

# Ficha Prática 3

António Nestor Ribeiro, José Creissac Campos  
{anr,jose.campos}@di.uminho.pt  
PIIV (LESI/LMCC)

2002/03

## Objectivos

1. Classes e instâncias.Métodos e variáveis de instância
2. Construtores.Usos de `this`.
3. Classes de Teste e BlueJ
4. 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(1,1,5);  
        Circulo c2 = new Circulo(5,2,3);  
  
        System.out.println("Área de c1: "+c1.area());  
        System.out.println("Perímetro de c2: "+c2.perimetro());  
        c2.aumentaRaio(4);  
        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
```

- (c) Uma alternativa a escrever uma classe de teste é utilizar o BLUEJ:
- i. Depois de iniciar o BLUEJ escolha a opção **Project/Open Non BlueJ** .... Selecciona a directoria em que escreveu a classe **Circulo** e carregue em **Open in BlueJ**. Aparecer-lhe-á a classe **Circulo** no BLUEJ (se anteriormente definiu a classe **TesteCirculo** esta também aparecerá, mas neste caso não vai ser necessária).
  - ii. Crie duas instâncias de **Circulo**: **c1** com centro em (1,1) e raio 5, e **c2** com centro em (5,2) e raio 3.

Para criar instâncias de `Circulo`, *clique* o botão direito do rato na classe `Circulo` e seleccione a opção `new Circulo(double, double, double)`, mude o nome da instância a criar para o nome que pretende e preencha os argumentos do construtor. O objecto aparecerá na barra ao fundo da janela do BLUEJ. Repita para criar a segunda instância de `Circulo`.

- iii. calcule a área de `c1` (isto é envie a mensagem `area()` a `c1`). Para enviar uma mensagem a `c1` *clique* o botão direito do rato no objecto `c1` e seleccione a mensagem pretendida (neste caso `double area()`).
- iv. Calcule o perímetro de `c2`.
- v. Pergunte a `c1` se ele é maior que `c2`. Para fazer isto seleccione a opção `double maior(Circulo)` em `c1`. Aparecerá uma nova janela em que é preciso indicar o parâmetro a utilizar: escreva `c2` (ou, alternativamente, *clique* em `c2`).

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 (usando o BLUEJ ou 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
```

3. Escreva uma classe `PortaMoedasMB` cujas instâncias deverão ter um comportamento semelhante ao do conhecido cartão. Cada instância deverá possuir, ao ser criada, um titular e um saldo. Cada instância deverá ser capaz de responder às seguintes mensagens:

- (a) `int saldo()` – apresentar o saldo do cartão;
- (b) `void pagar(int qt)` – efectuar um pagamento, caso seja possível (enquanto não falarmos de excepções, a classe deverá possuir um método `boolean levantamentoValido(int qt)` que indique se um dado levantamento é possível);
- (c) `void carregar(int qt)` – carregar o cartão;
- (d) `double mediaPagamentos()` – indicar a média de pagamentos do cartão;
- (e) `String toString()` – apresentar, como `String` a informação do cartão.

Para testar a sua classe escreva uma classe de teste, ou utilize o `BLUEJ`.