

Nome:

Matrícula: \_\_\_\_\_

Disciplina: **ARA0075 / PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA**

Data: \_\_\_\_/\_\_\_\_/\_\_\_\_

Período: **2024.1 / SM1**Turma: **3059****Leia com atenção as questões antes de responder.**

É proibido o uso de equipamentos eletrônicos portáteis e consulta a materiais de qualquer natureza durante a realização da prova.

Boa prova.

**1.**\_\_\_\_\_ de **0,10**

Sobre o pilar de polimorfismo da orientação a objetos, assinale a alternativa correta.

- ☐ A Polimorfismo e herança são usados da mesma forma em Java, porém com assinaturas de métodos diferentes em tempo de compilação.
- ☒ B Na linguagem Java, o polimorfismo dinâmico é a execução da mesma operação da classe com métodos de assinaturas diferentes, e a escolha do método ocorre em tempo de compilação.
- ☐ C O conceito de polimorfismo implica no uso de métodos iguais em classes diferentes independente de sua assinatura.
- ☐ D Polimorfismo é um conjunto de métodos dinâmicos usados para que uma classe consiga acessar os atributos protegidos de outra classe.
- ☐ E Polimorfismo é representado pelo comando extends na linguagem Java.

**2.**\_\_\_\_\_ de **0,10**

Analisar as afirmativas sobre Classes e Objetos como conceitos-chave da Programação Orientada à Objetos:

I. uma classe é uma descrição de um ou mais objetos por meio de um conjunto uniforme de atributos e funcionalidades por estes desempenhadas.

II. uma classe é capaz de armazenar estados através de seus atributos e reagir a mensagens enviadas a ela, assim como se relacionar e enviar mensagens a outras classes.

III. uma classe é uma abstração de alguma coisa no domínio de um problema ou na sua implementação, refletindo a capacidade de um sistema para manter informações sobre ela e/ou interagir com esta classe.

IV. um objeto de uma classe é apenas uma definição, pois a ação só ocorre quando o objeto é invocado através de um método.

Está correto o que se afirma APENAS em:

- ☐ A Apenas II, III e IV.
- ☐ B I, II, III e IV
- ☐ C Apenas I.
- ☐ D Apenas I, II e III
- ☒ E Apenas I e III

**3.**\_\_\_\_\_ de **0,10**

Um desenvolvedor Java SE possui, em uma aplicação, duas subclasses B e C, estendendo a mesma superclasse A. Na superclasse A há um método `calcularValor`, também presente nas duas subclasses B e C, mas implementado de maneiras diferentes. A partir do método `main` de uma classe principal, utilizou-se o seguinte bloco de código:

```
A t1 = new A();
```

```
System.out.println(t1.calcularValor());
```

```
A t2 = new B();
```

```
System.out.println(t2.calcularValor());
```

```
A t3 = new C();
```

```
System.out.println(t3.calcularValor());
```

Após compilar e executar a aplicação, foram exibidos na tela os valores 4, 9 e 5. Nessa aplicação evidencia-se o uso de

- A ☐ métodos estáticos.
- B ☐ métodos abstratos.
- C ☐ sobrecarga de construtor.
- D ☐ sobrecarga de métodos.
- E ☒ polimorfismo.

4.

\_\_\_\_\_ de 0,10

Escolher entre as opções que apresenta 2 códigos Java que implementam threads.

- A ☒

```
public class MinhaThread implements Runnable {  
    public void run() {  
        //Código  
    }  
}
```

  

```
public class MinhaThread extends Thread {  
    public MinhaThread() {  
        super("MinhaThread");  
    }  
    public void run() {  
        //Código  
    }  
}
```
- B ☐

```
public class MinhaThread implements Executable {  
    public void run() {  
        //Código  
    }  
}
```

  

```
public class MinhaThread extends Thread {  
    public MinhaThread() {  
        super("MinhaThread");  
    }  
    public void run() {  
        //Código  
    }  
}
```
- C ☐

```
public class MinhaThread extends Runnable {  
    public void run() {  
        //Código  
    }  
}
```

  

```
public class MinhaThread implements Thread {  
    public MinhaThread() {  
        super("MinhaThread");  
    }  
}
```

```
    public void run() {  
        //Código  
    }  
}
```

☐ D public class MinhaThread implements MinhaThread {  
 public void run() {  
 //Código  
 }  
}

```
public class MinhaThread extends MinhaThread {  
    public MinhaThread() {  
        super("MinhaThread");  
    }  
    public void run() {  
        //Código  
    }  
}
```

☐ E public class MinhaThread implements Connection{  
 public void run() {  
 //Código  
 }  
}

```
public class MinhaThread extends DriverManager{  
    public MinhaThread() {  
        super("MinhaThread");  
    }  
    public void run() {  
        //Código  
    }  
}
```

5. \_\_\_\_\_ de 0,10

Sobre a programação orientada a objetos com Java, é correto afirmar:

- ☐ A Em uma hierarquia de herança, a superclasse herda todos os métodos públicos, privados e protegidos das subclasses
- ☐ B É obrigatório escrever em toda classe um construtor vazio que não recebe parâmetros. Além desse construtor, pode-se criar outros construtores vazios
- ☐ C No interior do método *main* de uma classe, mesmo os métodos públicos e estáticos de outra classe só podem ser acessados por meio de um objeto explicitamente instanciado dessa classe
- ☒ D Uma classe pode ter mais de um método com o mesmo nome, desde que receba parâmetros diferentes
- ☐ E Uma interface possui métodos com conteúdo implementado que não precisam ser reescritos nas classes que a implementam

6. \_\_\_\_\_ de 0,10

Considere as seguintes afirmativas:

- I - Uma variável é um elemento sintático do algoritmo que tem como funcionalidade armazenar um determinado valor.
- II - Uma Constante é um elemento sintático do algoritmo que tem como funcionalidade armazenar mais de um determinado valor.
- III - O valor associado à variável depende do tipo de informação que se deseja manipular.
- IV - O valor associado à constante não depende do tipo de informação que se deseja manipular.

Assinale a Alternativa Correta

- ☒ A I, II, III

- B ☒ I e III
- C ☐ II e IV
- D ☐ III
- E ☐ I

7.

\_\_\_\_\_ de 0,10

Assinale a alternativa que corresponde à API do Java responsável por prover acesso a bancos de dados.

- A ☒ JDBC.
- B ☐ Socket.
- C ☐ JRMP.
- D ☐ RMI.
- E ☐ SCTP.

8.

\_\_\_\_\_ de 0,10

Assinale a opção correta sobre o código que define corretamente o uso de herança em Java.

- A ☐

```
public abstract class Vestuario {  
    protected String cor;  
}  
public class Calca Implements Vestuario {  
    String tpCalca;  
}
```
- B ☐

```
public abstract class Vestuario {  
    protected String cor;  
}  
public class Calca abstract Vestuario {  
    String tpCalca;  
}
```
- C ☒

```
public abstract class Vestuario {  
    protected String cor;  
}  
public class Calca extends Vestuario {  
    String tpCalca;  
}
```
- D ☐

```
public final abstract class Vestuario {  
    protected String cor;  
}  
public class Calca extends Vestuario {  
    String tpCalca;  
}
```
- E ☐

```
public class Vestuario {  
    protected String cor;  
}  
public class Calca throws Vestuario {  
    String tpCalca;  
}
```

9.

\_\_\_\_\_ de 0,10

A programação orientada a objetos é composta por três elementos básicos, quais são eles?

- I. Classe
- II. Identidade
- III. Comportamento
- IV. Objeto
- V. Instância

- A ☒ I, IV e V
- B ☐ II, III e IV
- C ☐ II, IV e V
- D ☐ I, II e V
- E ☐ I, III e IV

10.

\_\_\_\_\_ de 0,10

Sabemos que quando há herança entre classes, as subclasses podem reescrever os métodos da superclasse. Então como se chama o processo quando uma subclasse reescreve um método da superclasse e mantém sua assinatura?

- A ☐ synchronized
- B ☐ polimorfismo
- C ☐ encapsulamento
- D ☐ overloading
- E ☒ overriding

Campus:  
POLO ALCÂNTARA - SÃO GONÇALO - RJ

Prova Impressa em 10/04/2024 por  
EDIBERTO MARIANO DA SILVA

Ref.: 6984159362

Prova Montada em 10/04/2024