

# Modularização em C++

Prof. Andrei Braga



# Conteúdo

- Aprendizado de programação nas disciplinas cursadas até agora
- Modularização
- Modularização na linguagem C++

# Aprendizado de programação até agora

- Programação estruturada (ou procedural)
  - Que funções (ou procedimentos) o programa deve ter?
- Estruturas de dados fundamentais
- Programação orientada a objetos
  - Que entidades (as principais) existem no programa?
  - Que objetos existem no programa?

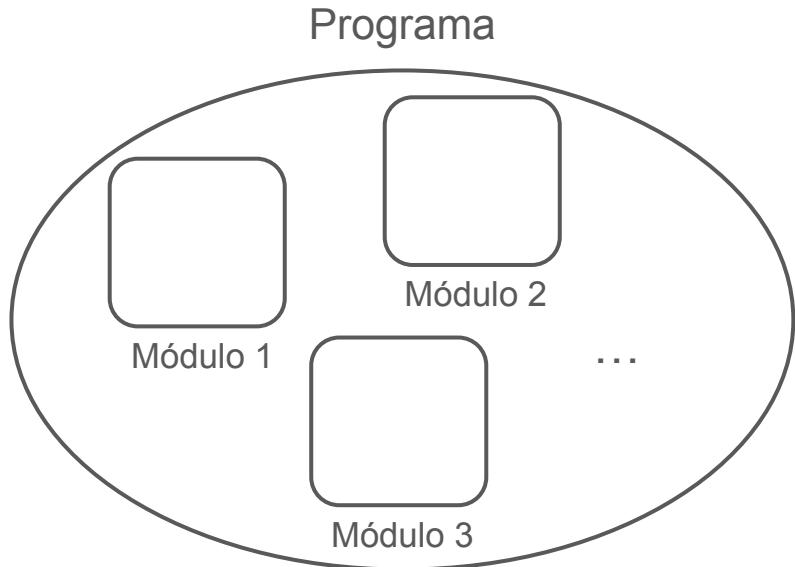
Algoritmos e  
Programação

Estruturas de  
Dados

Programação I

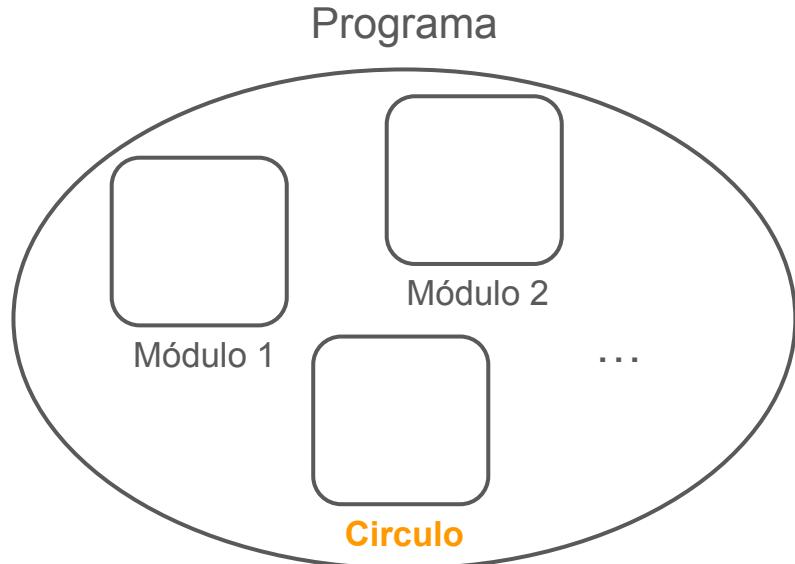
# Padrão de programação

- Como organizar o programa em partes tal que uma parte possa ser utilizada (várias vezes) dentro do programa?
  - **Modularização**
- Uma parte (um módulo) do programa pode representar, por exemplo,
  - um tipo de dado complexo (uma estrutura de dados) ou
  - uma classe de objetos



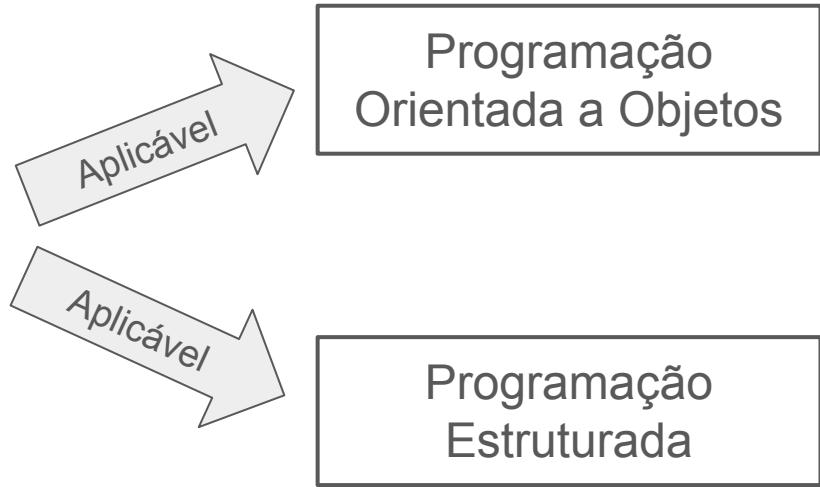
# Padrão de programação

- Como organizar o programa em partes tal que uma parte possa ser utilizada (várias vezes) dentro do programa?
  - **Modularização**
- Uma parte (um módulo) do programa pode representar, por exemplo,
  - um tipo de dado complexo (uma estrutura de dados) ou
  - uma classe de objetos



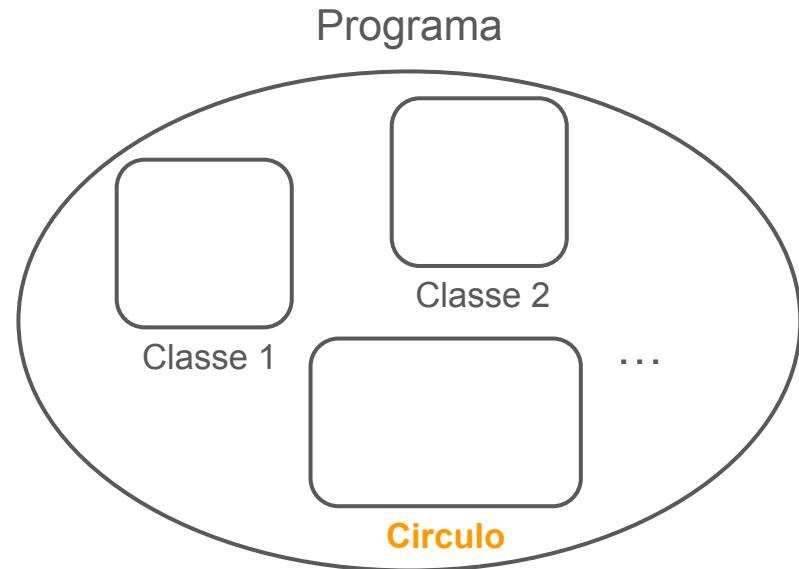
# Padrão de programação

- Como organizar o programa em partes tal que uma parte possa ser utilizada (várias vezes) dentro do programa?
  - **Modularização**
- Uma parte (um módulo) do programa pode representar, por exemplo,
  - um tipo de dado complexo (uma estrutura de dados) ou
  - uma classe de objetos



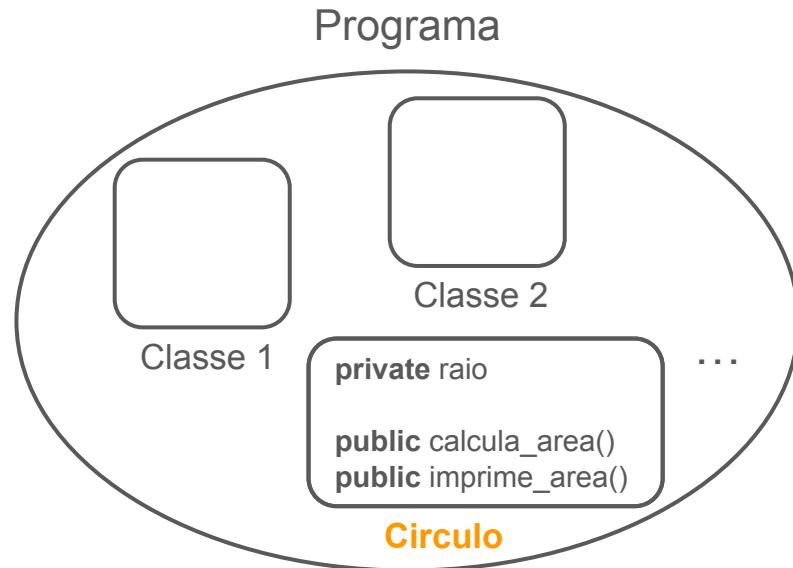
# Modularização na linguagem C++

- Podemos entender uma classe como uma coleção de dados e operações tal que alguns destes dados e operações podem ser acessados por outras partes do programa



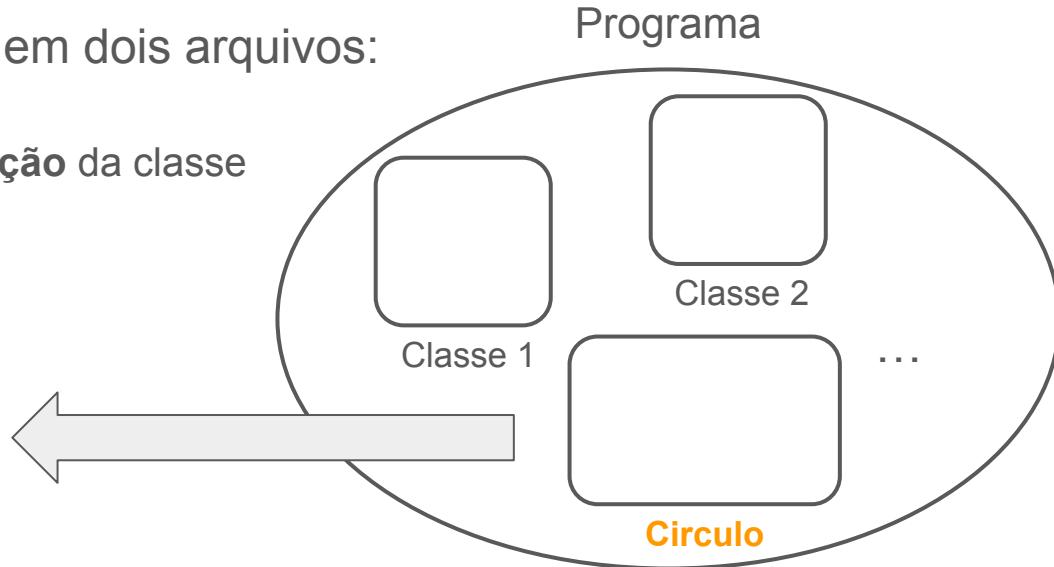
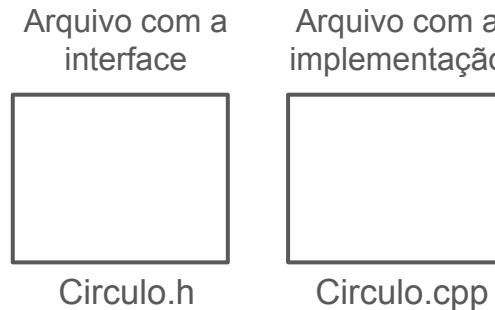
# Modularização na linguagem C++

- Podemos entender uma classe como uma coleção de dados e operações tal que alguns destes dados e operações podem ser acessados por outras partes do programa
- Os dados e operações de uma classe que podem ser **acessados por outras partes** do programa são comumente declarados como **públicos**



# Modularização na linguagem C++

- Podemos entender uma classe como uma coleção de dados e operações tal que alguns destes dados e operações podem ser acessados por outras partes do programa
- Em C++, uma classe é escrita em dois arquivos:
  - um arquivo com a **interface** e
  - um arquivo com a **implementação** da classe



# Modularização na linguagem C++

- Em C++, o arquivo com a **interface** de uma classe contém as **declarações** dos dados e operações (métodos) da classe
- O arquivo com a **implementação** da classe contém os **códigos** que são executados pelas operações (métodos) da classe

Arquivo com a interface

```
class Circulo {  
public:  
    Circulo(double raio);  
  
    double calcula_area();  
    void imprime_area();  
private:  
    double raio_;  
};
```

Circulo.h

Arquivo com a implementação

```
Circulo::Circulo(double raio) {  
    ...  
}  
double Circulo::calcula_area() {  
    ...  
}  
void Circulo::imprime_area() {  
    ...  
}
```

Circulo.cpp

# Modularização na linguagem C++

- (Exemplo completo)

# Exercícios

Após implementar a classe `Circulo` descrita nos slides anteriores, faça o seguinte:

1. Adicione à classe `Circulo` métodos que calculam e imprimem o perímetro do círculo. Teste estes métodos na parte principal do programa.
2. Na parte principal do programa, crie dois círculos e imprima os seus perímetros.

# Exercícios

3. Na parte principal do programa, faça o seguinte:
  - 3.1. Crie uma pilha cujos elementos são valores inteiros.
  - 3.2. Insira dois elementos na pilha.
  - 3.3. Remova os dois elementos da pilha imprimindo os seus valores.

# Referências

- Uma explicação mais detalhada sobre modularização na linguagem C++ pode ser encontrada no seguinte livro:  
Lippman, S. B., Lajoie, J. and Moo, B. E. C++ Primer, 5th. ed., Addison-Wesley, 2013.