

Programação Orientada a Objetos - COM230 - Turma 012

Página Inicial

Avisos

Cronograma

Atividades

Fóruns

Calendário Lives

Colaborate

Notas

Menu das Semanas

Semana 1

Semana 2

Semana 3

Semana 4

Semana 5

Semana 6

Semana 7

Semana 8

Orientações para realização da prova

Orientações para realização do exame

Documentos e informações gerais

Gabaritos

Referências da disciplina

Repositório de REAs

Revisar envio do teste: Semana 6 - Atividade Avaliativa

Usuário

LIZIS BIANCA DA SILVA SANTOS

Curso

Programação Orientada a Objetos - COM230 - Turma 012

Teste

Semana 6 - Atividade Avaliativa

Iniciado

15/11/23 19:43

Enviado

15/11/23 20:18

Data de vencimento

17/11/23 05:00

Status

Completaada

Resultado da tentativa

10 em 10 pontos

Tempo decorrido

35 minutos

Instruções

Olá, estudante!

1. Para responder a esta atividade, selecione a(s) alternativa(s) que você considerar correta(s);

2. Após selecionar a resposta correta em todas as questões, vá até o fim da página e pressione "Enviar teste".

3. A cada tentativa, você receberá um conjunto diferente de questões.

Pronto! Sua atividade já está registrada no AVA.

Resultados exibidos

Todas as respostas, Respostas enviadas, Respostas corretas, Comentários, Perguntas respondidas incorretamente

Pergunta 1

1,42 em 1,42 pontos

Considere o código a seguir que apresentará erro se for executado sem o tratamento de exceções. Assinale a alternativa correta que lista quais são os tipos de exceções geradas ao executar o código.

```
import java.io.IOException;
public class TesteException {

    public static void main(String[] args) {

        int[] numeros = {4, 8, 16, 32, 64, 128};
        int[] denominador = {2, 0, 4, 8, 0};

        for (int i = 0; i < numeros.length; i++) {
            System.out.println(numeros[i] + "/" + denominador[i] + " = " + (numeros[i]/denominador[i]));
        }
    }
}
```

Resposta Selecionada:

ArithmeticException e ArrayIndexOutOfBoundsException.

Respostas:

NullPointerException e ArrayIndexOutOfBoundsException.

ArithmeticException e ArrayIndexOutOfBoundsException.

NullPointerException.

ArithmeticException e NullPointerException.

ArrayIndexOutOfBoundsException.

Comentário da resposta:

A resposta correta é: "ArithmeticException e ArrayIndexOutOfBoundsException."

Justificativa

Dois são os tipos de exceções: 1) erro ao dividir por zero e 2) acesso a uma posição inválida do array.

Pergunta 2

1,42 em 1,42 pontos

Uma exceção é uma indicação de algum problema na execução de um programa de computador. Analise as afirmações a seguir e escolha a alternativa correta.

I. É mais vantajoso usar o tratamento de erros usando as condições (if).

II. O tratamento de exceções deixa o código mais legível e eficiente.

III. NullPointerException ocorre quando se tenta fazer coerção equivocada de um objeto.

IV. O tratamento de exceções em Java é feito somente por try/catch, de modo que todas as exceções lançadas por um programa necessitem obrigatoriamente ser capturadas e tratadas.

Resposta Selecionada:

Apenas II está correta.

Respostas:

Apenas II está correta.

Apenas I está correta.

Apenas I e II estão corretas.

Apenas III e IV estão corretas.

Apenas III está correta.

Comentário da resposta:

A resposta correta é: "Apenas II está correta."

Justificativa

Usar a condição if para tentar barrar todas as possibilidades de erros no código, além de ser ineficiente, prejudica muito a legibilidade do código. NullPointerException ocorre em uma referência null usada quando se espera um objeto. Nem toda exceção precisa ser capturada e tratada, sendo que algumas delas podem ser ignoradas. Se elas forem lançadas e não forem tratadas, elas vão causar a interrupção do programa.

Pergunta 3

1,48 em 1,48 pontos

A linguagem Java dispõe do bloco *try-catch* e da cláusula *throws*, usados para o tratamento de exceções. Nesse sentido, analise as asserções a seguir e a relação proposta entre elas.

I. O bloco *try* contém um conjunto de instruções em que uma exceção pode ocorrer.

II. É sempre obrigatório um bloco *try* ser seguido por um bloco *catch*, que trata a exceção, caso ela ocorra.

Analizando essas asserções, é correto afirmar que:

Resposta Selecionada:

e, as duas asserções são verdadeiras, mas a segunda não justifica a primeira.

Respostas:

a, as duas asserções são falsas.

b, a primeira asserção é falsa e a segunda é verdadeira.

c, as duas asserções são verdadeiras e a segunda justifica a primeira.

d, a primeira asserção é verdadeira e a segunda é falsa.

e, as duas asserções são verdadeiras, mas a segunda não justifica a primeira.

Comentário da resposta:

JUSTIFICATIVA
A proposição I está correta, pois o bloco *try* deve conter as instruções em que uma exceção pode ocorrer. A proposição II está correta, mas não justifica a primeira, pois o bloco *try* sempre é seguido por um bloco *catch*, para tratar a exceção que ocorre no bloco *try* associado. O termo "*catch*" é um complemento para o termo "*try*", e não uma justificativa.

Pergunta 4

1,42 em 1,42 pontos

Na linguagem Java, é comum o tratamento de exceções, sendo esse um de seus recursos mais importantes, pois permite tratar os erros, em tempo de execução, causados por exceções.

Em relação ao tratamento de exceções, assinale a alternativa que apresenta, corretamente, o melhor uso das exceções em linguagem Java.

Resposta Selecionada:

b. Uma exceção é um evento indesejado que interrompe o fluxo normal do programa.

Respostas:

a. Quando ocorre um erro de sintaxe na programação, há uma exceção.

b. Uma exceção é um evento indesejado que interrompe o fluxo normal do programa.
Para tratar uma exceção, utilizamos as estruturas if-then-else.

c.

d. Uma exceção é um evento desejado que permite seguir o fluxo normal do programa.

e. Uma exceção é uma instrução executada quando uma condição não é atendida.

Comentário da resposta:

JUSTIFICATIVA
Quando ocorre uma exceção, a execução do programa é encerrada; para que isso não ocorra, é necessário o tratamento. Ao lidarmos com as exceções, podemos fornecer uma mensagem significativa ao usuário sobre o problema, em vez de uma mensagem gerada pelo sistema, que pode não ser compreendida pelo usuário. Assim, uma exceção não ocorre devido ao não atendimento de condições, muito menos por erros de sintaxe. Quando ocorrem erros de sintaxe, o compilador já detecta e impede a execução do programa. As exceções têm a característica de interromper o fluxo normal do programa, e o tratamento das exceções ocorre com o uso das estruturas try-catch.

Pergunta 5

1,42 em 1,42 pontos

Um dos recursos mais importantes da linguagem Java e que diferenciam essa linguagem de muitas outras é o tratamento de exceções.

Assinale a alternativa que apresenta, corretamente, os comandos para o tratamento de exceções em linguagem Java.

Resposta Selecionada:

c. try-catch

Respostas:

a. for-each

b. if-error

c. try-catch

d. new Object()

e. if-then-else

Comentário da resposta:

JUSTIFICATIVA
Na linguagem Java, o tratamento das exceções ocorre com os comandos try-catch e try-catch-finally. As estruturas if-then-else são utilizadas no fluxo normal do programa como estruturas condicionais. O for-each é utilizado para estruturas de repetição. O new Object() é utilizado para instanciar um novo objeto, e o comando if-error não existe na linguagem Java.

Pergunta 6

1,42 em 1,42 pontos

A linguagem Java tem um mecanismo de tratamento de exceções simples e organizado em três categorias. Nesse sentido, avalie as afirmativas a seguir, em relação aos três tipos de exceções.

1 – *Exception* (*checked*).

2 – *RuntimeException* (*unchecked*).

3 – *Error* (*unchecked*).

I. Representam erros internos na aplicação, mas que, em geral, não são tratáveis pelo programador.

II. Representam erros externos à aplicação, que não são contornáveis pelo programador.

III. São situações excepcionais em um programa, em tempo de execução, que podem ser contornadas.

Assinale a alternativa que relaciona, adequadamente, os dois grupos de informações.

Resposta Selecionada:

a. 1-III; 2-I; 3-II.

Respostas:

a. 1-III; 2-I; 3-II.

b. 1-I; 2-II; 3-III.

c. 1-II; 2-I; 3-III.
1-I; 2-III; 3-II.

d.

e. 1-III; 2-II; 3-I.

Comentário da resposta:

JUSTIFICATIVA
A sentença 1 se enquadra no conceito III, pois o tipo *Exception* (*checked*) é uma exceção que ocorre em tempo de compilação e que não pode ser ignorada, caso contrário, é gerado um erro de compilação. A sentença 2 se enquadra no conceito I, pois o tipo *RuntimeException* (*unchecked*) não gera erros de compilação e ocorre durante a execução do programa e podem ser tratados pelo programador. A sentença 3 se enquadra no conceito II, pois o tipo *Error* (*unchecked*) representa erros externos à aplicação, que não são contornáveis pelo programador, nesse caso, o Java imprime uma mensagem de erro e termina o programa.

Pergunta 7

1,42 em 1,42 pontos

Analise o código a seguir sobre as exceções em linguagem Java (qualquer versão).

```
public class ExemploException {

    public static void testException() {
        String[] nomes = { "João", "Maria", "José" };

        for (int i = 0; i < 4; i++) {
            System.out.println(nomes[i]);
        }
    }

    public static void main(String[] args) {
        testException();
    }
}
```

Em relação ao código apresentado, assinale a alternativa correta.

Resposta Selecionada:

d. Trata-se de uma exceção do tipo *RuntimeException* (*unchecked*).

Respostas:

a. Trata-se de uma exceção do tipo ArithmeticException .

b. Trata-se de uma exceção do tipo IOException .

c. O programa não tem erros de sintaxe e não gera uma exceção.

d. Trata-se de uma exceção do tipo *RuntimeException* (*unchecked*).

e. Trata-se de uma exceção que ocorre durante a compilação.

Comentário da resposta:

JUSTIFICATIVA
A exceção gerada pelo programa apresentado é do tipo RuntimeException, especificamente do tipo ArrayIndexOutOfBoundsException, pois ocorre em tempo de execução, ao se tentar buscar um índice inexistente no vetor "nomes". O vetor tem apenas três elementos, e o laço de repetição exige quatro.

Quinta-feira, 15 de Agosto de 2024 21h57min40s BRT

← OK