Tecnologia em Análise e Desenvolvimento de Sistemas

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA

Disciplina: Programação Orientada a Objetos
Prof. Rodrigo R. Silva (rodrigorosa@ifsul.edu.br)
https://github.com/Prof-Rodrigo-Silva

Exercícios POO - Lista 13

1. Implemente uma classe executável que aceite a digitação de dois números e faça a divisão entre eles exibindo seu resultado. Sua classe deve tratar a exceções possíveis de ocorrer;

2. Implemente uma classe Usuário com os atributos:

   a) nome, cpf e telefone;

   Após implemente uma classe executável para realizar a entrada de dados, a classe deve tratar as exceções possíveis de ocorrer;

3. Modifique o projeto anterior(Item 2), e implemente uma classe personalizada de exceções. Use throw para lançar as exceções para a classe, além disso implemente também um bloco finally com alguma mensagem personalizada.

4. Implemente um classe Conta com o atributo saldo e implemente um método para sacar e depositar, um construtor sobrecarregado para inicializar o saldo com 1000. Implemente o uma classe executável que contenha mais dois métodos estáticos. A o método main deve chamar o metodo1 que por sua vez deve chamar o metodo2, o metodo2 possui a seguinte implementação:

```
public static void metodo2() {
    System.out.println("Inicio do metodo2");
    ContaTeste c = new ContaTeste(1000);
    for(int i = 0; i <= 15; i++) {
        c.depositar(i+1000);
        System.out.println(c.getSaldo());
        if (i == 5) {
            c = null;
        }
    }
}
```

   a) O quê acontece ao compilar a aplicação?

b) Adicione um try/catch em volta do for. O que o código imprime?

c) Em vez de fazer o try em torno do for inteiro, tente apenas com o bloco de dentro do for. Qual o resultado?

d) Retire o try/catch e coloque-o em volta da chamada do metodo2.

e) Faça a mesma coisa retirando o try/catch novamente e colocando-o em volta da chamada do metodo1. Rode os códigos, o que acontece?

5. Implemente um método sacar1 na classe Conta conforme o código abaixo:

```
public void sacar1(double valor) {
    if (this.saldo < valor) {
    } else {
    this.saldo-=valor;
    }
}
```

a) Use throw para lançar a exceção IllegalArgumentException do método sacar1 para a classe que fez o chamado, no caso a classe executável;
OBS: IllegalArgumentException é a melhor escolha quando um argumento sempre é inválido, por exemplo, números negativos, referências nulas, etc.

b) Instancie Conta com saldo zero, deposite um valor e realize um saque com valor maior que o depósito. Implemente o tratamento com o bloco try/catch e teste o código, qual a saída?

c) Retorne uma mensagem a partir do throw no método sacar1. Após use getMessage() no bloco catch;

d) O que acontece se tentar sacar um valor nulo? Qual o erro apresentado? Implemente a captura e tratamento do erro;