



Disciplina: Programação para Web I	Semestre: 3º
Professora: Silvia Bertagnolli	

LISTA DE EXERCÍCIOS

1) Faça um programa que cria uma coleção do tipo Double e:

- 1.1 Ordene os números da coleção criada no item 1 usando a classe Collections
 - 1.2 Modifique a ordem dos elementos da coleção criada no item 1 usando um método da classe Collections
 - 1.3 Leia um número e verifique se ele está na coleção criada no item 1
- Obs.: use para leitura a classe JOptionPane

2) Analise a classe Par abaixo e depois responda os itens 2.1 a 2.3 abaixo:

```
public class Par <P, S>{  
    private P primeiro;  
    private S segundo;  
    public Par(){}  
    public Par(P p, S s){  
        primeiro = p; segundo = s;  
    }  
    public P getPrimeiro(){return primeiro;}  
    public S getSegundo(){return segundo;}  
    public void setPrimeiro(P p){primeiro= p;}  
    public void setSegundo(S s){segundo = s;}  
    public String toString(){ return primeiro + " " + segundo; }  
}
```

2.1 Qual a saída produzida pela classe de Teste abaixo?

```
public class Teste {  
    public static void main(String[] args) {  
        Par<Integer, String> mes = new Par();  
        mes.setPrimeiro(1);  
        mes.setSegundo("Janeiro");  
        System.out.println(mes.toString());  
        mes.setPrimeiro(2);  
        mes.setSegundo("Fevereiro");  
        System.out.println(mes.toString());    }}
```

2.2 Use a classe Par definida anteriormente para armazenar o nome e a nota de um aluno (<String, Double>)

2.3 Use a classe Par definida anteriormente para armazenar um par de coordenadas x e y (<Float, Float>)



3) Crie a classe Produto abaixo:

Produto
- codigo : int
- descricao : String
- valor : double
+ toString() : String

4) Crie uma classe genérica Código para representar os códigos dos Produtos
códigos são compostos por duas partes que indicam o setor onde o produto é fabricado (setor pode ser String ou número); e números que são um código sequencial dentro do setor
Exemplos válidos: IMM120; IMM121 – 111120; 111121)

5) Modifique a classe Produto para que o seu código seja criado usando a classe Código acima.
Agora, monte a classe de Testes que deve criar objetos do tipo Produto com a modificação solicitada acima

6) Crie a interface genérica Lista como descrito abaixo:

6.1 A lista irá conter elementos

6.2 Defina os métodos abaixo usando genéricos onde for possível:

public void adicionar(Object obj)

public boolean remover(int i)

public String listar()

public int totalizar()

public void removerTodos()

public Object pegarElemento(int)

public void removerElemento(Object obj)

7) Agora, crie a classe MinhaLista que implementa a interface Lista genérica definida previamente

7.1 Essa classe deve declarar um atributo da classe LinkedList para armazenar objetos na classe MinhaLista e implemente todos os métodos da interface Lista

7.2 Monte uma classe de testes para armazenar objetos da classe Produto (que tem composição com a classe Código) em uma lista do tipo MinhaLista



8) Crie a interface genérica Mapa como descrito abaixo:

8.1 O mapa irá conter elementos compostos por chave e valor

8.2 Defina os métodos abaixo usando o conceito de genéricos onde for possível:

public Object get(Object obj) retorna o valor de uma chave específica

public boolean isEmpty() retorna true se o mapa está vazio

public Set keySet() retorna um conjunto com todas as chaves

public int size() retorna o número de objetos no mapa

public void put(Object chave, Object valor) – adiciona um objeto no mapa usando a chave e o valor

public void remove (Object obj) – remove um objeto usando a sua chave

public List values() – retorna uma lista com os valores armazenados no mapa

9) Agora, crie a classe MeuMapa que implementa a interface Mapa genérica definida previamente.

9.1 Essa classe deve declarar um atributo da classe LinkedHashMap para armazenar objetos na classe MeuMapa e implemente todos os métodos da interface Mapa

9.2 Monte uma classe de testes para armazenar objetos da classe String como chave e Double como valor