

1ª questão: Considere o código da classe *Book*, implementado em:

http://www.eng.uerj.br/~fariasol/disciplinas/Caracteristicas_das_LPs/clp-2013-02-p1/Book/src1-4.java

- i) inclua, nesse código, uma variável de instância que registre o número de páginas do livro em questão;
- ii) implemente na classe *Book* a interface *Relatable*, descrita em:
<http://docs.oracle.com/javase/tutorial/java/IandI/usinginterface.html>
- iii) Após ter implementado a interface *Relatable*, desenvolva um pequeno programa em Java que instancie adequadamente dois objetos do tipo *Book* e diga qual dos dois é o maior (segundo o parâmetro número de páginas).

2ª questão: Considere as classes *LinkedList* e *Node*, cujos códigos se encontram em arquivos .java em:

http://www.eng.uerj.br/~fariasol/disciplinas/Caracteristicas_das_LPs/clp-2013-02-p1/LinkedList/ e que implementam uma lista encadeada que pode conter quaisquer tipos de objetos.

- i) Altere a classe *LinkedList*, incluindo uma variável de instância - *capacity* - que delimite o número máximo de de nodos que a *LinkedList* poderá armazenar.
- ii) Faça com que a classe *LinkedList* lance exceções nos seguintes casos: a) tentativa de retirar um elemento de uma *LinkedList* que se encontre vazia; b) tentativa de inserir um elemento em uma *LinkedList* que já se encontra com a sua capacidade de armazenar elementos esgotada. Observação: Para isto você deverá criar tipos de exceções apropriados.
- iii) Inclua um método *void printLinkedList()*, para imprimir os elementos de sua lista.
- iv) Desenvolva uma pequena aplicação em java, através da classe *LinkedListExample*, que crie uma *LinkedList*, insira elementos na lista e retire elementos da lista, testando a ocorrência das exceções descritas em “ii”, as quais deverão ser tratadas em *LinkedListExample*.
- v) Imprima a sua *LinkedList* no código de *LinkedListExample*, antes e após a mesma ser alterada, afim de testar o seu funcionamento.

3ª questão: Considere as classes *LinkedList* e *Node* da questão anterior. i) Introduza nessas classes as modificações necessárias para criar os tipos genéricos (*generic types*) *Node<T>* e *LinkedList <T>*. ii) Como você faria para operar com lista encadeadas que apresentem a seguinte restrição: apenas podem armazenar objetos que sejam instâncias da classe *Book* ou de alguma subclasse de *Book*. iii) Desenvolva um pequeno programa, usando *generic types*, que crie uma *LinkedList* de alguns livros que você deseja ler (instâncias de *Book*) e após, imprima a lista recém-criada. Nessa solução, a inserção na *LinkedList* de qualquer objeto que não seja instância de *Book* ou de uma de suas subclasses deve ocasionar um erro de compilação..

4ª questão: Considere o código que se segue da classe *Aluno*. Como você faria em Java para criar um clone de um objeto que seja instância da classe *Aluno*? Forneça o código correspondente.

Obs. A classe *Pessoa* possui, como variáveis de instância apenas tipos de dados primitivos e membros imutáveis (p.ex., da classe *String*).

Obs.:

```
public class Aluno {
// declaração das variáveis de instância;
    private String matricula;
    private String nome;
    private Pessoa namorada;

// construtor da classe aluno
    public Aluno(String matr, String nome) {
        matricula = matr;
        this.nome = nome;
        this.namorada = namorada;
    }
    public setNamorada (Pessoa namorada) {
        this.namorada = namorada;
    }
// implementação de outros métodos da classe aluno
    ...
}
```

Boa Sorte! Prof. Oscar Luiz Monteiro de Farias