

## PRÁTICA 5

1. No projeto elaborado em sala na Aula08 (ListaEstaticaAluno.zip – está no Moodle), codifique na classe Cadastro os seguintes métodos:
  - a. `exibirAprovados()` – exibe os dados somente dos alunos aprovados, considerando-se que a média para ser considerado aprovado é 5.
  - b. `exibirSomaMedia()` – exibe a soma das médias de todos os alunos.
  - c. `exibirAlunosGraduacao()` – exibe os dados apenas dos alunos de graduação. Suponha que no seu programa tenha um objeto **a** declarado como sendo da classe Aluno. Em C#, para verificar se o objeto **a** é do tipo AlunoGraduacao, basta fazer:

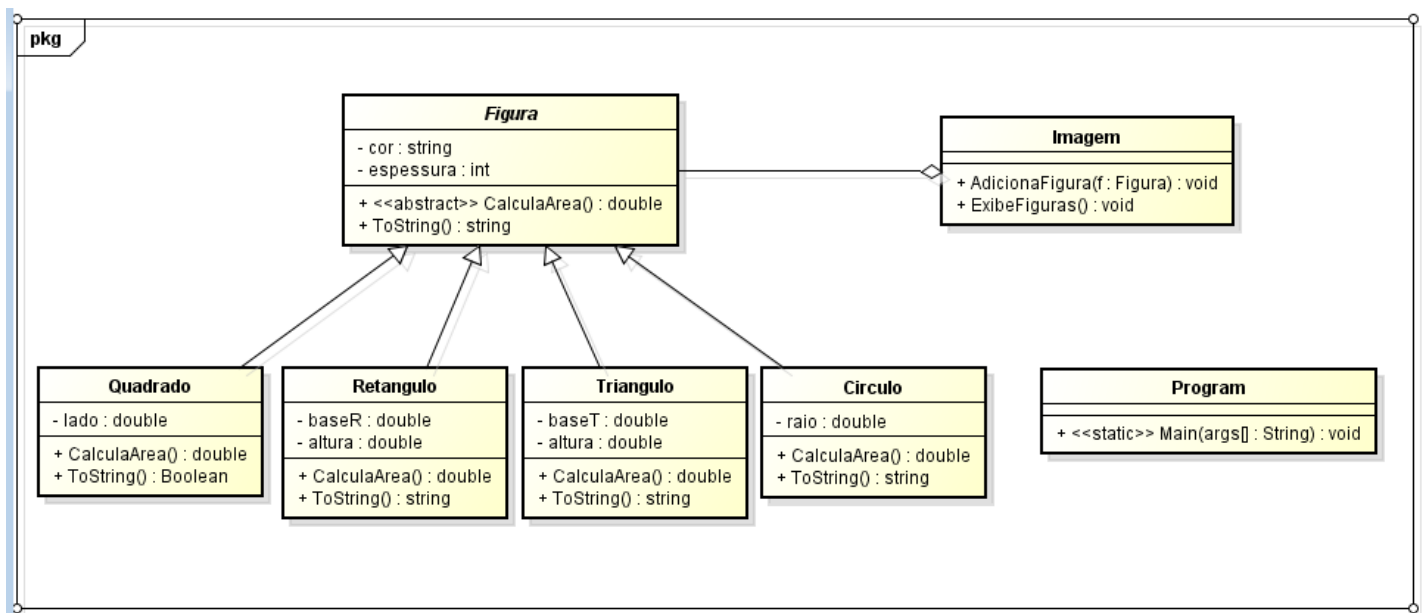
```
if (a is AlunoGraduacao)    // C#
```

Se o resultado for true, é porque **a** é um objeto da classe AlunoGraduacao.

Obs.: se fosse em Java, teria que fazer:

```
if (a instanceof AlunoGraduacao)    // Java
```

2. Considere a figura a seguir:



**Crie um projeto chamado ListaEstaticaFigura.**

Criar a **classe abstrata Figura**, que representa uma figura geométrica. Esta classe pode ter atributos como cor, espessura do contorno. Esses atributos devem estar encapsulados, e portanto, você deve prover os setters e os getters ou propriedades.

Essa classe também deve ter a declaração de um **método abstrato**

**float CalculaArea( ).**

Essa classe também deve ter o **método ToString()**, que sobrescreve o método ToString herdado da classe Object e retorna uma string contendo os dados da Figura.

Criar as **classes herdeiras (concretas) de Figura: Quadrado, Retangulo, Triangulo, Circulo**, com seus atributos, setters e getters. Essas classes deverão conter a implementação do método CalculaArea( ), que deverá calcular e retornar a área da figura (esse método é aquele que foi definido como abstrato na classe Figura). Essas classes deverão ter também o método ToString() que sobrescreve o método ToString da classe Figura e retorna uma String contendo os valores de seus atributos e o valor da sua área. Atenção: cada classe herdeira deverá ter como atributo o necessário para se calcular a sua área.

Obs: área do quadrado = lado \* lado

área do retângulo = base \* altura

área do triângulo = base \* altura / 2

área do círculo =  $\pi$  \* raio \* raio

Em C# e em Java, Math.PI fornece a constante equivalente ao número  $\pi$

Dentro do **método Main**, na classe Program, implemente o código para criar objetos das classes Quadrado, Retangulo, Triangulo, Circulo, e atribuir valores para os seus atributos e chamar o método CalculaArea() de cada objeto para verificar se está calculando corretamente.

Criar uma **classe Imagem**, que tem como atributo uma lista de objetos Figura. Utilize a classe ListaObj que tem no ListaEstaticaAluno.zip (Moodle).

Essa classe deve ter os métodos:

**void AdicionaFigura(Figura f)**

Recebe um objeto Figura e acrescenta esse objeto à lista.

**void ExibeFiguras( )**

Percorre a lista de Figuras e exibe as informações de cada objeto, incluindo a área da figura.

**void ExibirSomaArea( )**

Este método deve exibir a soma das áreas de todas as figuras que compõem a imagem.

**void ExibirFiguraAreaMaior20( )**

Este método deve exibir os dados de todas as figuras que compõem a imagem e que possuam área maior do que 20

**void ExibirQuadrado( )**

Este método deve exibir os dados de todas as figuras Quadrado que compõem a imagem.

No método Main, crie um objeto da classe Imagem, adicione os objetos de Quadrado, Retangulo, Circulo, Triangulo ao objeto Imagem. Teste os demais métodos e verifique se está executando corretamente.

Responda (revisão de conceitos de Programação Orientada a Objetos):

- a. Esse sistema de Figuras e Imagem apresenta o Polimorfismo? Justifique.
- b. De que maneira o Polimorfismo contribui para a reutilização de código?
- c. De que maneira o conceito de classe abstrata e método abstrato colaboram para a preparação do sistema para classes que poderão existir somente no futuro? Justifique utilizando este sistema como exemplo.