

# Ficha Prática 3

António Nestor Ribeiro, António Fernandes, Mário Martins  
{anr,af,fmm}@di.uminho.pt  
POO (LMCC)

2004/05

## Objectivos

1. Classes e instâncias. Métodos e variáveis de instância
2. Construtores. Uso de `this`.
3. Exercícios

## Exercícios

1. (a) Escreva a classe `Circulo` com variáveis de instância:

```
double x,y; // coordenadas do centro do Círculo
double raio;
```

(não se esqueça de declarar as variáveis como `private`) e métodos de instância :

```
double perimetro()
double area()
boolean maior(Circulo c)
double getRaio()
void aumentaRaio(double aumento_raio)
```

(não se esqueça de declarar como `public` os métodos que devem ser visíveis do exterior do objecto). Adicione ainda dois construtores:

```
Circulo()
Circulo(double x, double y, double raio)
```

(mais uma vez não se esqueça de os declarar como `public`).

- (b) Escreva agora a seguinte classe de teste (na mesma directoria/pasta da anterior):

```
public class TesteCirculo {
    public static void main (String[] args) {
        Circulo c1 = new Circulo(1.0,1.0,5.0);
        Circulo c2 = new Circulo(5.0,2.0,3.0);

        System.out.println("Área de c1: "+c1.area());
        System.out.println("Perímetro de c2: "+c2.perimetro());
        c2.aumentaRaio(4.0);
        System.out.println("Perímetro de c2: "+c2.perimetro());
        System.out.println("c1 maior que c2? "+c1.maior(c2));
    }
}
```

Compile a classe de teste:

```
> javac TesteCirculo.java
```

e execute o programa:

```
> java TesteCirculo
```

(não use `java TesteCirculo.class!`). Os resultados deverão ser:

```
Área de c1: 78.54
Perímetro de c2: 18.8496
Perímetro de c2: 25.1328
c1 maior que c2? true
```

2. (a) Numa nova directoria/pasta escreva a classe `Complexo`, que deverá representar números complexos ( $a + bi$ ). Um número complexo deverá possuir métodos de instância:
- i. **somar** — para somar a si próprio um outro número complexo  
 $(a + bi) + (c + di) = (a + c) + (b + d)i$
  - ii. **subtrair** — para subtrair a si próprio um outro número complexo  
 $(a + bi) - (c + di) = (a - c) + (b - d)i$
  - iii. **multiplicar** — para se multiplicar por outro número complexo  
 $(a + bi) \times (c + di) = (a \times c - b \times d) + (a \times d + b \times c)i$

iv. `toString` — para obter a representação do número na forma de `String`

(b) Teste a classe escrevendo uma classe de teste:

- i. Crie os números complexos `complexo_1` ( $2 + 3i$ ), `complexo_2` ( $1 + 2i$ ), e `complexo_3` ( $3 + 1i$ );
- ii. Some a `complexo_1` o `complexo_2`;
- iii. Subtraia a `complexo_3` o `complexo_2`;
- iv. Multiplique `complexo_2` por `complexo_2`.

Os valores dos números no final deverão ser:

```
complexo_1 = 3 + 5i  
complexo_2 = -3 + 4i  
complexo_3 = 2 - 1i
```