

Programação Orientada a Objetos em C++

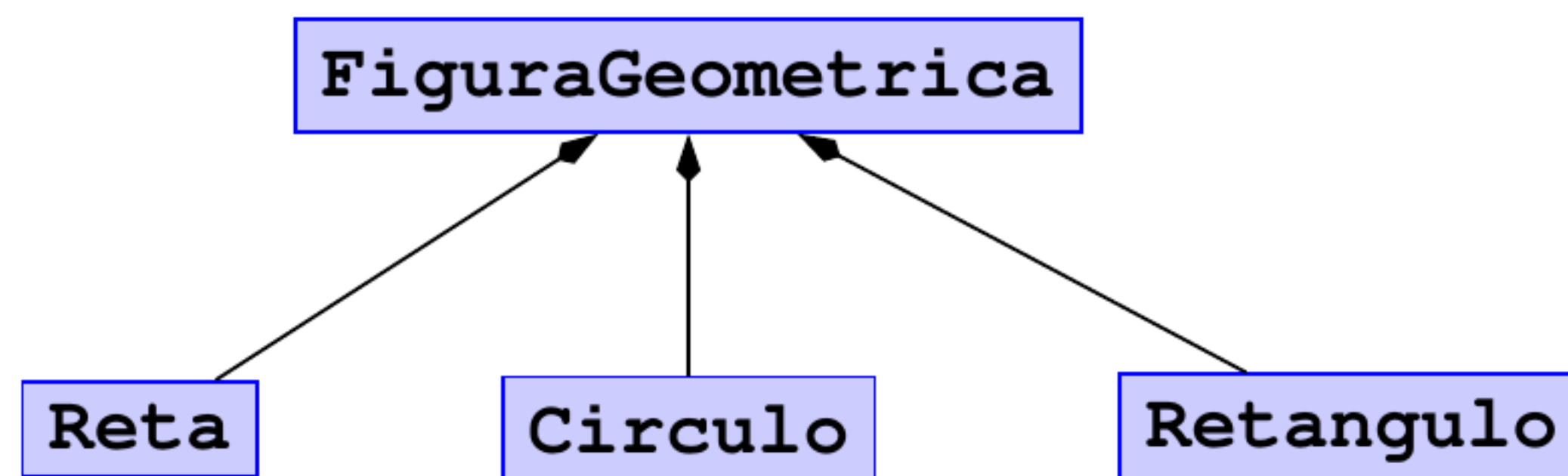
Polimorfismo e Classes Abstratas

Agostinho Brito

2020

- Considere uma situação-problema onde o desenvolvedor precisa criar uma interface gráfica para uma aplicação de desenho.
- Nessa aplicação, diversas figuras geométricas diferentes deverão aparecer, poderão estar superpostas umas às outras, mas compartilham muitas vezes propriedades comuns, como a cores de desenho, tipos de traço de linha, posição na tela.
- As mesmas estruturas também podem estar sujeitas a transformações comuns na interface visual, como rotação, espelhamento ou mudanças de escala.
- Como criar uma estrutura de classes simples para atender a essa demanda?

- Possível solução: criar uma classe base `FiguraGeometrica` e derivar as demais dessa.
- Considere um estágio inicial dessa ferramenta onde houvessem apenas três classes de figuras: `Reta`, `Circulo` e `Retangulo`.



- Poder-se-ia criar uma lista de elementos do tipo `FiguraGeometrica*` e armazenar nesta ponteiros para os elementos das subclasses criadas.

 **Praticando** `FiguraGeometrica...`

Algumas exigências...

- A subclasse PRECISA implementar o método `draw()`;
- A subclasse PRECISA implementar o método `area()`;
- A subclasse PRECISA implementar o método `posicao()`;
- A subclasse PRECISA implementar ...

???

Como forçar a subclasse a implementar os métodos desejados?

Algumas exigências...

- A subclasse PRECISA implementar o método `draw()`;
- A subclasse PRECISA implementar o método `area()`;
- A subclasse PRECISA implementar o método `posicao()`;
- A subclasse PRECISA implementar ...

???

Como forçar a subclasse a implementar os métodos desejados?

!!!

Criando uma classe abstrata

Criando uma classe abstrata

- Uma classe abstrata é uma classe que possui pelo menos um método virtual puro.
- Métodos virtuais são aqueles que sempre são ativados pelas subclasses nos processos de polimorfismo.
- Métodos virtuais **PUROS** não possuem implementação.
- Entretanto, as classes derivadas são obrigadas a implementá-los.

```
1  class A{  
2  ...  
3  public:  
4      virtual void metodo ()=0; // metodo virtual puro  
5  }
```


Criando uma classe abstrata

- Objetos de uma classe abstrata **NÃO PODEM SER INSTANCIADOS**.
- Masssss, ponteiros para ela podem existir.
- Ponteiros para superclasses apontando para objetos de subclasses são usados para construir o quarto pilar da Programação Orientada a Objetos: o **POLIMORFISMO**.

```
1  class A{
2  ...
3  public:
4      virtual void metodo ()=0;
5  }
6  class B : public A{
7  ...
8  public:
9      void metodo () { ... };
10 }
```

 Praticando classes abstratas...



Obrigado