



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

AD2 de Programação III

2º semestre de 2011.

Nome:

Matrícula:

Pólo:

Obs: *A solução para o exercício proposto deve ser entregue por escrito e em formato digital.*

O programa abaixo apresenta uma classe genérica que foi discutida na aula 9 do nosso curso.

```
import java.util.ArrayList;
import java.util.List;

class MinhaLista<T> {
    private List<T> itens = new ArrayList<T>();
    void add (T item) { itens.add(item); }
    T primeiro() { return itens.get(0); }
}

public class AD2_2008_1 {

    public static void main(String[] args) {
        MinhaLista<String> ml = new MinhaLista<String>();
        MinhaLista<String> ml2 = new MinhaLista<String>();
        ml.add("primeiro");
        ml.add("segundo");
        ml2.add("segundo");
        ml2.add("primeiro");
        System.out.println(ml.primeiro());
        System.out.println("Último elemento: " + ml.ultimo());
        System.out.println("Elemento qualquer: " +
ml.aleatorio());
        System.out.println("Terceiro pertence a lista? " +
ml.pertence("terceiro"));
        System.out.println("As listas são iguais? " +
ml.igual(ml2));
    }
}
```

Complete a classe genérica. Ou seja, para que as chamadas no método main() funcione, crie métodos nesta classe genérica para (não precisa repetir o código da classe, basta listar como o método seria implementado):

- a) Retornar o último elemento da lista
- b) Retornar um elemento aleatório (Dica: *Math.random()* é um método estático que retorna um valor entre 0 e 1)
- c) Testar se um dado elemento pertence à lista
- d) Testar se um dado objeto da classe *MinhaLista* é igual ao objeto ao qual o método é aplicado (Obs.: Duas listas são iguais se contém elementos seguindo uma mesma ordem.)
- e) Testar se uma lista está contida em outra
- f) Testar se uma lista contém outra
- g) Retornar uma nova lista, instância de *MinhaLista*, que contenha a interseção de duas listas
- i) Retornar a concatenação de 2 instâncias da classe *MinhaLista*
- j) Retornar o tamanho (quantidade de elementos) de uma instância da classe *MinhaLista*

GABARITO:

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

/**
 * Classe que implementa os métodos solicitados na AD2 de Prog III / 2011-2
 * @author Carlos Bazilio
 *
 * @param <T> Parâmetro da classe genérica
 */
class MinhaLista<T> {
    private List<T> itens = new ArrayList<T>();
    public void add (T item) { itens.add(item); }
    public T primeiro() { return itens.get(0); }
    public T ultimo() {
        return itens.get(itens.size()-1);
    }
    public T aleatorio() {
        return itens.get((int)(Math.random()*(itens.size()-1)));
    }
    public boolean pertence(T object) {
        return itens.contains(object);
    }
    public boolean igual(MinhaLista<T> ml2) {
        Iterator<T> i = itens.iterator();
        Iterator<T> i2 = ml2.itens.iterator();
        while (i.hasNext() && i2.hasNext()) {
            if (i.next() != i2.next())
                return false;
        }
        if (!i.hasNext() && !i2.hasNext())
```

```

        return true;
    }
    return false;
}
public boolean contida(MinhaLista<T> ml2) {
    Iterator<T> i = itens.iterator();
    List<T> l2 = ml2.itens;
    while (i.hasNext()) {
        if (!l2.contains(i.next()))
            return false;
    }
    return true;
}
public boolean contem(MinhaLista<T> ml2) {
    return ml2.contida(this);
}
public MinhaLista<T> intersecao(MinhaLista<T> ml2) {
    MinhaLista<T> mlNova = new MinhaLista<T>();
    Iterator<T> i = itens.iterator();
    while (i.hasNext()) {
        T obj = i.next();
        if (ml2.pertence(obj))
            mlNova.add(obj);
    }
    return mlNova;
}
public void exhibe() {
    Iterator<T> i = itens.iterator();
    while (i.hasNext()) {
        System.out.print(i.next().toString());
    }
    System.out.println();
}
public MinhaLista<T> concatena(MinhaLista<T> ml2) {
    MinhaLista<T> mlNova = new MinhaLista<T>();
    for (T t : itens)
        mlNova.add(t);
    for (T t : ml2.itens)
        mlNova.add(t);
    return mlNova;
}
public int tamanho(){
    return itens.size();
}
}

```