

Programação Orientada a Objetos I

Exercícios: polimorfismo

1 Exercícios

1. Crie a seguinte hierarquia de classes:

- Uma interface para representar qualquer forma geométrica, definindo métodos para cálculo do perímetro e cálculo da área da forma.
- Uma classe abstrata para representar quadriláteros. Seu construtor deve receber os tamanhos dos 4 lados e o método de cálculo do perímetro já pode ser implementado.
- Classes para representar retângulos e quadrados. A primeira deve receber o tamanho da base e da altura no construtor, enquanto a segunda deve receber apenas o tamanho do lado.
- Uma classe para representar um círculo. Seu construtor deve receber o tamanho do raio.

No programa principal, crie um vetor de formas e armazene quadrados, retângulos e círculos neste vetor. A seguir, imprima as seguintes informações:

- (a) os dados (lados ou raio) de cada forma geométrica no vetor.
- (b) os perímetros das formas.
- (c) as áreas das formas.

Para implementar (b) e (c), tire vantagem do polimorfismo, enquanto que para fazer o item (a) utilize `instanceof` e `downcast`.

2. Implemente uma lista encadeada, composta por uma sequência de células que possuem um elemento (conteúdo) e uma referência para a próxima célula. Implemente a classe que representa a célula e a classe que representa o conteúdo como classes internas (veja abaixo) da classe lista. A classe elemento deve conter um par de números inteiros, de forma que a lista encadeada seja uma lista de pares.

2 Classes internas

Em Java, quando queremos criar estruturas contidas dentro de uma outra classe é possível usar *classes internas*. Veja um exemplo:

```
public class A {  
  
    class B {}  
  
    class C {}  
  
}
```

As classes B e C são internas à classe A. Note que, apesar do exemplo não mostrar, classes internas podem ter atributos e métodos como uma classe normal. Para referenciar a classe B fora do arquivo A.java, usa-se A.B.