private int numero;
```

```
    private double saldo;
```

```
    public String getTitular() {
```

```
        return titular;
```

```
    }
```

```
    public void setTitular(String titular) {
```

```
        this.titular = titular;
```

```
    }
```

```
    public int getNumero() {
```

```
        return numero;
```

```
    }
```

```
    public void setNumero(int numero) {
```

```
        this.numero = numero;
```

```
    }
```

```
    public double getSaldo() {
```

```
        return saldo;
```

```
    }
```

```
    public void setSaldo(double saldo) {
```

```
        this.saldo = saldo;
```

```
    }
```

```
}
```

```
c:\Users\Eric\Documents\GitHub\Exercicios-UIT.Computa-o - Console do VS Code
Olá jose, seu saldo é de 30.5
O numero da sua conta e 6
Pressione qualquer tecla para continuar. . .
```

Questão 4 – 2 pontos -Implemente a classe ClasseTeste01 de forma a funcionar conforme a saída impressa abaixo. Ou seja, você deverá implementar o método m da classe ClasseTeste01 de forma que ao executar o código abaixo ele imprima conforme apresentado.

```
class Testa {
    public static void main(String [] args) {
        ClasseTeste01 oExercicio = new ClasseTeste01();
        oExercicio.m("Pedro", 7);
        oExercicio.m("Maia", 4);
    }
}
```

.. O fragmento de código acima deve gerar a seguinte saída na tela:

Olá Pedro, você tem 7 anos.
Olá Maria, você tem 4 anos.

/*Questao 4 - 2 pontos -Implemente a classe ClasseTeste01 de forma a funcionar conforme a saída impressa abaixo.
Ou seja, você deverá implementar o método m da classe ClasseTeste01 de forma que ao executar o código abaixo ele imprima conforme apresentado.

```
class Testa {  
public static void main(String [] args) {  
ClasseTeste01 oExercicio = new ClasseTeste01(); oExercicio.m("Pedro", 7);  
oExercicio.m("Maia", 4);  
}  
  
.. O fragmento de código acima deve gerar a seguinte saída na tela:  
  
Olá Pedro, você tem 7 anos.  
Olá Maria, você tem 4 anos.  
*/  
  
package Questao04;  
  
public class ClasseTeste01 {  
  
    public void m(String nome, int numero) {  
        System.out.println("Ola "+ nome + ",voce tem " + numero + " anos.");  
    }  
  
}
```

/*Questão 4 - 2 pontos -Implemente a classe ClasseTeste01 de forma a funcionar conforme a saída impressa abaixo.
Ou seja, você deverá implementar o método m da classe ClasseTeste01 de forma que ao executar o código abaixo ele imprima conforme apresentado.

```
class Testa {  
public static void main(String [] args) {  
ClasseTeste01 oExercicio = new ClasseTeste01(); oExercicio.m("Pedro", 7);  
oExercicio.m("Maia", 4);  
}  
  
.. O fragmento de código acima deve gerar a seguinte saída na tela:  
  
Olá Pedro, você tem 7 anos.  
Olá Maria, você tem 4 anos.  
*/  
  
package Questao04;  
  
public class Testa {  
  
    Run | Debug  
    public static void main(String[] args){  
  
        ClasseTeste01 oExercicio = new ClasseTeste01();  
        oExercicio.m("Pedro", 7);  
        oExercicio.m("Maia", 4);  
    }  
  
}
```

c:\Users\Eric\Documents\GitHub\Exercicios-UIT.Computa-o - Console do VS Code

Ola Pedro,voce tem 7 anos.
Ola Maia,voce tem 4 anos.
Pressione qualquer tecla para continuar. . .

Questão 5 – 5 pontos - Crie uma classe denominada Elevador para armazenar as informações de um elevador dentro de um prédio. A classe deve armazenar o andar atual (térreo = 0), total de andares no prédio, excluindo o térreo, capacidade do elevador, e quantas pessoas estão presentes nele. A classe deve também disponibilizar os seguintes métodos:

- Inicializa(Constructor): que deve receber como parâmetros a capacidade do elevador e o total de andares no prédio (os elevadores sempre começam no térreo e vazio);
- Entra: para acrescentar uma pessoa no elevador (só deve acrescentar se ainda houver espaço);
- Sai: para remover uma pessoa do elevador (só deve remover se houver alguém dentro dele);
- Sobe: para subir um andar (não deve subir se já estiver no último andar);
- Desce: para descer um andar (não deve descer se já estiver no térreo);
- Encapsular todos os atributos da classe (criar os métodos set e get).

```
/*Questão 5 – 5 pontos - Crie uma classe denominada Elevador para armazenar as informações de um elevador dentro de um prédio.
A classe deve armazenar o andar atual (térreo = 0), total de andares no prédio, excluindo o térreo, capacidade do elevador,
e quantas pessoas estão presentes nele. A classe deve também disponibilizar os seguintes métodos:
Inicializa(Constructor): que deve receber como parâmetros a capacidade do elevador e o total de andares no prédio (os elevadores sempre começam no térreo e vazio);
Entra: para acrescentar uma pessoa no elevador (só deve acrescentar se ainda houver espaço);
Sai: para remover uma pessoa do elevador (só deve remover se houver alguém dentro dele);
Sobe: para subir um andar (não deve subir se já estiver no último andar);
Desce: para descer um andar (não deve descer se já estiver no térreo);
Encapsular todos os atributos da classe (criar os métodos set e get).
*/

package Questao05;
import java.util.Scanner;
public class UIT1709Elevador {

    Run | Debug
    public static void main(String[] args){

        Scanner input = new Scanner(System.in);

        System.out.println("Informe a Capacidade de pessoas no Elevador:");
        int capacidadeDoElevador = input.nextInt();
        System.out.println("Informe o total de andares no Elevador:");
        int totalDeAndares = input.nextInt();

        Elevador elevador = new Elevador(capacidadeDoElevador, totalDeAndares);

        System.out.println(elevador);

        int opcao=0;
        do{
            System.out.println("Digite o que deseja:");
            System.out.println("0-Sair");
            System.out.println("1-Entrar uma pessoa no elevador:");
            System.out.println("2-Sair uma pessoa do elevador ");
            System.out.println("3-Subir Andar");
            System.out.println("4-Descer Andar");
            System.out.println("5-Imprimir D etalhes");
            opcao=input.nextInt();
            switch(opcao){
                case 0:
                    System.out.println("Muito Obrigado ate Mais.");
                    break;
                case 1:
                    elevador.entrar();
                    break;
                case 2:
                    elevador.sair();
                    break;
                case 3:
                    elevador.sobe();
                    break;
                case 4:
                    elevador.desce();
                    break;
                case 5:
                    System.out.println(elevador.imprimeDetalhe());
                    break;
                default:
                    System.out.println("Opção Inválida");
            }
        }while(opcao!=0);

        input.close();
    }
}
```

```

/*Questão 5 - 5 pontos - Crie uma classe denominada Elevador para armazenar as informações de um elevador dentro de um prédio.
A classe deve armazenar o andar atual (térreo = 0), total de andares no prédio, excluindo o térreo, capacidade do elevador,
e quantas pessoas estão presentes nele. A classe deve também disponibilizar os seguintes métodos:
Inicializa(Construtor): que deve receber como parâmetros a capacidade do elevador e o total de andares no prédio (os elevadores sempre começam no térreo e vazio);
Entra: para acrescentar uma pessoa no elevador (só deve acrescentar se ainda houver espaço);
Sai: para remover uma pessoa do elevador (só deve remover se houver alguém dentro dele);
Sobe: para subir um andar (não deve subir se já estiver no último andar);
Desce: para descer um andar (não deve descer se já estiver no térreo);
Encapsular todos os atributos da classe (criar os métodos set e get).
*/

package Questao05;

public class Elevador {

    private int andarAtual;
    private int totalDeAndares;
    private int capacidadeDoElevador;
    private int pessoasElevador;

    public Elevador(int capacidadeDoElevador, int totalDeAndares) {
        setCapacidadeDoElevador(capacidadeDoElevador);
        setTotalDeAndares(totalDeAndares);
    }

    public int getAndarAtual() {
        return andarAtual;
    }

    public void setAndarAtual(int andarAtual) {
        this.andarAtual = andarAtual;
    }

    public int getTotalDeAndares() {
        return totalDeAndares;
    }

    public void setTotalDeAndares(int totalDeAndares) {
        if (totalDeAndares > 0){
            this.totalDeAndares = totalDeAndares;
        }
    }

    public int getCapacidadeDoElevador() {
        return capacidadeDoElevador;
    }

    public void setCapacidadeDoElevador(int capacidadeDoElevador) {
        if (capacidadeDoElevador > 0){
            this.capacidadeDoElevador = capacidadeDoElevador;
        }
    }

    public int getPessoasElevador() {
        return pessoasElevador;
    }

    public void setPessoasElevador(int pessoasElevador) {
        this.pessoasElevador = pessoasElevador;
    }

    public void entrar() {
        if (pessoasElevador < capacidadeDoElevador) {
            pessoasElevador++;
        }
    }

    public void sair() {
        if (pessoasElevador > 0) {
            pessoasElevador--;
        }
    }

    public void sobe() {
        if (andarAtual < totalDeAndares) {
            andarAtual++;
        }
    }

    public void desce() {
        if (andarAtual > 0) {
            andarAtual--;
        }
    }

    public String imprimeDetalhe() {
        return "Pessoas no elevador: " + getPessoasElevador() + "\nAndar Atual: " + getAndarAtual();
    }

    public String toString() {
        return "Elevador [andarAtual=" + andarAtual + ", capacidadeDoElevador=" + capacidadeDoElevador
            + ", pessoasElevador=" + pessoasElevador + ", totalDeAndares=" + totalDeAndares + "]\n";
    }
}

```

```
Informe a Capacidade de pessoas no Elevador:
30
Informe o total de andares no Elevador:
5
Elevador [andarAtual=0, capacidadeDoElevador=30, pessoasElevador=0, totalDeAndares=5]
Digite o que deseja:
0-Sair
1-Entrar uma pessoa no elevador:
2-Sair uma pessoa do elevador
3-Subir Andar
4-Descer Andar
5-Imprimir D etalhes
5
Pessoas no elevador: 0
Andar Atual: 0
Digite o que deseja:
0-Sair
1-Entrar uma pessoa no elevador:
2-Sair uma pessoa do elevador
3-Subir Andar
4-Descer Andar
5-Imprimir D etalhes
```