

Lista de atividades ECOP05 – Aula 8

Davi dos Santos Costa - 2021010666

1ª - Liste cinco exemplos de exceções comuns.

- divisões com denominador 0
- operações matemáticas onde uma das partes é uma “String” ou um formato que não seja o da variável resultado.
- exceções de tipo, para indicar que o tipo de uma variável não é o que deveria ser.
- exceções de construtores para indicar que os parâmetros de construtor não são válidos.
- exceções de faixa de entrada onde o valor deve estar dentro de uma faixa de valores.

2ª - Se nenhuma exceção é lançada em um bloco try, onde o controle prossegue quando o bloco try completa a execução?

O bloco try termina sua execução e o controle prossegue para o bloco finally, caso haja um, caso contrário continua para o fim da função, já que o bloco catch não será executado.

3ª - Dê uma vantagem de utilizar catch(Exception nomeDaExceção).

Deixa mais claro para o desenvolvedor a que aquela exception se refere de modo a facilitar a manutenção futura do código e ficar mais claro a que aquela exception está relatando ou a o que ela está relacionada.

4ª - O que acontece se nenhuma rotina de tratamento catch corresponder ao tipo de um objeto lançado?

Se uma exceção não tiver sido capturada quando o controle entrar em um bloco *finally* e esse bloco lançar uma exceção que não será capturada por ele, a primeira exceção será perdida e a exceção do bloco será retornada ao método chamador

5ª - O que acontece se vários blocos catch correspondem ao tipo do objeto lançado?

Se múltiplos blocos **catch** correspondem a um tipo particular de exceção, somente o **primeiro** bloco **catch** correspondente executará na ocorrência de uma exceção desse tipo. É um **erro de compilação** capturar *exatamente* o mesmo tipo em dois blocos **catch** diferentes associados com um bloco **try** particular

6ª - Por que um programador especificaria um tipo de superclasse como o tipo em um bloco catch?

Se uma rotina de tratamento **catch** for escrita para capturar objetos de exceção de *superclasse*, ele também pode capturar todos os objetos de *subclasses* dessa classe. Isso permite que **catch** trate exceções relacionadas polimorficamente. Você pode capturar cada *subclasse* individualmente se essas exceções exigirem processamento diferente.