A+

Alterar modo de visualização

Peso da Avaliação 3,00
Prova 48954499
Qtd. de Questões 10
Acertos/Erros 3/7
Nota 3,00

1 A estrutura de uma classe na linguagem de programação Java comporta um nome, um conjunto de atributos, também conhecidos como o estado desta classe e um conjunto de métodos, também conhecidos como o comportamento desta classe. Considerando a estrutura da Classe Principal demonstrada na figura anexa, classifique V para as sentenças verdadeiras e F para as falsas:

( ) Já que o método "Aluno()" não possui parâmetros as linhas 7 a 9 geram erros de compilação.
( ) O método chamado main é quem torna uma classe executável, sendo que a sua existência em uma classe é opcional.
( ) A linha 1 define, através da palavra reservada "package", que a classe está agrupada obedecendo as suas responsabilidades dentro de uma aplicação.
( ) Pode-se afirmar que o tipo de dados do atributo "nome", existente na classe aluno é do tipo alfanumérico.

```
1  package basico;
2
3  public class Principal {
4
5      public static void main(String[] args){
6          Aluno a = new Aluno();
7          a.matricula = 12;
8          a.nome = "Catarina";
9          a.calcularMedia();
10     }
11 }
```

Assinale a alternativa que apresenta a sequência CORRETA:

A V - F - F - V.

B V - F - V - F.

C F - V - V - V.

D F - V - F - V.

2 A herança, juntamente ao encapsulamento, o polimorfismo e a abstração, representa um dos pilares da programação orientada a objetos e pode, quando bem modelada e utilizada, diminuir o esforço na manutenção do código-fonte. Considerando o relacionamento de herança entre duas ou mais classes, assinale a alternativa CORRETA:

A A palavra extends define a herança na linguagem de programação Java.

B Com a utilização do recurso de herança é possível compartilhar atributos entre uma classe pai e uma classe filha, porém esse compartilhamento não é possível no tocante aos métodos.

C A herança de métodos é possível, porém os métodos do tipo "get" e "set" devem ser reescritos

através do polimorfismo.

D Com a utilização do recurso de herança é possível compartilhar métodos entre uma classe pai e uma classe filha, porém esse compartilhamento não é possível no tocante aos atributos.

3 Assim como na programação procedural, a programação orientada a objetos tem recursos para tratamentos de erros. Neste sentido, a plataforma Java possui recursos robustos que tratam os erros que podem acontecer em tempo de execução, tanto para exceções verificadas quanto para as não verificadas. É importante que o programador trate as possíveis exceções que podem ocorrer, pois, caso contrário, elas estourarão no usuário. Partindo desse pressuposto, classifique V para as sentenças verdadeiras e F para falsas:

( ) Na linguagem de programação Java, podemos tratar as exceções através dos comandos "try{} catch(){}".
( ) Na linguagem de programação Java, quanto um comando tenta acessar um índice inválido de um ArrayList, será retornada a exceção "ArrayIndexOutOfBoundException".
( ) As exceções não verificadas podem ser tratadas através de validações no código-fonte, evitando a ocorrência de erros.
( ) Uma exceção nunca pode ser delegada pelo método que invocou um determinado método, o qual pode sofrer uma exceção, devendo ser tratado sempre no próprio método.

Assinale a alternativa que apresenta a sequência CORRETA:

A F - V - F - F.

B V - F - V - F.

C V - V - V - F.

D V - F - F - V.

4

A cidade de São Paulo, que possuía uma população de 10.000.000 de habitantes, teve um aumento de mais 2.000.000 de novos habitantes. Na associação dessa afirmação aos conceitos da modelagem orientada a objetos, é correto afirmar que São Paulo, população e aumento, referem-se, respectivamente, a quê?

A Objeto, instância, operação.

B Classe, objeto, instância de classe.

C Classe, objeto, atributo.

D Objeto, atributo, implementação por um método do objeto.

5 O padrão de projeto Singleton aumenta a qualidade e a produtividade do desenvolvimento de software orientado a objetos, pois é capaz de solucionar problemas rotineiros que normalmente ocorrem durante as etapas do desenvolvimento de um software. Esse padrão descreve uma implementação na qual uma classe é instanciada uma única vez durante a execução de uma aplicação. Referente aos códigos-fonte que implementam um exemplo da aplicação Singleton e sua execução para a classe Memoria, classifique V para as sentenças verdadeiras e F para as falsas:

```
(1)public class Memoria{
(2)   private static Memoria memoryInstance;
(3)   private double quantidadeMemoria;
(4)
(5)   private Memoria(){}
(6)
(7)
(8)   public static Memoria getInstance() {
(9)         if(memoryInstance == null) {
(10)              memoryInstance = new Memoria();
(11)        }
(12)        return memoryInstance;
(13) }
(14)
(15) public double getQuantidadeMemoria() {
(16)        return this.quantidadeMemoria;
(17) }
(18)
(19) public void setQuantidadeMemoria (float quantidadeMemoria) {
(20)        this.quantidadeMemoria = quantidadeMemoria;
(21) }
(22)}
```

```
(1)   public static void main(String[] args){
(2)
(3)      Memoria memoria1 =  Memoria.getInstance();
(4)      System.out.println(memoria1);
(5)   }
```

( ) Na linha 3, do código-fonte de execução da classe Memoria, o trecho Memoria.getInstance() cria, e retorna, um objeto único da classe Memoria.
( ) A linha 5, do código-fonte de classe Memoria, utiliza um modificador de visibilidade igual ao padrão usado nas implementações de classes Java, e é usado como um recurso de segurança do padrão Sigleton.
( ) A linha 3, do código-fonte de execução da classe Memoria, poderia ser substituído pelo código Memoria memoria1 = new Memoria(), sem qualquer prejuízo para o funcionamento chave do padrão Singleton.
( ) A linha 5, do código-fonte da classe Memoria, utiliza um modificador de visibilidade que foge ao padrão usado nas implementações de classes Java, uma vez que complementa a capacidade do padrão Singleton em permitir a criação de apenas um objeto para as classes as quais os comportamentos necessitam dessa condição.

Assinale a alternativa que apresenta a sequência CORRETA:

A V - F - F - V.

B F - F - V - V.

C F - V - V - F.

F - V - V - F.

D V - V - F - F.

6 A sobrecarga de métodos, considerada um tipo de polimorfismo, é um mecanismo que permite que dois ou mais métodos compartilhem o mesmo nome, contanto que tenham diferentes conjuntos de parâmetros. Dessa forma, com o código-fonte apresentado, referente à classe Bicicleta, analise as sentenças a seguir sobre a correta sintaxe aplicada ao polimorfismo de sobrecarga para os dois métodos construtores da referida classe:

```
public class Bicicleta{
	private char tamanhoQuadro;
	private int numeroMarchas;
	private char tamanhoRoda;

	public void movimentar(){
		System.out.println("O objeto bike está em movimento!!");
	}
}
```

I-

```
public Bicicleta(char tamanhoQuadro, int numeroMarchas){
		this.tamanhoQuadro = tamanhoQuadro;
		this.numeroMarchas = numeroMarchas;
}

public Bicicleta(){
		this.tamanhoQuadro = tamanhoQuadro;
		this.numeroMarchas = numeroMarchas;
}
```

II-

```
public Bicicleta(char tamanhoQuadro, int numeroMarchas, char tamanhoRoda){
		this.tamanhoQuadro = tamanhoQuadro;
		this.numeroMarchas = numeroMarchas;
		this.tamanhoRoda = tamanhoRoda;
}
public Bicicleta(){
}
```

III-

```
public Bicicleta(char tamanhoQuadro, int numeroMarchas, char tamanhoRoda){
		this.tamanhoQuadro = tamanhoQuadro;
		this.numeroMarchas = numeroMarchas;
		this.tamanhoRoda = tamanhoRoda;
}

public Bicicleta(int numeroMarchas, char tamanhoRoda){
		this.tamanhoQuadro = tamanhoQuadro;
		this.numeroMarchas = numeroMarchas;
		this.tamanhoRoda = tamanhoRoda;
}
```

IV-

```
public Bicicleta(int numeroMarchas){
```

```
        this.numeroMarchas = numeroMarchas;
}
public Bicicleta(char tamanhoQuadro, int numeroMarchas){
        this.tamanhoQuadro = tamanhoQuadro;
        this.numeroMarchas = numeroMarchas;
}
```

Assinale a alternativa CORRETA:

A As sentenças II e III estão corretas.

B As sentenças III e IV estão corretas.

C As sentenças I e II estão corretas.

D As sentenças II e IV estão corretas.

7 Ao utilzarmos o modificador "static", indicamos que determinado atributo ou método de uma classe Java pertence à classe e não a uma instância específica. Com relação a esse modificador, classifique V para as sentenças verdadeiras e F para as falsas:

( ) Um atributo com visibilidade estática não pode ser alterado com um método não estático.
( ) O Singleton é um padrão de projeto que prevê uma instância estática.
( ) Ao alterarmos o valor de um atributo estático, todas as instâncias da classe terão o seu valor alterado.
( ) Atributos estáticos inicializados no método construtor não podem ser alterados ao longo da execução.

Agora, assinale a alternativa que apresenta a sequência CORRETA:

A V - F - F - V.

B F - F - V - V.

C V - V - V - F.

D F - V - F - V.

8

Os conceitos da programação orientada a objetos (POO) surgiram no final da década de 1960, quando a linguagem Simula-68 introduziu as ideias de objetos e troca de mensagens para construção de programas. Tais concepções foram posteriormente amadurecidas e aprimoradas durante a década de 1970 pela linguagem de programação Smalltalk. No entanto, a popularização da POO só se deu ao longo das décadas de 1980 e 1990, com as linguagens C++ e Java.

Sobre a definição de classe na POO, assinale a alternativa CORRETA:

A Um modelo ou molde de construção de objetos, em que não se podem definir características.

B Um objeto com função de instanciação, em que não se podem definir características

C Um objeto com função de instanciação, em que se definem comportamentos por meio de atributos.

D Um modelo ou molde de construção de objetos, em que se definem características e comportamentos.

9 Diz-se que no paradigma da orientação a objetos, os objetos podem criar relações entre si, e duas maneiras de se estabelecer uma relação entre objetos são através da associação e da herança que devem ser utilizados na solução de problemas diferentes. Dessa forma, sobre esses relacionamentos, assinale a alternativa CORRETA:

A Nos relacionamentos de associação, o conceito de subclasse é utilizado de maneira diferente do que nos relacionamentos de herança.

B Os dois tipos de relacionamento são implementados na linguagem de programação Java através de palavras reservadas.

C Um relacionamento que responde de maneira positiva à pergunta "é um tipo de" representa uma associação ou uma herança.

D Uma associação não permite o compartilhamento de métodos e de atributos entre classes.

10 Uma das funções do tratamento de exceções da linguagem de programação Java é informar ao desenvolvedor que determinado código fonte está tentando acessar um recurso fora da “sandbox” fornecida pela Máquina Virtual Java. Recursos como rede, disco etc. são de controle do sistema operacional e existem tipos especiais de exceções para garantir que o código continue executando em caso de falhas no acesso a estes recursos. Com relação às características do tratamento de exceções na plataforma Java, classifique V para as sentenças verdadeiras e F para as falsas:

( ) Essencialmente, uma exceção é causada por uma instrução que não consegue ser executada.
( ) O tratamento de exceções deixa o código fonte mais performático e, por consequência, mais robusto.
( ) Pode-se tratar exceções de forma local, através do bloco try catch ou delegar o tratamento, através da cláusula throws.
( ) As exceções não verificadas são aquelas que lidam com problemas ao acessar recursos externos à JVM.

Agora, assinale a alternativa que apresenta a sequência CORRETA:

A V - F - F - V.

B V - V - F - V.

C V - F - V - F.

D F - V - F - F.

Imprimir