

Ficha Prática 3

António Nestor Ribeiro, António Fernandes, F. Mário Martins
{anr,af,fgm}@di.uminho.pt
PPIV (LESI)

2003/04

Objectivos

1. Classes e instâncias. Métodos e variáveis de instância
2. Utilização do BlueJ para prototipagem de sistemas de objectos.
3. Exercícios.

Exercícios

1. Considere agora que pretendemos reformular a classe `Circulo` feita na ficha anterior, substituindo os valores em `x` e `y` por uma instância da classe `Ponto2D`.
 - (a) Crie a classe `Ponto2D` utilizando código já desenvolvido nas aulas práticas;
 - (b) Implemente o método de `Ponto2D`: `boolean igual(Ponto2D ponto);`
 - (c) Modifique a classe `Circulo` para passar a ter como variável um `Ponto2D`, e implemente também o método `boolean igual(Circulo circ)`.
2. Uma alternativa a escrever uma classe de teste é utilizar o BLUEJ:
 - (a) 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).

- (b) 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(...)`, 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`.

- (c) 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()`).

- (d) Calcule o perímetro de `c2`.

- (e) 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`).

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 toString1()` – apresentar, como `String` a informação do cartão.

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

4. Considere agora que pretende reproduzir a estrutura de uma carteira com espaço para cartões (considere que é uma carteira com slots infinitos...para guardar muitos cartões!!!).

A carteira deve ter como informação os dados do dono (nome, morada, número de telefone), bem assim como os cartões que ele possui. Sobre uma carteira deve ser possível efectuar, entre outras, as seguintes operações:

- (a) Guardar mais um cartão;
- (b) Determinar o montante da carteira, isto é, o somatório do valor de todos os cartões;
- (c) Determinar quantos cartões estão na carteira;
- (d) Eliminar o cartão da posição X da carteira. Atenção que tal só é possível se o saldo devedor do cartão for positivo;
- (e) Listar todos os cartões existentes.