

Programação Orientada a Objetos

INTRODUÇÃO

em

C++

Introdução

- **Programação Orientada a Objetos**

- É uma **abordagem** para o **desenvolvimento de programas** baseada nas ideias de:

- Abstração
- Encapsulamento
- Polimorfismo
- Herança

Um dos principais objetivos dessa abordagem é melhorar e facilitar o **reuso de código**.

Abordagem Estruturada

- Um programa é **constituído por**:

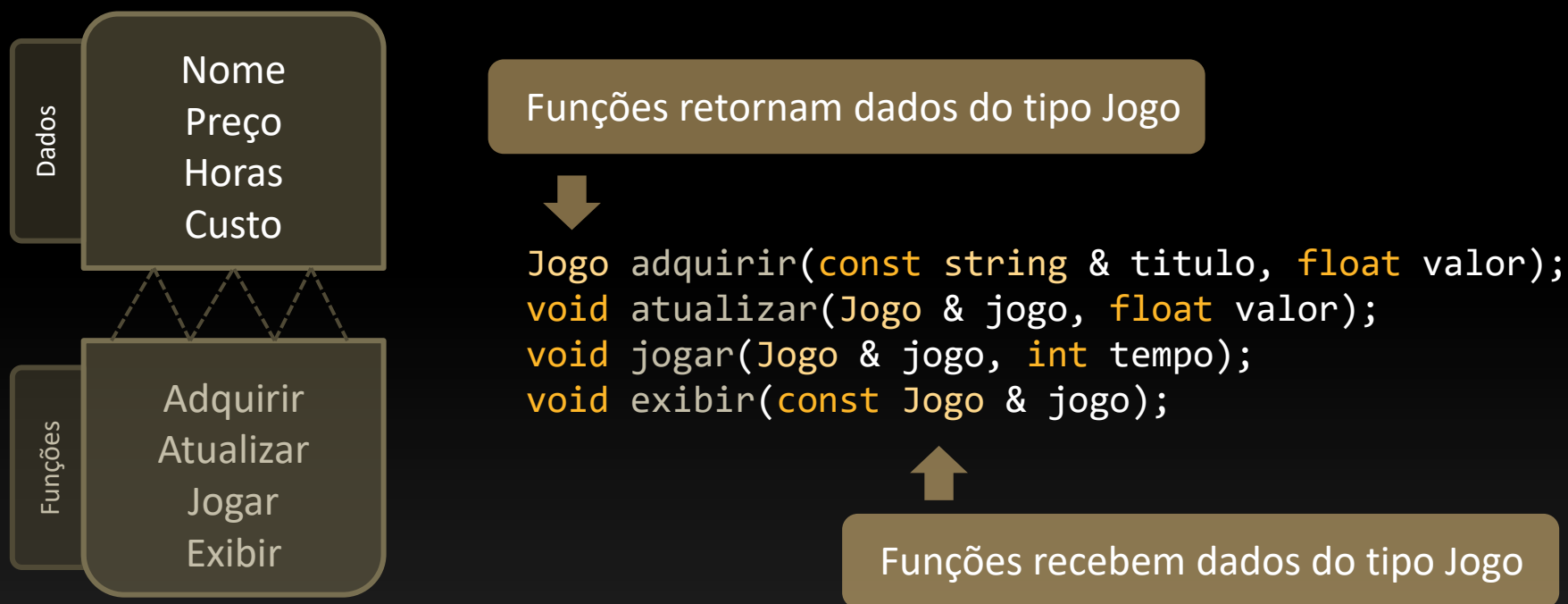


```
struct Jogo
{
    string nome;    // nome do jogo
    float preco;    // preço do jogo
    int horas;      // quantidade de horas jogadas
    float custo;    // valor por hora jogada
};
```

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

Abordagem Estruturada

- As **funções** facilitam a **manipulação dos dados**
 - Sempre indicamos sobre quais dados trabalhar



Abordagem Estruturada

- As **funções** facilitam a **manipulação dos dados**
 - Mas nada impede de acessarmos os dados diretamente



```
struct Jogo
{
    string nome;    // nome do jogo
    float preco;    // preço do jogo
    int horas;      // quantidade de horas jogadas
    float custo;    // valor por hora jogada
};

void jogar(Jogo & jogo, int tempo);

-----

Jogo gow;
jogar(gow, 2);    // jogar por 2 horas
gow.horas += 2;   // jogar por 2 horas
```

Abordagem Orientada a Objetos

- Um programa é **constituído por**:



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

Abordagem Orientada a Objetos

- As **funções** não precisam indicar **sobre qual dado agir**
 - Elas sempre agem sobre os dados associados



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

Abstração

- Um dos pilares da orientação a objetos é a abstração
 - A vida é cheia de complexidades
 - A abstração é uma forma de lidar com essa complexidade

"Abstração é a habilidade de concentrar nos aspectos essenciais de um contexto qualquer, ignorando características menos importantes ou acidentais."

Abstração

- **Carro** em um Jogo 2D

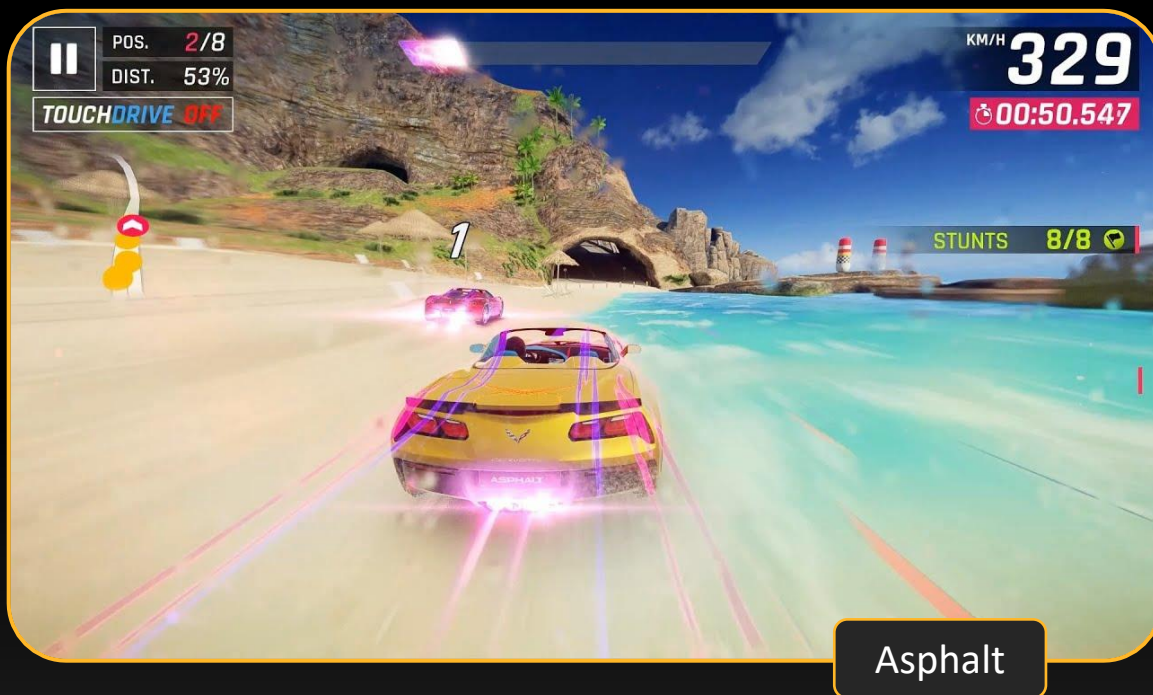
O carro em um jogo
é uma **abstração** de
um veículo real



Dados	Cor
	Tamanho
Funções	Acelerar
	Frear

Abstração

- Carro em um jogo 3D



Dados

Cor
Marca
Modelo
Velocidade
Aceleração

Funções

Esterçar
Acelerar
Frear
Nitro

Abstração

- **Carro** em uma simulação 3D



- Motor
 - Potência
 - Consumo
- Pneus
 - Aderência
 - Temperatura
 - Pressão
 - Desgaste

Abstração

- A **POO** **abstrai** os problemas **em Classes e Objetos**
 - Carro é uma classe

Dados

Atributos

Funções

Métodos

- Ele tem características
 - Cor
 - Tipo
 - Velocidade máxima
- Podemos definir operações sobre ele
 - Acelerar
 - Frear
 - Ligar

Abstração

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

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

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



Abstração

- Classes e objetos **abstraem o mundo real**

```
class Pessoa
{
private:
    string nome;
    int idade;
    float altura;

public:
    void comer();
    void trabalhar();
    void divertir();
    void dormir();
};
```



Resumo

- **Abstração** é o primeiro **conceito importante**
 - O mundo real pode ser abstraído
- A POO abstrai objetos através de uma classe
 - **Uma classe** agrupa/reúne/associa
 - Dados: atributos da classe
 - Funções: métodos da classe
 - Objetos são representantes de uma classe