

Exercícios:

Data: 20/11/2017

1. Escreva um programa que lê uma sequência de primeiros nomes e os armazena em uma coleção Java, que não permita armazenar nomes duplicados. Implemente também um método que permita ao usuário procurar um primeiro nome na coleção.
2. Escreva um programa que utilize a classe `StringTokenizer` de Java para separar em tokens uma linha de texto inserida pelo usuário e que coloque cada token em um `TreeSet`. Imprima os elementos do `TreeSet`. (Obs.: isso deve fazer com que os elementos sejam impressos na ordem de classificação ascendente).
3. Implemente um programa que determine e imprima o número de palavras duplicadas em uma frase, usando coleções. Trate da mesma maneira letras minúsculas e maiúsculas. Ignore a pontuação.
4. Desenvolva um programa que permite realizar o jogo de Bingo de algumas prendas que uma instituição de amparo e saúde resolveu colocar no jogo afim de arrecadar fundos para esta instituição. Seu programa deve permitir a participação de N jogadores - valor lido do teclado -, tal que cada participante inscrito no Bingo só pode ter acesso a uma única cartela. Para o sorteio dos números do Bingo em cada rodada, implemente um método semelhante a classe `Loteria` dos slides da aula 9, o qual utiliza a coleção `TreeSet` de Java para armazenar os números sorteados. O Bingo implementado deve representar números de 1 a 50 e, no final, seu programa deve imprimir o jogador que venceu e vai levar a prenda sorteada pela instituição.
5. Na Aula 10 é mostrado um exemplo de uso da cláusula `finally()` para manipulação de exceções na linguagem Java. Para entender melhor seu funcionamento, implemente o exemplo da aula da classe `CarroBomba` e explique o que acontece quando é executado o código desse programa. Utilize ainda pelo menos 10 valores diferentes para a variável `temperatura` e construa uma tabela associando o valor ao tipo de exceção ocorrido (`SuperAquecimento` ou `Fogo`).
6. Crie um programa que leia dois inteiros do teclado, realize a divisão do primeiro pelo segundo e utilize classes de exceções pré-definidas em Java para tratar os seguintes erros:
 - a. Que os números informados sejam do tipo inteiro;
 - b. Que o divisor seja diferente de zero;
 - c. Que o resultado será um número inteiro sem perda de informação.

O tratamento de exceção a ser feito no bloco `catch()`, deve solicitar ao usuário que digite um novo número inteiro, caso este apresente um dos erros acima. O programa só se encerrará quando a digitação dos dois valores estiver correta.

7. O código mostrado abaixo é um programa em Java que faz a manipulação de exceções. Explique o que esse programa faz, implementando-o num ambiente de programação Java. Por que a classe *main()* tem em seu cabeçalho uma referência para a exceção *IOException*?

```
public static void main(String[] args) throws IOException {
    System.out.println("Bloco 1");
    try {
        System.out.println("Bloco 2");
        try {
            System.out.println("Bloco 3");
            try {
                switch(Math.abs(new Random().nextInt())%4+1){
                    default:
                    case 1: throw new EOFException();
                    case 2: throw new NullPointerException();
                    case 3: throw new NumberFormatException();
                    case 4: throw new IOException();
                }
            } catch (NumberFormatException e) {
                System.out.println("Trata no bloco 3");
            }
        } catch (NullPointerException e) {
            System.out.println("Trata no bloco 2");
        }
    } catch (EOFException e){
        System.out.println("Trata no bloco 1");
    }
}
```

8. Crie um programa que crie um construtor passando informação sobre a ocorrência de exceções para um manipulador de exceções. Crie a classe ***SomeException***, que lança uma exceção do tipo ***Exception*** no construtor. Seu programa deve tentar criar um objeto desta classe e capturar a exceção que é lançada no construtor.