

Aula prática nº 7

Tópicos

- Polimorfismo
- Classes abstratas

Exercícios

1. Continue a trabalhar nos programas desenvolvido nas aulas anteriores relativamente às figuras geométricas (classes Ponto, Figura, Circulo, Quadrado e Retângulo), crie uma nova classe ColecaoFiguras que suporta um conjunto (coleção) de Figuras não repetidas. Implemente os seguintes métodos:

```
public ColecaoFiguras(double maxArea) // O construtor define a área
máxima da coleção de figuras
public boolean addFigura(Figura f)    // Adiciona uma figura à coleção
public boolean delFigura(Figura f)    // Remove uma figura da coleção
public double areaTotal()             // Retorna a área total das figs.
public boolean exists(Figura f)       // Verifica se uma figura existe
public String toString()              // Retorna as características da coleção
public Figura[] getFiguras()          // Retorna uma lista com todas as figs.
public Ponto[] getCentros()           // Retorna uma lista com todos os centros
```

Teste as classes desenvolvidas com o programa A07E01.java que se encontra no elearning. Verifique se obteve o resultado apresentado na página seguinte.

```

true
true
false
true
true
true
false
true
true

```

Area Total: 41.132741228718345

Lista de Figuras:

Circulo de Centro: (1.00 , 3.00) e de Raio: 2.00, Cor: Branco, Area: 12.57,
Perimetro: 0.00

Retangulo de Centro: (3.00 , 4.00), Dimensoes: 2.00x3.00, Cor: Branco, Area:
6.00, Perimetro: 0.00

Circulo de Centro: (0.00 , 0.00) e de Raio: 2.00, Cor: Branco, Area: 12.57,
Perimetro: 0.00

Quadrado de Centro: (3.00 , 4.00), Dimensoes: 2.00x2.00, Cor: Branco, Area:
4.00, Perimetro: 0.00

Circulos na Lista de Figuras:

Circulo de Centro: (1.00 , 3.00) e de Raio: 2.00, Cor: Branco, Area: 12.57,
Perimetro: 0.00

Circulo de Centro: (0.00 , 0.00) e de Raio: 2.00, Cor: Branco, Area: 12.57,
Perimetro: 0.00

Centro das Figuras:

```

(1.00 , 3.00)
(3.00 , 4.00)
(0.00 , 0.00)
(3.00 , 4.00)

```

2. Continue a trabalhar no ex. 1 da aula 5. Às classes existentes Pessoa e Aluno adicione tipos seguintes:

- classe Professor derivada da classe Pessoa. A classe Professor deve ter um atributo adicional que é área principal de investigação. Redefina o método `toString()` para que este devolva para além do nome e apelido, a área de investigação do professor. Não reescreva o método `toString()` completamente, aproveite o método existente na classe base.

- classe AlunoPosGrad derivada da classe Aluno. A classe AlunoPosGrad deve ter um atributo adicional – referência para o orientador. Redefina o método `toString()` para que este devolva para além do nome, apelido e número mecanográfico do aluno, o nome, apelido e a área de investigação do orientador.

- classe Disciplina que contém informação sobre o nome da disciplina, o professor responsável e alunos inscritos. Imagine que numa disciplina podem inscrever-se os alunos de mestrado integrado e os de pós-graduação e que o número de alunos é ilimitado. Inicialmente não se sabe quantos alunos vão inscrever. Assegure que o vetor de referências para alunos cresça dinamicamente.

Teste as classes desenvolvidas com uma função `main` que permita criar alunos e disciplinas, inscrever alunos em disciplinas e imprimir toda a informação sobre disciplinas, incluindo o seu nome, informação sobre o professor responsável, número de alunos inscritos e os dados de todos os alunos.

3. Construa três interfaces (`IDemo1`, `IDemo2`, `IDemo3`), cada uma com dois métodos. Crie uma nova interface (`IDemo123`) que estenda as 3 anteriores adicionando um novo método. Construa uma classe que implemente `IDemo123` e que herde uma classe concreta já desenvolvida anteriormente (`Pessoa`, `Data`, etc.).

a) Construa agora a classe principal. Escreva quatro métodos, cada um dos quais assume como um argumento um objeto das interfaces criadas.

b) Teste na função `main` várias possibilidades de utilização de cada interface. Por exemplo:

```
IDemo123 id123 = new Classe123();
metodo1(id123);
metodo2(id123);
...
IDemo1 id1 = new Classe123();
...
```

Verifique (e utilize) igualmente quais os métodos que tem disponíveis para cada referência.