Universidade Federal de Uberlândia

Faculdade de Computação
GGI030 – Programação Orientada a Objetos
2o. Semestre de 2018
Prof. Renato Pimentel

Laboratório 06 – Interfaces

Atividade individual.

1. Crie uma interface e uma classe como as apresentadas a seguir:

```
1  public interface Classificavel {
2
3     boolean eMenorQue(Classificavel obj);
4  }
```

```
1  public class Classificador {
2
3     public void ordena(Classificavel[] a) {
4        Classificavel elem, menor;
5        int pos;
6        for (int i = 0; i < a.length - 1; i++) {
7           elem = a[i];
8           menor = a[i + 1];
9           pos = i + 1;
10          for (int j = i + 2; j < a.length; j++) {
11             if (a[j].eMenorQue(menor)) { // encontrando o menor
                   elemento
12                menor = a[j];
13                pos = j;
14             }
15          }
16          if (menor.eMenorQue(elem)) { // troca
17             a[i] = a[pos];
18             a[pos] = elem;
19          }
20       }
21    }
22 }
```

Considere agora uma empresa com as classes `Produto`, `Cliente` e `Servico`, que implementam a interface `Classificavel`. Defina os atributos e os

métodos convenientes para essas classes. Inclua ao menos os seguintes atributos: código, para produto; nome, para cliente; e preço, para serviço. Esses serão os atributos passíveis de serem ordenados em cada uma dessas classes. Como exemplo, veja:

```java
public class Produto implements Classificavel {

   private int codigo;

   public boolean eMenorQue(Classificavel o) {
      Produto compara = (Produto) o;
      if (this.codigo < compara.codigo) {
         return true;
      } else {
         return false;
      }
   }
}
```

Crie uma classe contendo o método `main()`. Defina nessa classe vetores de clientes, produtos e serviços. Adicione a eles ao menos três elementos e, em seguida, ordene esses vetores segundo seus atributos classificáveis e depois imprima o conteúdo deles.

Outras opções para ordenação

```java
public static int[] selectionSort(int[] vetor){
  for (int i = 0; i < vetor.length-1; i++) {
    int menor = vetor[i];
    int menorI = i;

    for (int j = i+1; j < vetor.length; j++) {
      if (vetor[j] < menor) {
        menorI = j;
        menor = vetor[j];
      }
    }

    int aux = vetor[i];
    vetor[i] = menor;
    vetor[menorI] = aux;
  }
  return(vetor);
}
```

Algoritmo: ordenação por inserção

```java
public static int[] insertionSort(int[] vetor) {

  int n = vetor.length;

  for (int j = 1; j < n; j++) {
    int chave = vetor[j];
    int i = j - 1;

     // procura lugar de insercao e desloca numeros
    while (i >= 0 && vetor[i] > chave) {
        vetor[i+1] = vetor[i];
        i = i - 1;
    }
    vetor[i+1] = chave;
  }

  return(vetor);
}
```