

The background features abstract green geometric shapes. On the left, a thin, elongated green triangle points downwards. On the right, a complex arrangement of overlapping green triangles and polygons in various shades of green (from light lime to dark forest green) creates a layered, architectural effect. A thin, light gray line extends diagonally from the bottom left towards the right, passing behind the text.

Interface

Prof. Nelson Bellincanta Filho

Interface

- ▶ Mecanismo de generalização - limitada a padronização de interface;
- ▶ Não tem relação com herança;
 - ▶ Não é um mecanismo de reuso de código;
- ▶ Em sua forma mais simples, a interface pode conter métodos sem implementação e constantes;

Interface

- ▶ Os métodos das interfaces são implicitamente públicos e abstratos;
 - ▶ Os métodos deverão ser implementados por todas as classes que implementam a interface;
- ▶ As constantes das interfaces são, por padrão, públicas, estáticas e finais;
- ▶ Além de métodos abstratos e constantes, interfaces podem conter métodos default, estáticos e privados.

Interface

```
1 public interface Impressao {  
2  
3     public static final String INICIO = "<inicio>";  
4     public static final String FIM = "<fim>";  
5     public abstract void imprimir ();  
6     public abstract void imprimirNoConsole ();  
7 }
```

```
1 public interface Editavel {  
2     long chave = 0xABDCEF0; // hexadecimal  
3  
4     // método abstrato  
5     void editar(String conteudo);  
6 }
```

The background features abstract green geometric shapes. On the left, a solid green triangle points downwards. On the right, a complex arrangement of overlapping, semi-transparent green triangles and polygons in various shades of green creates a layered, architectural effect. A thin, light gray line extends from the bottom left towards the right, passing through the green shapes.

Usando interfaces

Usando interfaces

- ▶ Uma classe que deva possuir as operações definidas por uma interface deve implementá-las, fazendo essa indicação com a palavra reservada **implements**;
- ▶ Assim, a classe estabelece um contrato com a interface declarada, tornando obrigatória a implementação de seus métodos abstratos para que se torne concreta;
- ▶ Uma classe pode implementar ou realizar tantas interfaces quanto necessário.

Usando interfaces

```
public interface Geometria {  
    public float getPerimetro();  
    public float getArea();  
}
```

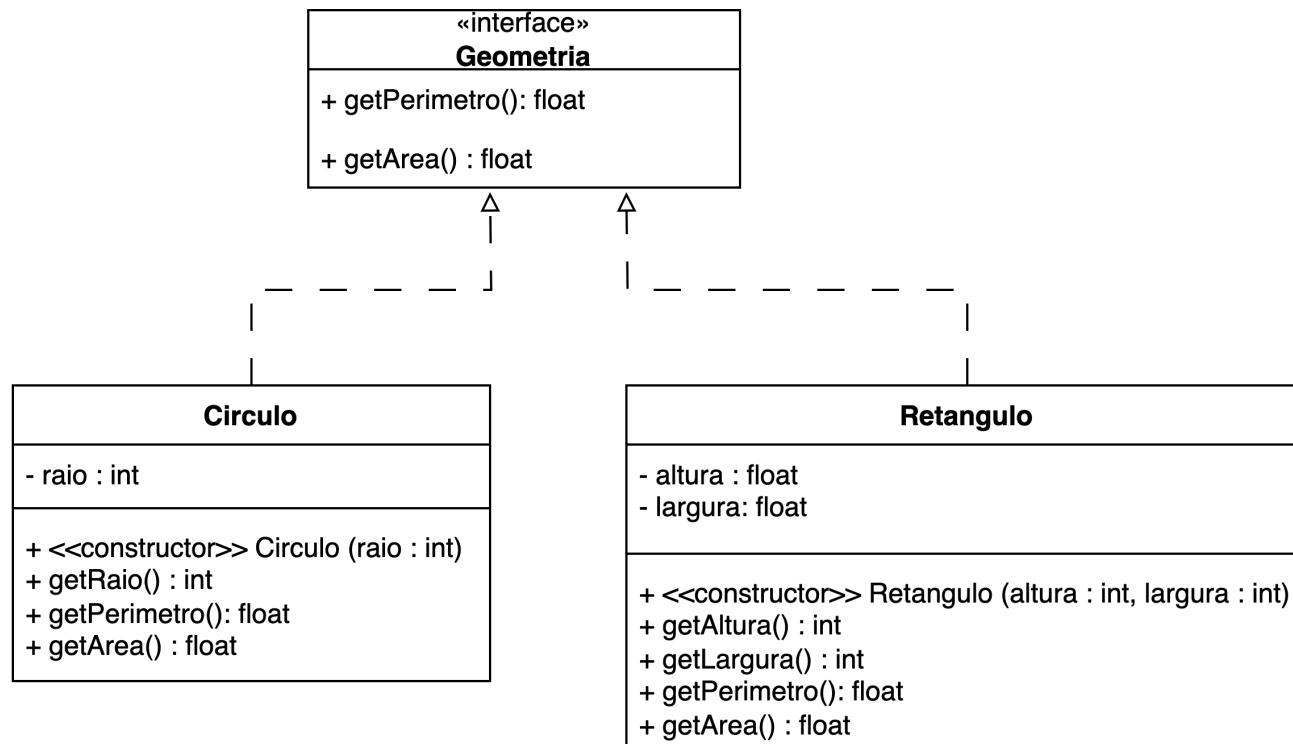
Interface em Java.

```
«interface»  
Geometria  
  
+ getPerimetro()  
+ getArea()
```

Interface em UML.

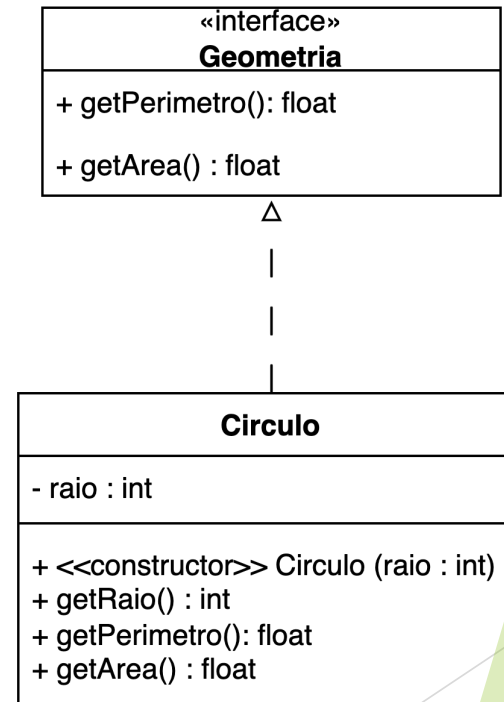
- Todas as classes que a implementarem a interface Geometria precisarão implementar os métodos getPerimetro() e getArea(), com as assinaturas indicadas.

Usando interfaces



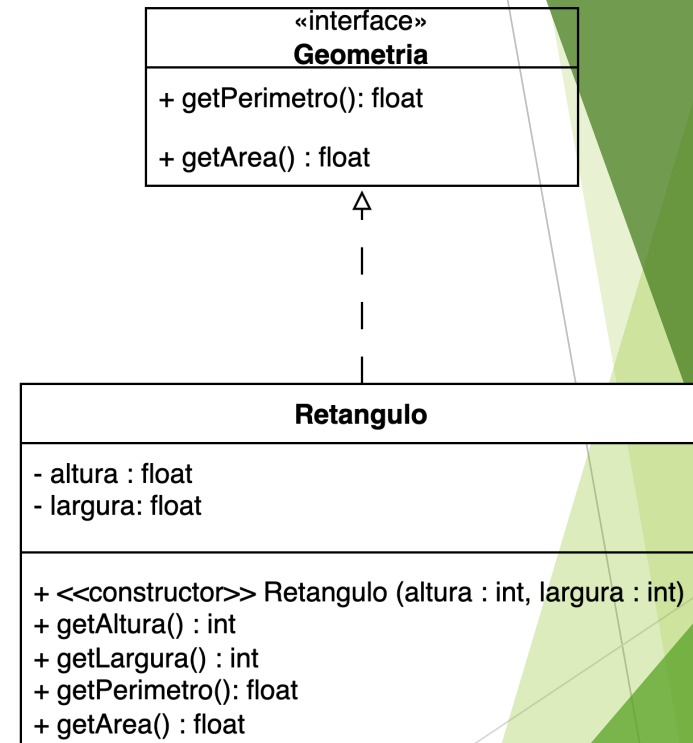
Usando interfaces

```
1 public class Circulo implements Geometria {  
2     ....  
3  
4     public float getPerimetro() {  
5         return 2 * Circulo.PI * raio;  
6     }  
7  
8     public float getArea() {  
9         return Circulo.PI * raio * raio;  
10    }  
11 }
```



Usando interfaces

```
1 public class Retangulo implements Geometria {  
2     ...  
3     public float getPerimetro() {  
4         return 2 * (altura + largura);  
5     }  
6  
7     public float getArea() {  
8         return altura * largura;  
9     }  
10 }
```



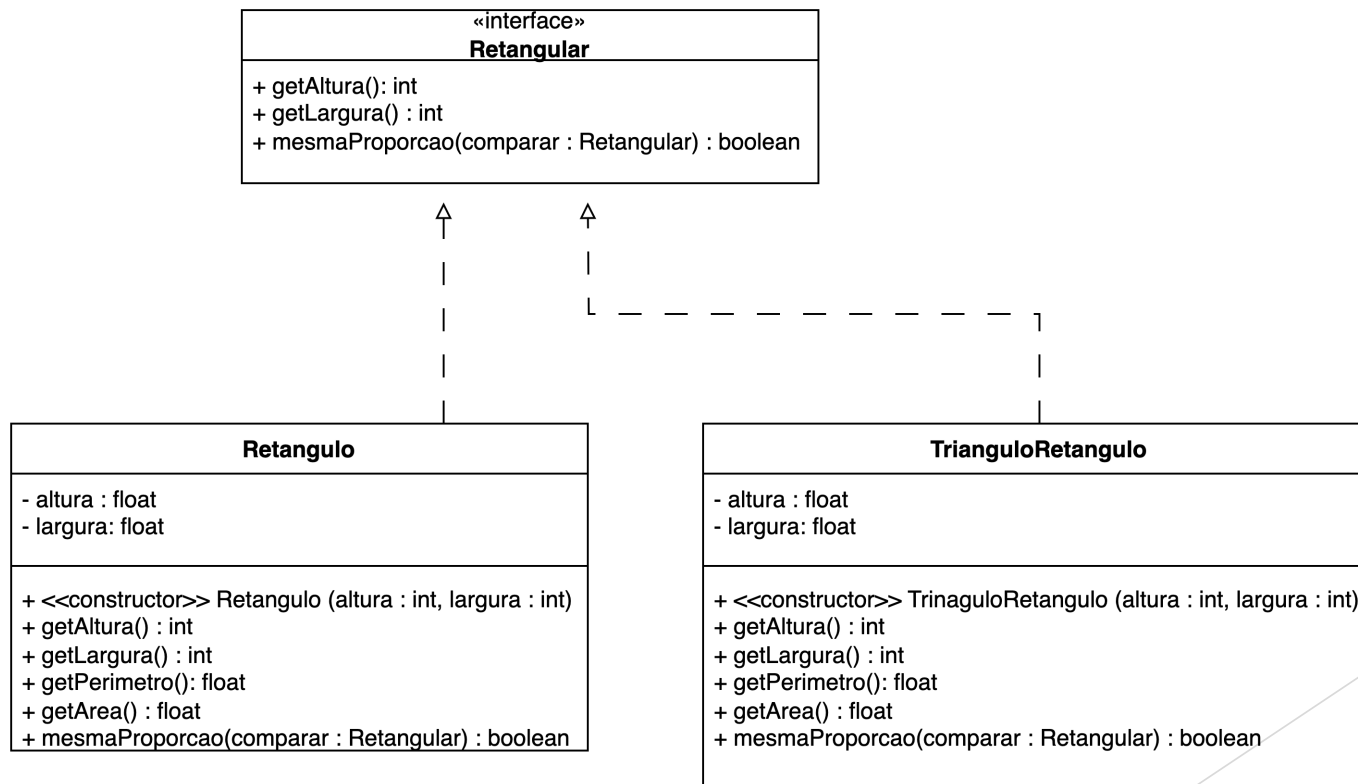
Usando interfaces

- ▶ Uma variável declarada em uma interface **A**, pode referenciar qualquer classe **B** que implemente a interface;
- ▶ Através dessa variável, só estarão "visíveis: ”
 - ▶ Métodos declarados na interface **A**.

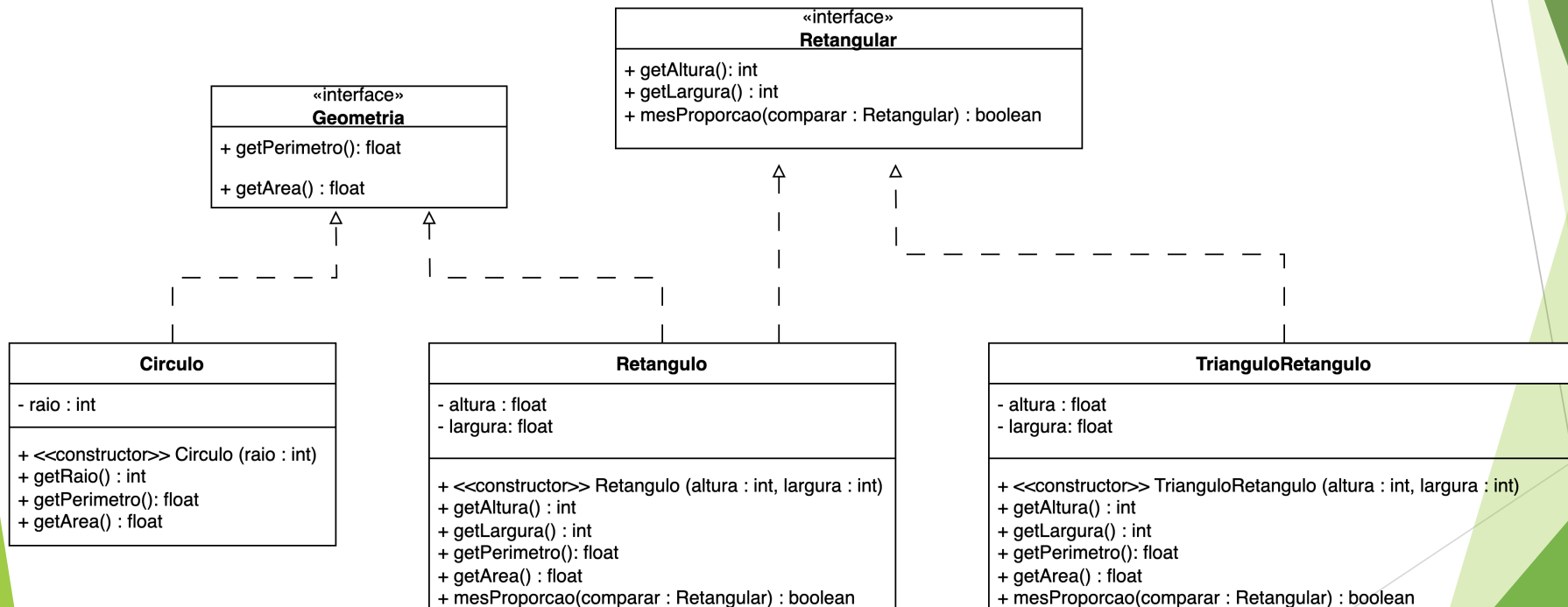
Tópicos avançados em Interfaces

An abstract graphic design featuring various shades of green. On the right side, there is a large, complex shape composed of overlapping triangles and polygons in different green tones, ranging from light lime to dark forest green. A thin, light gray line extends from the bottom left towards the center of this green shape. On the far left edge, there is a small, solid green triangle pointing towards the center.

Recebendo uma interface como parâmetro

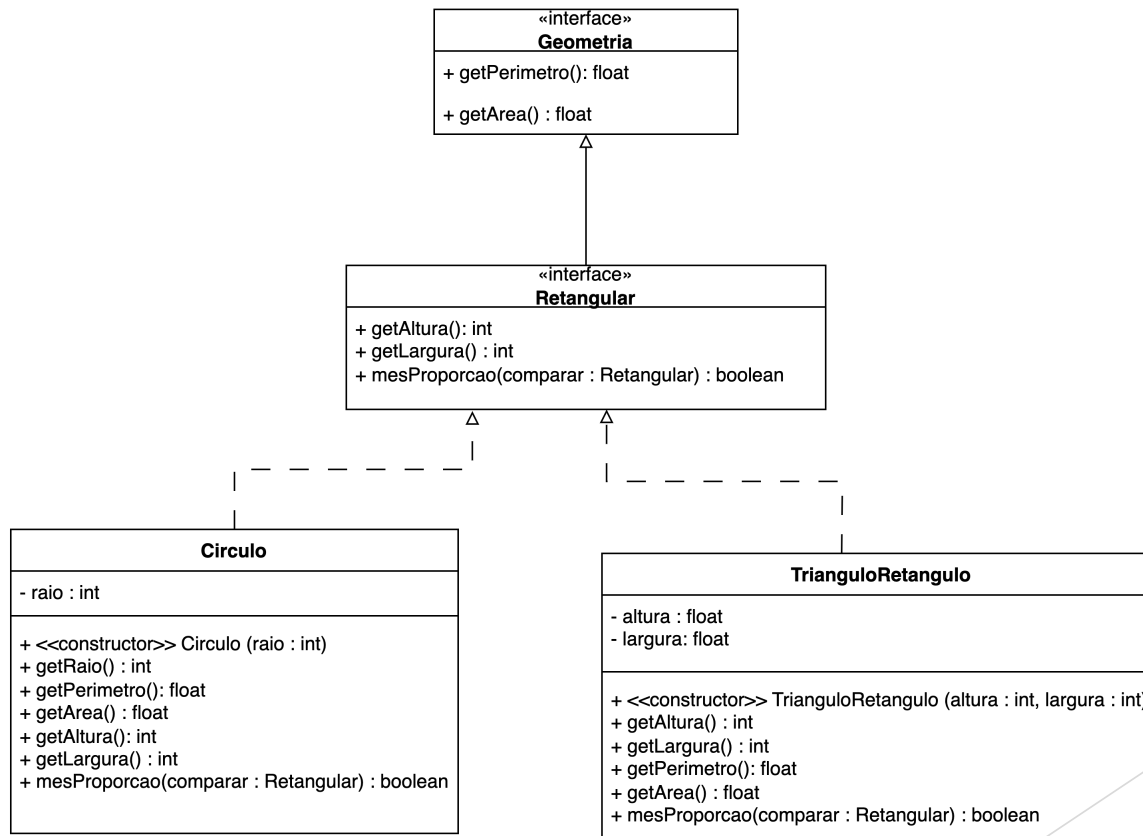


Implementando múltiplas interfaces



Herança de Interfaces

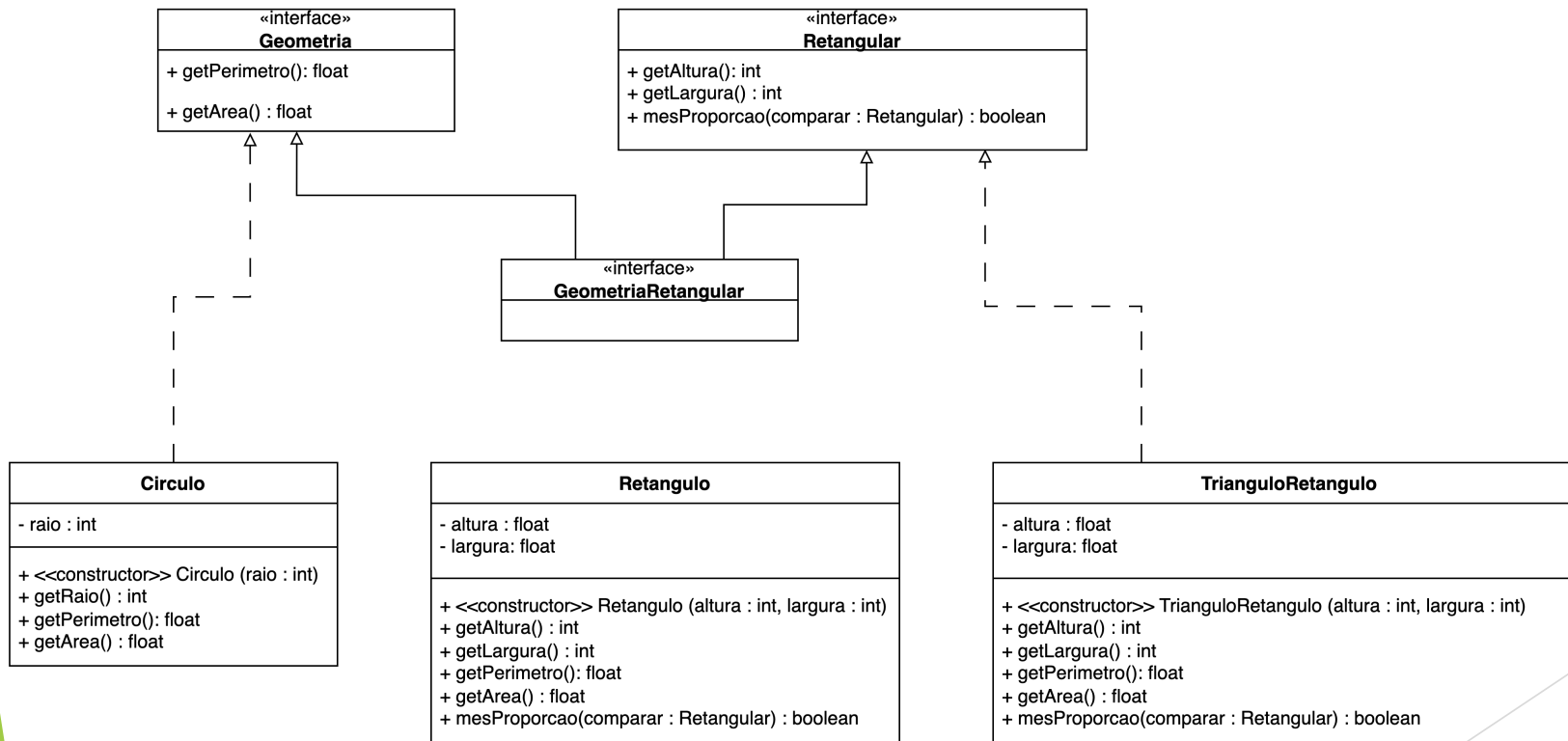
- Mecanismo de extensão de interface a partir da existente.



Herança Múltipla de Interfaces

- ▶ Uma interface pode ser herdeira de mais de uma interface.
- ▶ Nesse caso, será requerido a combinação de todos os métodos declarados nas interfaces que ela herda (mais métodos que ela possa acrescentar);

Herança Múltipla de Interfaces



Referências

FURGERI, Sérgio. **Java 8: Ensino Didático**. 1ª ed., São Paulo: Érica, 2015.

DEITEL, Harvey M.; DEITEL, Paul J. **Java: como programar**. 10ª ed. Editora Pearson Education do Brasil, 2017.

JUNIOR, Peter Jandl. **Java Guia do Programador**. 4a Edição: Atualizado para Java 16. Novatec Editora, 2021.