

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 a Programação Orientado a Objeto(POO), dentre as características da linguagem de programa Java, marque a alternativa correta.

- ☐ A Pode ser executado em qualquer sistema operacional desde que tenha uma máquina virtual adequada.
- ☐ B Em java somente as variáveis estão dentro das classes. Os metodos são implementados durante a instancia da classe.
- ☐ C Em POO, uma classe é uma maneira de se criar objetos que possuem diferentes comportamentos e diferentes estruturas.
- ☐ D Java é não uma linguagem de programação puramente orientada a objetos, pois, com exceção de seus tipos primitivos de dados, tudo em Java são classes ou instâncias de classes (objetos).
- ☐ E Java difere da maioria das outras plataformas porque depende exclusivamente do hardware do computador para ser executado.

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

A reusabilidade de código é uma das vantagens de linguagens orientadas a objetos como Java. Escolha a opção correta em que temos a palavra reservada que permite o programador aplicar a reusabilidade de código.

- ☐ A public
- ☐ B class
- ☐ C import
- ☐ D extends
- ☐ E implements

**3.**\_\_\_\_\_ 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

- ☐ Uma interface possui métodos com conteúdo implementado que não precisam ser reescritos nas classes que a implementam

4.

\_\_\_\_\_ 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

- ☐ II, IV e V
- ☐ I, IV e V
- ☐ I, III e IV
- ☐ II, III e IV
- ☐ I, II e V

5.

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

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

- ☐ Polimorfismo e herança são usados da mesma forma em Java, porém com assinaturas de métodos diferentes em tempo de compilação.
- ☐ 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.
- ☐ O conceito de polimorfismo implica no uso de métodos iguais em classes diferentes independente de sua assinatura.
- ☐ Polimorfismo é um conjunto de métodos dinâmicos usados para que uma classe consiga acessar os atributos protegidos de outra classe.
- ☐ Polimorfismo é representado pelo comando extends na linguagem Java.

6.

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

Sobre a linguagem de programação Java, analise as assertivas e assinale a alternativa que aponta a(s) correta(s).

I. Nesta linguagem de programação, programas são construídos a partir de classes. A partir de uma definição de classe, podemos criar qualquer quantidade de objetos, que são conhecidos como instâncias daquela classe.

II. Uma classe, nesta linguagem de programação, contém membros, sendo campos e métodos as principais espécies. Campos são variáveis de dados que pertencem ou à própria classe ou a objetos da classe; eles constituem o estado do objeto ou classe.

III. Encontramos também, em uma classes Java, métodos. Métodos são coleções de comandos que operam sobre os campos para manipular o estado. Comandos definem o comportamento de classes; eles podem atribuir valores a campos e outras variáveis, avaliar expressões aritméticas, invocar métodos e controlar o fluxo de execução.

IV. Uma classe, nesta linguagem, pode ser compilada para bytecodes.

- ☐ Apenas II, III e IV.
- ☐ Apenas I, II e III
- ☐ Apenas I, III e IV

- ☐ D Apenas I
- ☐ E I, II, III e IV

7.

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

Na hierarquia de exceções em Java, é correto afirmar que

- ☐ A as classes NullPointerException e ArrayIndexOutOfBoundsException não são válidas no tratamento de exceção em Java.
- ☐ B a classe Error herda da classe Exception.
- ☐ C a classe Exception e a classe Error são subclasses da classe NullPointerException.
- ☐ D a classe RuntimeException é uma subclasse da classe Exception.
- ☐ E a classe Exception é uma subclasse da classe IOException.

8.

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

Analise 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

9.

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

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

- ☐ A 

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

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

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

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

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

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

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

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

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

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

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

