

~~Aula Prática 5~~

Objectivos

Utilização de Classes Abstractas e Interfaces.

~~Problema 5.1~~

Tome como base para este trabalho as classes desenvolvidas nas aulas anteriores relativamente às figuras geométricas (classes Ponto, Figura, Circulo, Quadrado e Rectângulo).

- a) Aplicando os conceitos de Classes Abstractas e de Interfaces construa o código necessário para executar o seguinte programa:

```
public class Test {

    public static void main(String[] args) {
        Figura listaFig[] = new Figura[] {
            new Circulo (2),
            new Circulo (1,3, 1),
            new Quadrado(5),
            new Quadrado(3,4, 2),
            new Rectangulo(2,3),
            new Rectangulo(3,4, 5,3),
            new Rectangulo(1,1, 5,6)
        };
        for (Figura f : listaFig) {
            System.out.println(f);
        }
        System.out.println("\nComparação da área do primeiro elemento com todos");
        for (int i = 0; i < listaFig.length; i++) {
            System.out.printf("%2d %12s %8.2f compareTo = %2d\n", i,
                listaFig[i].getClass().getSimpleName(),
                listaFig[i].area(),
                listaFig[i].compareTo(listaFig[0]));
        }

        System.out.println("\nFigura com maior Area: "
            + UtilCompare.findMax( listaFig ) );

        // Ordena (crescente) o array de Figuras por areas
        UtilCompare.sortArray(listaFig);

        System.out.println (" \nLista de Figuras Ordenadas por Area:");
        for (int i = 0; i < listaFig.length; i++) {
            if (listaFig[i] != null){
                System.out.print (listaFig[i]);
                System.out.println (" -> area: " +
                    String.format("%2.2f", listaFig[i].area())+" e perimetro:"+
                    String.format("%2.2f", listaFig[i].perimetro()));
            }
        }
    }
}
```

Notas:

- Tenha em atenção que o método “compareTo” deve estar definido numa Interface Comparable, a ser criada para o efeito.
- Os utilitários “findMax” e “sortArray” devem ser colocados numa classe “UtilCompare.java”
- Garanta que estes métodos podem ser aplicados a qualquer conjunto de “Comparable”s.

Verifique se obteve o seguinte resultado:

```
Circulo de Centro (0.0,0.0) e de raio 2.0
Circulo de Centro (1.0,3.0) e de raio 1.0
Quadrado de Centro (0.0,0.0) e de lado 5.0
Quadrado de Centro (3.0,4.0) e de lado 2.0
Rectangulo de Centro (0.0,0.0), altura 3.0, comprimento 2.0
Rectangulo de Centro (3.0,4.0), altura 3.0, comprimento 5.0
Rectangulo de Centro (1.0,1.0), altura 6.0, comprimento 5.0

Comparação da área do primeiro elemento com todos
0      Circulo      12,57 compareTo = 0
1      Circulo      3,14 compareTo = -1
2      Quadrado     25,00 compareTo = 1
3      Quadrado     4,00 compareTo = -1
4      Rectangulo   6,00 compareTo = -1
5      Rectangulo   15,00 compareTo = 1
6      Rectangulo   30,00 compareTo = 1

Figura com maior Area: Rectangulo de Centro (1.0,1.0), altura 6.0, comprimento 5.0

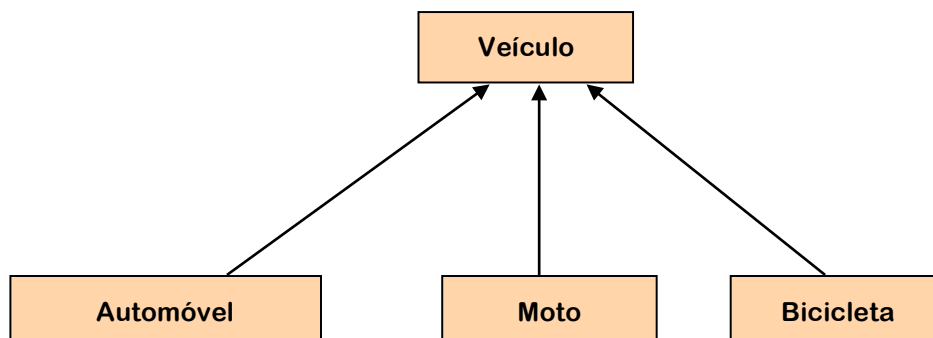
Lista de Figuras Ordenadas por Area:
Circulo de Centro (1.0,3.0) e de raio 1.0 -> area: 3,14 e perimetro: 6,28
Quadrado de Centro (3.0,4.0) e de lado 2.0 -> area: 4,00 e perimetro: 8,00
Rectangulo de Centro (0.0,0.0), altura 3.0, comprimento 2.0 -> area: 6,00 e perimetro: 10,00
Circulo de Centro (0.0,0.0) e de raio 2.0 -> area: 12,57 e perimetro: 12,57
Rectangulo de Centro (3.0,4.0), altura 3.0, comprimento 5.0 -> area: 15,00 e perimetro: 16,00
Quadrado de Centro (0.0,0.0) e de lado 5.0 -> area: 25,00 e perimetro: 20,00
Rectangulo de Centro (1.0,1.0), altura 6.0, comprimento 5.0 -> area: 30,00 e perimetro: 22,00
```

~~Problema 5.2~~

Utilizando Interfaces, reimplente o exercício 4.4. desenvolvido na aula anterior.

Problema 5.3

Considere o seguinte cenário de herança:



- a) ~~Implemente a hierarquia de classe acima descrita de forma a utilizar o conceito de generalização abordado nas aulas TP. Tenha em atenção as características e comportamento dos objectos apresentados como, por exemplo, o ano, a matrícula, a cor base, o n.º de rodas, a cilindrada, a velocidade máxima, etc.~~
- b) ~~Para além da hierarquia acima descrita, pretendemos ter alguns veículos (por exemplo BicicletaPolícia, MotoPolícia, CarroPolícia) que possam ter os seguintes comportamentos:~~
- a. ~~Polícia~~
 - i. ~~getTipo()~~
valores possíveis de retorno: INEM, Bombeiros, GNR, PSP, PJ
 - ii. ~~String getID() // identificador interno da viatura~~
 - b. ~~Motorizado~~
 - i. ~~getPotencia();~~
 - ii. ~~getConsumo();~~
 - iii. ~~getCombustivel();~~
- ~~Note que para obter uma solução poderá ter de modificar a hierarquia de classes ou acrescentar novas classes e/ou interfaces.~~
- g) ~~Desenvolva um programa *main* para demonstrar todas as funcionalidades implementadas em a) e b). Crie também uma lista de Veículos e ordene-os usando a função *UtilCompare.sortArray* desenvolvida no problema anterior (considere o ano como atributo de comparação).~~

Problema 5.4

Utilizando o programa desenvolvido nas aulas anteriores relativamente à Agenda de Contactos (Pessoas), acrescente-lhe a capacidade de guardar e carregar uma agenda de um ficheiro de texto.

Por uma questão de compatibilidade com outras soluções, a aplicação deverá suportar 3 tipos diferentes de formatos (Nokia, vCard e CSV). A primeira linha do ficheiro identifica o respectivo formato. De acordo com o formato, os registos estão armazenados de forma diferenciada:

- Nokia: Cada linha contém um dos elementos de informação (i.e. um campo). Existe uma linha em branco a separar cada registo;
- vCard: Cada linha contém um registo e os campos estão delimitados pelo caracter #;
- CSV: Cada linha contém um registo e os campos estão delimitados pelo caracter TAB (\t);

Sugestão: Defina uma interface normalizada com as operações (abstractas) necessárias à leitura e escrita de contactos. Implemente classes específicas que permitam operar com diferente formatos de ficheiros de contactos.

Exemplos de ficheiros de contactos com diferentes formatos:

File1.txt

Nokia
 Maria João Sousa
 9123454
 12/05/1989

Rui Manuel Costa
 985423
 20/09/19

Joaquim Ferreira
 9312452
 18/04/1964

...

File2.txt

vCard
 #Maria João Sousa#9123454#12/05/1989
 #Rui Manuel Costa#985423#20/09/19
 #Joaquim Ferreira#9312452#18/04/1964

...

File3.txt

CSV		
Maria João Sousa	9123454	12/05/1989
Rui Manuel Costa	9854233	20/09/19
Joaquim Ferreira	9312452	18/04/1964

...