

Programação Orientada a Objetos

CLASSES

em

C++

Introdução

- A **POO** **abstrai** os problemas **usando classes**
 - Carro é uma classe constituída por

Dados

Atributos

- Cor
- Tipo
- Velocidade máxima

Funções

Métodos

- Acelerar
- Frear
- Ligar

O carro é uma
abstração de um
veículo real



Introdução

- Uma classe **representa** vários **objetos**

```
class Carro
{
private:
    int cor;
    string tipo;
    float velocidade;

public:
    void ligar();
    void desligar();
    void acelerar();
    void frear();
};
```



Classes e Tipos

- A um **tipo** é **associado**:

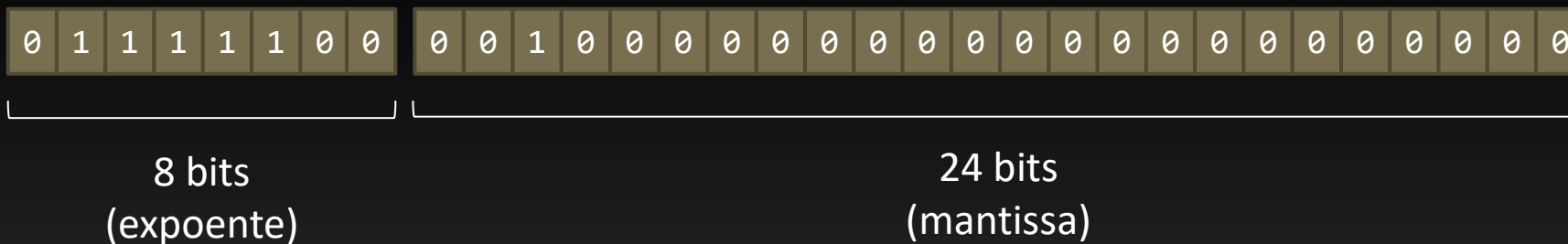
Dados

- Uma **representação**:
 - Quantidade de memória
 - Organização dos dados
 - Interpretação para os bits

Funções

- Um conjunto de operações
 - Somar / Subtrair
 - Multiplicar / Dividir
 - Comparar

float (32 bits) = 0.15625

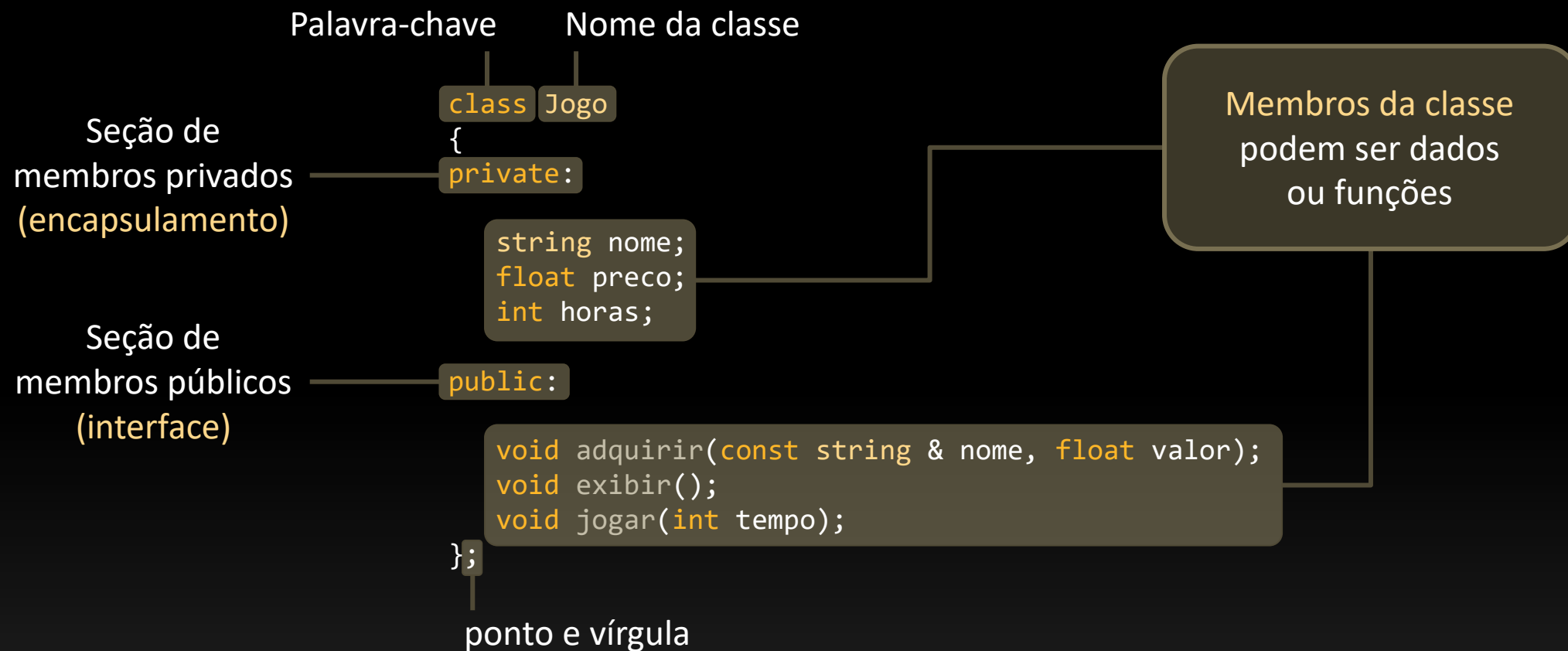


Classes e Tipos

- A **classe** é a forma do C++ **traduzir uma abstração em um tipo** definido pelo programador
 - A classe é definida em **duas partes**:
 - **Declaração:**
 - Descreve os atributos
 - Descreve os métodos
 - **Definição:**
 - Contém a implementação dos métodos

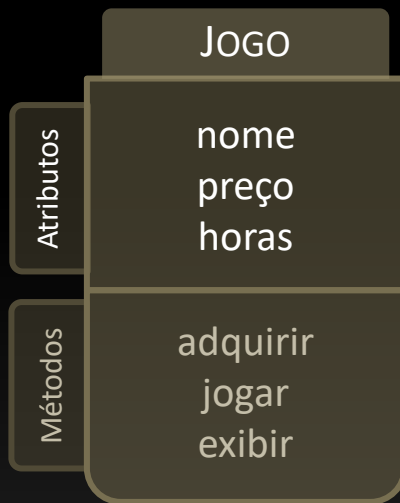
Assim como **registros**,
uma classe define um
novo tipo.

Declaração da Classe



Interface

- Os **métodos** são também chamados de **interface pública**
 - Eles fornecem acesso aos dados da classe
 - Garantem a integridade dos dados



```
// acesso direto aos atributos  
✗ horas = 2;  
  
// acesso através da interface pública  
// os métodos protegem contra alterações  
// incorretas dos atributos  
✓ jogar(2);
```

Interface

- Os **métodos** funcionam como uma interface

```
class Jogo  
{  
private:
```

```
    string nome;  
    float preco;  
    int horas;
```

```
public:
```

```
    void adquirir(const string & titulo, float valor);  
    void jogar(int tempo);  
    void exibir();
```

```
};
```

Os métodos
fornecem **acesso**
aos atributos



Encapsulamento

- A classe **encapsula** os atributos

```
class Jogo  
{  
  private:
```

```
    string nome;  
    float preco;  
    int horas;
```

```
  public:
```

```
    void adquirir(const string & titulo, float valor);  
    void jogar(int tempo);  
    void exibir();  
};
```

Os atributos são
ocultados do resto
do programa



Encapsulamento

- A classe fornece:
 - **Encapsulamento** dos atributos
 - **Interface** através dos métodos



Bateria

Contém produtos químicos

- Detalhes desconhecidos
- Encapsulados para proteção

Interface fornece energia

- Dispositivos compatíveis

A Classe Jogo

```
// declaração da classe Jogo
#include <string>
using std::string;

class Jogo
{
private:
    string nome;                // nome do jogo
    float preco;                // preço do jogo
    int horas;                  // quantidade de horas jogadas
    float custo;                // valor por hora jogada

    void calcular() { if (horas > 0) custo = preco/horas; } // calcular custo da hora jogada

public:
    void adquirir(const string & titulo, float valor);      // preencher informações
    void atualizar(float valor);                             // atualizar preço do jogo
    void jogar(int tempo);                                   // registrar horas jogadas
    void exibir();                                           // mostrar informações
};
```

A Classe Jogo

- As operações aparecem como **métodos da classe**

```
class Jogo  
{  
private:
```

« atributos da classe »

```
void calcular() { if (horas > 0) custo = preco/horas; }
```

```
public:
```

```
void adquirir(const string & nome, float valor);  
void atualizar(float valor);  
void jogar(int tempo);  
void exibir();
```

```
};
```

Métodos podem ser definidos **inline** ou representados por seus **protótipos**

A Classe Jogo

```
// definição da classe Jogo
#include <iostream>
#include "Jogo.h"

void Jogo::adquirir(const string & titulo, float valor)
{
    nome = titulo;
    preco = valor;
    horas = 0;
    custo = valor;
}

void Jogo::atualizar(float valor)
{
    preco = valor;
    calcular();
}

...
```

```
class Jogo
{
private:
    string nome
    float preco
    int horas;
    float custo;

    void calcular()
    {
        if (horas > 0)
            custo = preco/horas;
    }

    ...
}
```

A Classe Jogo

```
// continuação

...

void Jogo::jogar(int tempo)
{
    horas = horas + tempo;
    calcular();
}

void Jogo::exibir()
{
    std::cout << nome << " R$"
               << preco << " "
               << horas << "h = R$"
               << custo << "/h\n";
}
```

```
class Jogo
{
private:
    string nome
    float preco
    int horas;
    float custo;

    void calcular()
    {
        if (horas > 0)
            custo = preco/horas;
    }

    ...
}
```

Resumo

- A **declaração** de uma **classe** define:
 - Um **tipo criado pelo programador**
 - Formato de armazenamento dos dados
atributos da classe
 - Operações associadas
métodos da classe
- A classe fornece:
 - **Encapsulamento** dos atributos
 - **Interface** através dos métodos