INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA
Campus Eunápolis

Curso: Análise e Desenvolvimento de Software.

Disciplina: Programação Orientada a Objetos.

Exercícios de Auto-Revisão

1. Lista cinco exemplos de exceções comuns.

2. Por que as técnicas de tratamento de exceções não devem ser utilizadas para controle convencional de programação?

3.Se nenhuma exceção for disparada em um bloco try, para onde segue o controle quando o bloco try completa a execução?

4.O que acontece se ocorrer uma exceção e um tratador de exceções apropriado não puder ser localizado?

5.O que acontece se vários tratadores correspondem ao tipo de objeto disparado?

6.Qual é a razão fundamental para utilizar blocos finally?

7.O que acontece quando um tratador catch dispara uma Exception?

8.O que acontece com uma referência local em um bloco try quando esse bloco dispara uma Exception?

9.Escreva um programa em Java que mostra que a ordem dos tratadores de exceções é importante. Se você tentar capturar um tipo de exceção de superclasse antes de um tipo de subclasse, o compilador deve gerar erros. Explique por que ocorrem esse erros.

10. Construa uma classe chamada “ContaCorrente” , com os atributos “limite” que armazena a quantidade atual disponível do limite da conta que o usuário possui , o atributo “saldo” que é o valor que realmente é pertencente ao usuário, e o atributo “valorLimite” que consiste no valor máximo que o banco lhe fornece como valor de limite, todos float. Construa os métodos public void, sacar(float valor), depositar(float valor), e setValorLimite(float valor) .
Na construção dos três métodos faça com que eles lancem exceptions relacionados aos argumentos, por exemplo, sacar, depositar ou setar um valor negativo para esses eventos. Lance também uma exception no caso de tentar sacar um valor maior que o possível.
Nós métodos lancem a exception “IllegalArgumentException()” (Java)com o comando adequado, passe o motivo da exception(uma String) em seu construtor.