**Instituto Federal de Educação, Ciência e Tecnologia do Ceará**

**Campus Maracanaú**

**Coordenadoria de Computação**

**Curso de Bacharelado em Ciência da Computação**

**Disciplina: Programação Orientada a Objetos**

**Professor: Igor Rafael Silva Valente**

# ATIVIDADE

**Assunto:**

Exceções – fundamentos, captura e tratamento.

**Orientações:**

A atividade deve ser executada individualmente e entregue através do ambiente *Google Classroom*.

**Regras de criação dos programas:**

Crie um novo projeto Java denominado **AtividadeExcecoes1**. As classes devem possuir os nomes informados no texto. Ao final, o projeto deve ser exportado para um arquivo em formato ZIP.

**Nome completo: Francisco Aldenor Silva Neto**

1. O que é uma exceção? Explique.

Uma exceção é uma ocorrência que interrompe a execução normal do programa. Ela representa um evento que ocorre durante a execução do programa e que afeta o fluxo normal de execução. Exceções podem ser lançadas tanto pelo sistema quanto pelo próprio programa.

1. Explique detalhadamente a diferença entre as exceções verificadas e não verificadas.

Exceções verificadas são exceções que o compilador Java obriga o desenvolvedor a lidar ou declarar explicitamente que a exceção pode ser lançada. Essas exceções são herdadas da classe java.lang.Exception. Isso significa que, se você escrever um método que pode lançar uma exceção verificada, você precisa lidar com essa exceção ou declarar que a exceção pode ser lançada, usando a palavra-chave "throws". Um exemplo de exceção verificada é a exceção "IOException", que é lançada quando ocorre um erro de entrada/saída durante a execução de um programa.

Por outro lado, exceções não verificadas, também conhecidas como exceções de tempo de execução, não precisam ser tratadas ou declaradas explicitamente. Essas exceções são herdadas da classe java.lang.RuntimeException. Elas geralmente ocorrem quando há um erro de lógica no código ou uma condição anormal de execução, como uma divisão por zero ou um acesso a um índice inválido de uma matriz. Essas exceções podem ser evitadas por meio de uma programação cuidadosa e testes rigorosos.

1. Qual a hierarquia de classes do mecanismo de tratamento de exceções da linguagem Java?

• java.lang.Throwable: é a classe raiz da hierarquia de exceções em Java. Ela tem duas subclasses diretas:

• Exception: representa exceções que podem ser tratadas pelo programa. Ela tem diversas subclasses, como FileNotFoundException, IOException, NullPointerException, entre outras.

• RuntimeException: representa exceções que ocorrem em tempo de execução e normalmente são causadas por erros de programação ou problemas na lógica do programa. Ela tem diversas subclasses, como NullPointerException, ArithmeticException, IndexOutOfBoundsException, entre outras.

• Error: representa exceções que ocorrem em situações graves que normalmente não podem ser tratadas pelo programa, como falta de memória. Ela tem diversas subclasses, como OutOfMemoryError, StackOverflowError, entre outras.

As classes Exception e RuntimeException são as mais comuns e importantes para os desenvolvedores.

1. Explique conceitualmente as diretivas a seguir, bem como crie um código-fonte simplificado demonstrando o seu uso (pode ser um código-fonte único demonstrando todas as diretivas):
   1. try... catch

O bloco "try" é usado para envolver um código que pode lançar exceções e o bloco "catch" é usado para capturar e tratar as exceções que podem ocorrer durante a execução do código do bloco "try".

* 1. Throws

é usada para declarar que um método pode lançar exceções de um determinado tipo. Ao declarar as exceções que um método pode lançar, você informa aos usuários do método quais exceções eles devem lidar ou tratar em seu próprio código.

* 1. throw

é usada para lançar uma exceção manualmente dentro de um bloco de código. Isso permite que você crie exceções personalizadas ou lance exceções pré-definidas em situações específicas.

1. Crie a classe Cadastro que contém um método main e solicita do usuário o fornecimento das informações a seguir: nomeCompleto e telefone (composto por ddd e numero). Você pode utilizar as classes Scanner ou JOptionPane para realizar esta tarefa. Após receber as informações, verifique se nomeCompleto é diferente de vazio. Adicionalmente, faça a leitura do telefone em duas etapas. A primeira deve solicitar o ddd e a segunda o numero. O ddd deve possuir dois dígitos inteiros e o numero oito ou nove dígitos inteiros. Se o usuário fornecer corretamente estas informações, você deve emitir uma mensagem informando que o cadastro foi realizado e imprimir os valores recebidos. Caso alguma das verificações falhe, você deve notificar o usuário através do lançamento e tratamento de uma exceção. Em seguida, deve reiniciar o processo desde o início, repetindo esta operação até que todas as informações sejam preenchidas.

Boa sorte!

Prof. Igor.