

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:

Jorge Fernando Ramos Bezerra

1. O que é uma exceção? Explique.

R: Uma exceção em Java é um objeto da classe `java.lang.Exception`, ou de uma de suas subclasses. Como todo objeto, a exceção é capaz de armazenar dados nas suas variáveis de instância. Quando um erro ou situação anormal é encontrado durante a execução de um método, um objeto exceção é construído, e diversos dados sobre essa ocorrência são registrados nos campos desse objeto.

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

R: Exceções runtime (não verificadas):

- Podem capturar e tratar exceções..
- Menor trabalho para codificar.
- Geralmente representam bugs.
- O código deve ser consertado para que ela não ocorra mais.

Exceções não runtime (verificadas):

- Captura e tratamento de exceções são obrigatórios para essas exceções.
- Aumentam a confiabilidade do código.
- Geralmente representam situações inesperadas.
- Em seu tratamento, a situação deve ser contornada para que o programa continue a funcionar.

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

R: A classe raiz `Throwable`, da qual as classes `Error` e `Exception` herdam, sendo esta última herdada pela `RuntimeException` e não `Runtime`.

4. 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):

- a. `try... catch`

R: Quando uma exceção ocorre dentro de um bloco `try`, procura-se um tratador adequado, às cláusulas `catch` são cheçadas em sequência, para que a exceção seja tratada.

- b. `throws`

R: É possível propagar uma exceção para fora do método onde ela ocorreu. Uso da palavra reservada **`throws`** A responsabilidade de tratar fica com o código que chama o método. Por sua vez, pode tratá-la ou lançá-la novamente.

c. `throw`

R: Caso não seja encontrado um tratador no bloco onde a exceção ocorreu, o método `throw` propaga essa exceção para um bloco mais externo. A propagação continua até que algum tratador seja encontrado ou até chegar ao nível da JVM. O tratamento padrão da JVM é imprimir a exceção na saída padrão.

5. 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.