



Universidade do Estado de Minas Gerais – UEMG
Engenharia de Computação - Ituiutaba
Disciplina: Programação Orientada a Objetos
Professor: Mateus de Paula Ferreira

Exercícios Generics e Collections

- 1 Crie uma versão genérica simplificada do método *isEqualTo*, que compara seus dois argumentos com o método *equals* e retorna *true* se forem iguais e *false* caso contrário. Teste o método genérico com diversos tipos, incluindo *Object*.
- 2 Crie uma classe genérica *Par* que possui dois parâmetros – *P* e *S* – cada um representando o tipo do primeiro e segundo elementos do par, respectivamente. Adicione *getters* e *setters* para os dois elementos.
- 3 Crie uma função genérica *localizar()*, que recebe como parâmetros um vetor, um elemento do mesmo tipo que os elementos do vetor e um inteiro informando o tamanho do vetor. A função deve tentar localizar o elemento representado pelo segundo parâmetro no vetor. Caso o elemento seja encontrado, a função deve retornar o índice do elemento, caso contrário, deve retornar -1.
- 4 Crie uma classe genérica *Vetor*, que recebe como parâmetros o número e o tipo de elementos do vetor. Nesta classe devem ser criados um construtor para alocar o vetor, um método para retornar a quantidade de elementos no vetor (chamado *getSize()*) e outro para adicionar um elemento (chamado *add()*). **Não utilize a classe *Vector*.**
- 5 Crie um programa que lê uma série de nomes e os armazena em uma *LinkedList*. Não armazene nomes repetidos. O programa ainda deve permitir que o usuário localize nomes armazenados.
- 6 *Utilizando Collections faça um programa* para contar as ocorrências de cada letra em uma string.
- 7 Crie um programa que determina e imprime o número de palavras duplicadas em uma sentença usando coleções. Trate maiúsculas e minúsculas sem diferenciação, e ignore a pontuação.
- 8 Ordene um vetor de 100 números inteiros gerados aleatoriamente utilizando *List*.