

**Paradigmas e Linguagens de Programação**

**Exercício sobre Exceptions**

**Professor:** Daniel Abella

**Curso:** Sistemas de Informação

# Requisitos

* Os códigos devem ser escritos na linguagem Java
* Os códigos fontes devem ser armazenados no GitHub
* Os testes devem ser realizados utilizando o JUnit 4 ou posterior

# Lista de Exercícios

**Exercício 1.** Construa uma classe chamada “ContaCorrente” com os atributos:

- “limite” que armazena a quantidade atual disponível do limite da conta que o usuário possui

- “saldo” que é o valor que realmente é pertencente ao usuário

- “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. Nos métodos lancem a exception “IllegalArgumentException()” (Java)com comando “throw”, passe o motivo da exception(uma String) em seu construtor.

**Exercício 2.** Crie um TestCase no JUnit, onde se deve instanciar um objeto da classe “ContaCorrente”, usar os três métodos construídos acima e tratar as exceções propostas com os comandos “try” e “catch”, no catch imprimir na tela o método “getMessage()” da exception declarada no catch. Faça condições apropriadas para a execução das exceções.

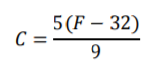
**Exercício 3.** As exceptions criadas no exercício 1 são exceções unchecked que o Java não obriga a tratá-las mude todas as exceptions do exercício de “IllegalArgumentException()” para “Exception()” que são exceções checked e observe o que ocorre na compilação.

**Exercício 4.** Considere o problema de conversão de temperaturas entre Celsius e Fahrenheit em Java.

Crie duas exceções, TemperatureException que herda de Exception e FahrenheitException que herda de TemperatureException.

Crie uma classe utilitária (Utils) com um método para converter temperaturas em graus Fahrenheit para graus Celsius (toCelsius()), ambas representadas em ponto-flutuante (double).

Porém, caso o valor a ser convertido seja menor que zero absoluto (-459,67°F) deve-se lançar a exceção FahrenheitException. A formula para conversão é dada por:



**Exercício 5.** Suponha que o método "sacar" da classe Conta vai ser rescrito de forma a lançar uma exceção criada por você, cuja classe é SaldoInsuficienteExcecao (extends Exception). A exceção é lançada sempre que o saldo da conta for inferior ao valor sacado. Implemente o método saca que lança esta exceção. E rescreva o código da caixa com o devido tratamento da exceção.

**Exercício 6.** Com base no exercício 5, suponha que quando lançada a exceção SaldoInsuficienteExcecao, através do objeto exceção instanciado, seja possível recuperar o saldo da pessoa. Como você implementaria isso? Mostre tudo que deve ser modificado/acrescentado no exercício para que isto funcione

**Referências:**

<http://homepages.dcc.ufmg.br/~rimsa/documents/decom009/exercises/Lista08.pdf>