

Nome:

Matrícula: \_\_\_\_\_

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

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

Período: **2024,1 / SM1**Turma: **3001****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 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.

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

**2.**\_\_\_\_\_ de **0,10****Analise o seguinte trecho de programa e marque a alternativa que demonstra os valores impressos pelo programa:**

```
public class Avalia {  
    public static void main(String args[]) {  
        int i;  
        int v [] = {5,12,18,3,5,66,7,23,4,67};  
        for (i=1; i<10; i++){  
            if ( (i%2) == 0 )  
                System.out.println (v[i]);  
        }  
    }  
}
```

```
}  
}
```

- A ☐ 12 - 18 - 66 - 4  
B ☐ 12 - 3 - 66 - 23 - 67  
C ☐ 18 - 66 - 7 - 4  
D ☐ 5 - 18 - 5 - 7 - 4  
E ☒ 18 - 5 - 7 - 4

3.

\_\_\_\_\_ 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  
    }  
}
```

  

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

```
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
    }
}

```

**4.**

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

Qual a opção correta que demonstra o código mínimo que permite a conexão com um banco de dados na porta 5432 do SGBD Postgre.

☒ **A** `public Connection conectarBanco() {`  
 `try {`  
 `Class.forName("org.postgresql.Driver ");`  
 `String url = "jdbc:postgresql://localhost:5432/postgres";`  
 `Connection con = DriverManager.getConnection(url, "postgres", "12345");`  
 `return con;`  
 `} catch (ClassNotFoundException e) {`  
 `e.printStackTrace();`  
 `} catch (SQLException e) {`  
 `e.printStackTrace();`  
 `}`  
 `return null;`  
`}`  
`// fim da classe`

☐ **B** `public Connection conectarBanco() {`  
 `try {`  
 `Class.forName("org.postgresql.Driver ");`  
 `String url = "jdbc:postgresql://localhost:5432/postgres";`  
 `return con;`  
 `} catch (ClassNotFoundException e) {`  
 `e.printStackTrace();`  
 `} catch (SQLException e) {`  
 `e.printStackTrace();`  
 `}`  
 `return null;`  
`}`  
`// fim da classe`

☐ **C** `public Connection conectarBanco() {`  
 `try {`  
 `String url = "jdbc:postgresql://localhost:1111/postgres";`  
 `Connection con = DriverManager.getConnection(url, "postgres", "12345");`  
 `return con;`  
 `} catch (ClassNotFoundException e) {`  
 `e.printStackTrace();`  
 `} catch (SQLException e) {`

```

        e.printStackTrace();
    }
    return null;
}
} // fim da classe

```

☐ D public Connection conectarBanco() {

```

    try {
        Class ("org.postgresql.Driver ");
        String url = "jdbc:postgresql://localhost:5432/postgres";
        Connection con = DriverManager.getConnection(url, "postgres", "12345");
        return con;
    } catch (ClassNotFoundException e) {

        e.printStackTrace();
    } catch (SQLException e) {

        e.printStackTrace();
    }
    return null;
}
} // fim da classe

```

☐ E public Connection conectarBanco() {

```

    try {
        Class.forName("org.postgresql.Driver ");
        String url = "jdbc:postgresql://localhost:5432/postgres";
        Connection con = Statement.getConnection(url, "postgres", "12345");
        return con;
    } catch (ClassNotFoundException e) {

        e.printStackTrace();
    } catch (SQLException e) {

        e.printStackTrace();
    }
    return null;
}
} // fim da classe

```

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

Observe as afirmativas abaixo acerca de um ambiente de execução e as suas threads

- I. As threads dentro de um processo são protegidas umas das outras.
- II. Criar uma nova thread dentro de um processo existente é computacionalmente oneroso em relação a criar um processo.
- III. As threads dentro de um processo podem compartilhar dados e outros recursos convenientes e eficientemente, em comparação a processos distintos.
- IV. O chaveamento para uma thread diferente dentro de um mesmo processo é menos oneroso do que chavear entre threads pertencentes a processos diferentes.

Está correto apenas o que se afirma em

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

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

Estamos no auge da pandemia de covid 19 e ainda longe de uma solução definitiva para o problema. A principal arma para enfrentar essa situação é informação. Vamos supor que um programador está desenvolvendo um software que irá realizar a

coleta das mortes diárias por covid 19 que ocorrem em uma cidade. Todo dia ele insere de forma fixa, o dado de mortes. Supondo que até o momento há apenas seis amostras. Abaixo segue um trecho do programa desenvolvido.

```
public static void main(String[] args) {  
  
    int mortesCovid [] = {2, 35, 57, 80, 28, 31};  
  
    float soma = 0;  
    for (int i = 0; i  
        soma = soma + mortesCovid[i];  
    }  
  
    float media = soma / mortesCovid.length;  
  
    System.out.println("A média de mortes é: " + media);  
  
    System.out.println(mortesCovid[6]);  
  
}
```

Qual exceção será lançada caso esse programa fosse executado?

- A ☐ RuntimeException
- B ☐ IllegalAccessException
- C ☐ IndexOutOfBoundsException
- D ☒ ArrayIndexOutOfBoundsException.
- E ☐ Exception.

7.

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

(CONSULPLAN & TRF, 2017) (Adaptada) A grande maioria das aplicações, independente de porte, utiliza a persistência de dados atrelada a seus sistemas. O banco de dados se faz necessário em uma aplicação não só para persistir as informações, como também é preciso nos comunicarmos com ele para recuperar, modificar e apagar informações. Portanto, podemos afirmar que o gerenciamento desses dados é de fundamental importância para o correto funcionamento da aplicação. Quando se trata de persistência de dados em Java, geralmente a forma utilizada para guardar dados é um banco de dados relacional. Para abrir uma conexão com um banco de dados, precisamos utilizar sempre um driver. A classe DriverManager é responsável por realizar essa comunicação, o driver JDBC & Java Database Connectivity é a especificação de como a linguagem Java irá se comunicar com um banco de dados. O parâmetro que é passado é do tipo String contendo URL para localizar o banco de dados que, por sua vez, contém informações para conexão com o banco de dados. Assinale a alternativa correta que contém a URL padrão para a conexão com o banco MySQL, considerando que: usuário, servidor e porta do banco serão padrão, o banco não tem senha e o nome do banco de dados é teste.

- A ☐ ("mysql:jdbc://localhost:3306/teste","root","").
- B ☒ ("jdbc:mysql://localhost:3306/teste","root","").
- C ☐ ("mysql-jdbc://localhost:3306/teste","root","").
- D ☐ ("mysql:jdbc://localhost:3306/teste","","root").
- E ☐ ("jdbc:mysql://localhost:3306/teste","","root").

8.

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

O paradigma de programação orientada a objetos nasceu da necessidade de trazer o entendimento de problemas computacionais para mais próximo do mundo real. Levando em consideração os conceitos de programação orientada a objetos, julgue os itens a seguir:

I & Classes e objetos possuem atributos e métodos, no entanto, uma classe é apenas um modelo que é usado para criar objetos diferentes do mesmo tipo.

II & Os atributos de um objeto são o que ele sabe fazer e os métodos são o que ele sabe.

III - 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.

Está(ão) correta(s):

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

9.

\_\_\_\_\_ 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 ☐ class
- B ☒ extends
- C ☐ public
- D ☐ implements
- E ☐ import

10.

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

Na linguagem Java, a palavra-chave que deve ser usada para especificar que uma variável não pode ser modificada é

- A ☐ const
- B ☒ final
- C ☐ public
- D ☐ constant
- E ☐ static