

Em um sistema de gerenciamento de biblioteca desenvolvido em Java, cada livro é representado por um objeto. Estes objetos são instanciados a partir de uma classe Livro, que possui atributos como título, autor e ano de publicação.

Qual princípio da programação orientada a objetos é primordialmente demonstrado neste cenário?

A

Herança.

~~B~~

~~Abstração.~~

C

Encapsulamento.

D

Polimorfismo.

E

Recursão.

Em um projeto Java, um desenvolvedor precisa implementar um método para calcular a média de uma lista de notas. Ele deve utilizar um laço de repetição adequado para percorrer a lista, somar as notas e, ao final, calcular a média.

Qual é o laço de repetição mais apropriado para esta tarefa e por quê?

☐ A do-while, para garantir que a lista seja percorrida pelo menos uma vez.

☐ B switch-case, pois pode selecionar diferentes operações com base no valor de cada nota.

☐ C try-catch, para tratar possíveis erros durante a iteração.

☒ D for, pois permite iterar a lista de forma concisa.

☐ E if-else, para verificar se cada nota é válida antes de somá-la.

Analise o trecho de código Java a seguir.

	0	1	2	3	4
0	0	0	0	0	0
1	0	0	r=1	0	0
2	0	p=1	q=1	t=1	0
3	0	0	0	0	0
4	0	0	0	0	0

A partir da análise do código, observa-se que:

- ☐ A Será exibido um texto no console.
- ☒ B O código não compila.
- ☐ C O código compila, mas não mostra nada.
- ☐ D Os modificadores de acesso não podem estar ocultos.
- ☐ E Será exibido o endereço de memória do objeto.

Ano: 2009 Banca: INSTITUTO CIDADES Órgão: UNIFESP Prova: INSTITUTO CIDADES - 2009 - UNIFESP - Analista de Tecnologia da Informação (ADAPTADA)

Acerca do método equals, analise as seguintes afirmativas:

I. O equals recebe um Object como argumento e verifica se ele mesmo é igual ao Object recebido para retornar um boolean exclusivamente false.

II. Um exemplo do uso do equals é para datas. Se usarmos criar duas datas, isto é, dois objetos diferentes, contendo 18/08/2009, ao comparar com o ==, receberá false, pois são referências para objetos diferentes.

III. Quando comparamos duas variáveis referência no Java, o == verifica se as duas variáveis se referem ao mesmo objeto.

Podemos afirmar corretamente que:

☐ A Apenas a afirmativa III está correta.

☒ B As afirmativas II e III estão corretas.

☐ C As afirmativas I e II estão corretas.

☐ D Apenas a afirmativa II está correta.

☐ E Apenas a afirmativa I está correta.

Na implementação de um sistema bancário, a classe Conta serve como base para ContaCorrente e ContaPoupanca. Ambas as subclasses sobrescrevem o método calcularJuros, definido em Conta. Qual é um exemplo de sobrescrita de método no cenário descrito?

- ☐ A CalcularJuros é uma interface implementada por ContaCorrente e ContaPoupanca.
- ☒ B ContaCorrente e ContaPoupanca têm seus próprios métodos calcularJuros, distintos de Conta.
- ☐ C Conta chama o método calcularJuros de suas subclasses.
- ☐ D O método calcularJuros é exclusivo de ContaCorrente e ContaPoupanca.
- ☐ E ContaCorrente e ContaPoupanca usam o método calcularJuros de Conta sem alterações.

Um jogo de computador utiliza a classe Personagem como base para diferentes tipos de personagens, como Guerreiro e Mago. Cada subclasse possui métodos únicos, como atacar para Guerreiro e conjurar para Mago. Qual característica da programação orientada a objetos é exemplificada pela presença de métodos únicos em Guerreiro e Mago?

- ☐ A Polimorfismo, permitindo que Guerreiro e Mago sejam tratados como Personagem.
- ☐ B Sobrecarga de métodos, onde métodos com o mesmo nome têm parâmetros diferentes.
- ☐ C Herança, com Guerreiro e Mago herdando a estrutura de Personagem.
- ☒ D Especialização, onde subclasses adicionam métodos e atributos específicos.
- ☐ E Encapsulamento, através da ocultação de detalhes internos das subclasses.

Com base no código Java a seguir, responda:

```
public class Automovel {  
  
    public void nome(String nome) {  
  
        return nome;  
  
    }  
  
}  
  
public class Carro extends Automovel {  
  
    @Override public void nome(String nome) {  
  
        return "O carro é " + super.nome(nome);  
  
    }  
  
}  
  
public class Executa { public static void main(String[] args) {  
  
    Carro carro = new Carro();  
  
    System.out.println(carro.nome("Corsa"));  
  
}  
  
}
```

☐ A O carro é null

☐ B null

☒ C O código não será compilado, pois apresenta erros

☐ D O carro é Corsa

☐ E Corsa

Qual das seguintes alternativas indica o resultado da execução do código?

(VUNESP - 2020 - FITO - Técnico em Gestão - Informática)

No Java, a classe que implementa uma lista ordenada que permite a inclusão de elementos nulos (null) é:

☐ A List

☐ B Collection

☐ C HashSet

☒ D ArrayList

☐ E HashMap

Ano: 2017 Banca: FCC Órgão: TRE-PR Prova: FCC - 2017 - TRE-PR - Técnico Judiciário - Programação de Sistemas

Uma aplicação Java pode conter, entre outros recursos, classes, interfaces e classes abstratas. Uma instrução correta que uma interface Java pode conter é:

☐ A Public void exibe() {System.out.print("Ok");}

☐ B Private int calcularQuantidade(int n1);

☐ C Protected sacar(double valor);

☒ D Double calcular(double n1, double n2);

☐ E Public double calcularTributos() {return 27.5;}

Ano: 2015 Banca: CETAP Órgão: MPC-PA Prova: CETAP - 2015 - MPCM-PA - Técnico em Informática - Desenvolvimento

A herança entre classes, em Java, define-se utilizando a palavra-chave:

☐ A References

☐ B Matches

☒ C Extends

☐ D Implements

☐ E Inherits

(COMPERVE -UFRN- Analista de Tecnologia da Informação - 103 - 2018 - modificada)

Considere as seguintes afirmativas sobre os tipos de exceções que existem na linguagem Java e o que caracteriza cada um desses tipos.

I - Todas as exceções em Java herdam direta ou indiretamente da classe Throwable.

II - Existem dois tipos de exceções Java: (1) exceções checadas (checked exceptions) que herdam da classe Exception ou de outra exceção checada e (2) exceções não checadas que herdam da classe Throwable.

III - Um exemplo de exceção não checada é a do tipo Error que geralmente representa restrições de recursos ou problemas internos à máquina virtual Java (JVM).

IV - O compilador Java não permite que sejam definidos tratadores (cláusulas catch) para as exceções de tipo RuntimeException e Error.

Estão corretas as afirmativas

☐ A I e IV

☒ B I e III

☐ C II e IV

☐ D III e IV

☐ E II e III

(UNIRIO - Técnico de Tecnologia da Informação -2019)

Em Java, a palavra-chave throws é usada para:

- ☐ A Causar uma exceção e fornecer seus argumentos.
- ☐ B Especificar os métodos que tratarão uma dada exceção.
- ☒ C Especificar os tipos de exceções que um dado método pode causar.
- ☐ D Causar a exceção que está sendo tratada.
- ☐ E Especificar um código a ser executado, após qualquer tratamento de exceção no bloco.

(INSTITUTO AOCP -IBGE -Analista Censitário -Análise de Sistemas -Desenvolvimento de Aplicações Web Mobile -2019)

Ao se implementar programas de computador em Java, há possibilidade de ocorrerem erros imprevistos durante sua execução. Esses erros são conhecidos como exceções. Sabendo disso, considere que Márcio está fazendo uma operação em um banco de dados, porém essa operação dispara uma exceção de que o campo que ele está manipulando para inserir nesse banco está vazio, ou seja, acontece um `NullPointerException`. O problema, nesse caso, é a conexão com o banco de dados ficar aberta, o que não pode acontecer. Assim, assinale a alternativa que apresenta o bloco/código no qual deve ser inserida essa operação de fechamento da conexão com o banco de dados.

☐ A if

☐ B catch

☒ C finally

☐ D else

☐ E try

(CS-UFG - SANEAGO-GO- Analista de Sistemas - 2018 - modificada)

Uma desenvolvedora JAVA chamou um método X que pode gerar uma exceção não verificada. Porém, essa chamada não foi circundada por um bloco de tratamento de exceção. O que acontecerá com esse código?

- ☐ A Compilará e as exceções serão ignoradas.
- ☒ B Compilará e executará normalmente, caso a exceção não seja lançada por X.
- ☐ C Compilará e lançará exceção, caso o método X seja chamado.
- ☐ D Não compilará, pois o JAVA exige tratamento de exceção.
- ☐ E Não compilará, pois o JAVA não suporta exceções não verificadas.

(AOCP - BRDE - Analista de Sistemas - 2012) Sobre Prioridade e Agendamento de Thread na linguagem de programação Java, analise as afirmações e assinale a(s) correta(s).

I. Por padrão, toda thread recebe uma `DEFAULT_PRIORITY` (uma constante de 5).

II. Toda thread do Java tem uma prioridade de thread que auxilia na ordenação em que estão agendadas.

III. O agendamento de thread consegue garantir que, em diferentes sistemas operacionais, cada processo gastará a mesma fatia de tempo.

IV. Em Java, cada nova thread herda a prioridade da thread que a cria.

☐ A Apenas I

☒ B Apenas II e IV

☐ C Apenas I, III e IV

☐ D Apenas II, III e IV

☐ E I, II, III e IV

Em Java, quando uma thread está aguardando para obter a trava de um semáforo, ela encontra-se no estado:

☐ A TIMED_WAITING

☐ B STOPPED

☐ C RUNNABLE

☒ D BLOCKED

☐ E WAITING

No que se refere à linguagem de programação Java, "thread safe" é um conceito segundo o qual threads concorrentes utilizam algoritmo de sincronismo para cooperarem entre si. É exemplo da aplicação desse conceito:

☐ A Classes estáticas

☐ B Objetos finais

☐ C Classes finais

☐ D Objetos estáticos

☒ E Objetos imutáveis

Após instanciar os drivers JDBC no aplicativo Java, precisamos estabelecer a conexão com o banco de dados e, a partir da obtenção da conexão, instanciar os objetos que serão responsáveis por executar as instruções SQL. Selecione a classe, pertencente à API JDBC, que administra e seleciona o driver adequado para iniciar uma conexão com o banco de dados na linguagem Java.

☒ A DriverManager

☐ B Statement

☐ C ResultSet

☐ D PreparedStatement

☐ E Connection

Considere as linhas apresentadas a seguir, extraídas de um método em uma aplicação Java SE que acessa um banco de dados MySQL.

```
Class.forName("com.mysql.jdbc.Driver");  
Connection con = DriverManager.getConnection(  
    ..... "jdbc:mysql://localhost:3307/b", "r", "p");
```

Ao executar as linhas de código podem ocorrer, respectivamente, as seguintes exceções:

- ☐ A InterruptedException e SQLException
- ☒ B ClassNotFoundException e SQLException
- ☐ C IOException e IllegalArgumentException
- ☐ D InterruptedException e NullPointerException
- ☐ E ClassNotFoundException e NullPointerException

Com o surgimento do DAO foi possível padronizar a forma com que as operações sobre o banco de dados são implementadas, o que levou à definição de ferramentas como o JPA, onde a obtenção de uma coleção de entidades, a partir da consulta através de sintaxe JPQL, ocorre com base no método:

☐ A createResultList

☐ B getEntityManager

☐ C createEntityManager

☐ D createNamedQuery

☒ E getResultList