



Nota: 10.0



Tempo: Terminou



Sair

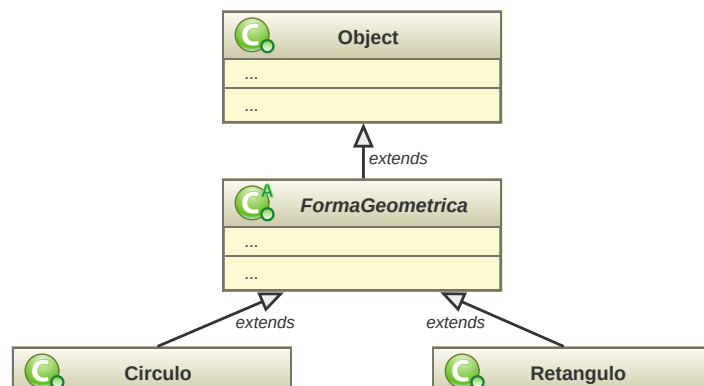
## Laboratório 6: Pacotes, Herança

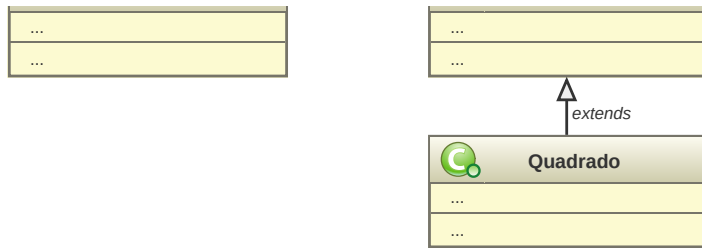
### Objetivo

- Exercitar os conceitos de [pacotes](#), [herança](#), [sobreposição](#), [polimorfismo](#) e [classes abstratas](#).
- Use os slides do arquivo "" como exemplos de código em Java para resolver este laboratório.
- Neste trabalho, você implementará a hierarquia de classes ao lado para representar formas geométricas.
- O detalhamento das classes, seus atributos e métodos, serão descritos nas questões abaixo.

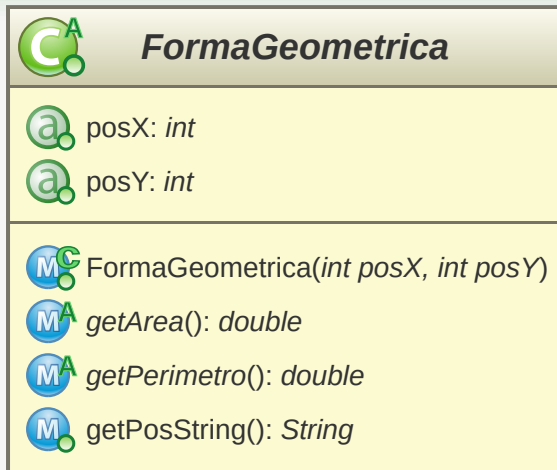
### Observações **Importantes**

- Nome do projeto: "Lab-Heranca".
- As classes deste trabalho deverão estar dentro do pacote `br.edu.icomp.ufam.lab_heranca`
- Para isso, ao criar uma nova classe, vá em "File" → "New" → "Class" e preencha o campo "package".
- Como estamos usando pacotes, use o modificador `public` na declaração das classes, dos atributos e dos métodos. Explicaremos isso na próxima aula.





## Questão 1: Classe FormaGeometrica



Crie uma classe para representar uma **FormaGeometrica**. Uma **FormaGeometrica** possui os atributos `posX` e `posY` indicando a posição da forma na tela. Esta classe possui os métodos:

- `getPosString`: retorna uma **String** contendo a posição da forma geométrica na tela conforme o exemplo abaixo (para `posX = 32` e `posY = 87`):  

```
|posição (32, 87)
```
- `getArea()`: método abstrato, não possui implementação.
- `getPerimetro()`: método abstrato, não possui implementação.

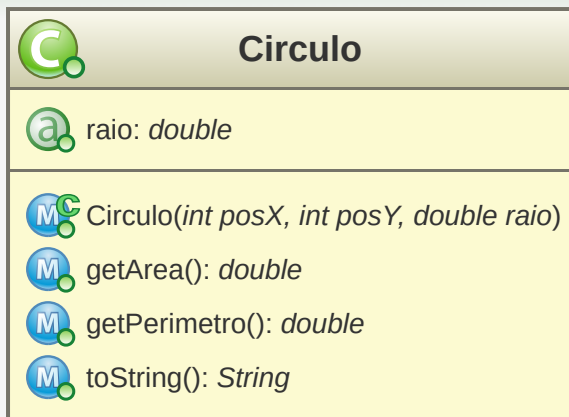
Note que como a classe possui métodos abstratos, ela também deverá ser declarada como abstrata. Note também que como a classe é abstrata, você não terá como gerar instâncias dela (objetos) para testar o método `getPosString`.

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "FormaGeometrica.java"

Solução correta!

## Questão 2: Classe Circulo



Crie uma classe para representar um **Circulo**. Um **Circulo** é uma subclasse da classe **FormaGeometrica** que, além dos atributos e métodos herdados, possui o atributo `raio` e os métodos:

- `getArea()`: implementação do método abstrato que veio da superclasse. Retorna a área do círculo. **Importante:** use a constante `PI` do java para realizar o cálculo (veja dicas abaixo).
- `getPerimetro()`: implementação do método abstrato que veio da superclasse. Retorna o perímetro do círculo. **Importante:** use a constante `PI` do java para realizar o cálculo (veja dicas abaixo).
- `toString`: sobrepõe o método `toString` da classe **Object**. Deve retornar uma descrição do círculo atual conforme exemplo abaixo. **Importante:** como este método sobrepõe outro, este precisa ter o mesmo modificador de acesso (ou um mais permissível) que o original. Neste caso, o método precisa ser **public** (note o círculo verde na descrição do método no diagrama de classes ao lado).

`|Circulo na posição (32, 87) com raio de 6.0cm (área=113.09733552923255cm2, perímetro=37.69911184307752cm)`

### Dicas:

- A área de um círculo é dada pela fórmula:  $A = \pi r^2$
- O perímetro de um círculo é dado pela fórmula:  $C = 2\pi r$
- Use a constante `PI` presente na classe **Math** do Java para pegar o valor de  $\pi$ . Exemplo: `double pi = Math.PI;`

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "Circulo.java"

Solução correta!

### Questão 3: Classe Retangulo



**Retangulo**



largura: *double*