Programação Orientada a Objetos - Exercícios

Tratamento de exceções

Profa. Fernanda Cunha

Obs.: Utilize as convenções na nomeação das classes, atributos e métodos.

1) Crie uma classe chamada Conta – contendo um campo do tipo double para saldo, um método construtor sem parâmetros que cria a conta com saldo zero, e um método chamado deposita(), que recebe um valor por parâmetro, correspondente a uma quantia a ser depositada na conta. O método deve lançar uma exceção do tipo argumento ilegal (IllegalArgumentException) quando um valor inválido (menor ou igual a zero) é passado. Para isso, use um teste if-else, e o comando throw para lançar a exceção (caso o valor esteja o.k., some o valor ao saldo da conta). Crie a classe principal da aplicação anterior, que deverá instanciar um objeto do tipo conta. Experimente depositar valores inválidos e veja o que ocorre. Trate a exceção lançada, colocando a chamada do método deposita() do item anterior dentro de um bloco try-catch.

2) Utilize herança para criar uma superclasse de exceção não verificada chamada ExceptionA e subclasses de exceção ExceptionB e ExceptionC, em que ExceptionB herda de ExceptionA e ExceptionC herda de ExceptionB. Escreva um programa para demonstrar que o bloco catch para o tipo ExceptionA captura exceções de tipos ExceptionB e ExceptionC. Ao final, o programa deve escrever na tela “Fim do Programa”, independente de exceções. Experimente colocar um catch para cada exceção, em ordens diversas para verificar o que ocorre (A, B, C, depois C, B, A).

3) Implemente classes que criam exceções não verificadas EmailNaoContemArrobaException e EmailNaoContemPontoComException para informar ao usuário que o valor fornecido por ele em uma String para o e-mail não possui o símbolo “@” ou que o e-mail não contém “.com”, dependendo de qual for o problema. Escreva um programa que leia uma String fornecida pelo usuário, e que trate estas exceções por meio de mensagem ao usuário, e depois mostre na tela o valor da variável, independentemente de haver ocorrido exceção ou não. É obrigatório o uso do tratamento de exceções (throw, try, catch, finally, etc.).

4) Implemente classes que criam exceções não verificadas NumerosIguaisException e SegundoNumeroIgualZeroException para informar ao usuário que os dois números não podem ser iguais, ou que o segundo número deve ser diferente de zero, dependendo de qual for o problema. Escreva um programa que leia dois números inteiros fornecidos pelo usuário, e que trate estas exceções por meio de mensagem ao usuário, e depois mostre na tela os valores, independentemente de haver ocorrido exceção ou não. É obrigatório o uso do tratamento de exceções (throw, try, catch, finally, etc.). Inclua uma exceção verificada NumerosNegativosException que informa o usuário que os dois números não podem ser negativos ao mesmo tempo.