

Computação Científica B

Programação Orientada a Objetos
(Classes, Objetos, Atributos e Métodos)

Prof. Ricardo A. S. Fernandes

Na Programação Procedural temos:

- **Dados do Programa → Variáveis;**
- **Ações que operam sobre os dados → Funções / Procedimentos;**
- **As Funções podem ser organizadas dentro do próprio programa ou criadas de forma genérica para atuarem como Bibliotecas;**
- **Muitas Funções fazem parte de uma gama de Bibliotecas padrão da Linguagem C ou C++, como <stdio.h>, <iostream.h> e <stdlib.h>**
- **Ao criar Funções como Bibliotecas, permite-se que estas possam ser reutilizadas por outros programadores;**
- **No entanto, a documentação de software procedurais não é algo trivial.**

Vale destacar que:

- Os Dados e Funções são independentes;
- Dados são fornecidos como entradas para Funções que, por consequência, poderão retornar um resultado;
- Há pouco (ou nenhum) controle de acesso às variáveis, ou seja, o programador deve se preocupar constantemente com a possibilidade de variáveis serem sobrescritas;
- Também não há um controle de acesso eficaz para as Funções;
- Esses fatores são motivadores para o paradigma de Programação Orientada a Objetos;
- A Programação Orientada a Objetos tem como base o modelo de programação procedural.

Conceito

- Representa uma nova forma de pensar em um programa, em que são feitas abstrações da vida real;
- Dentre essas abstrações tem-se os Objetos, Atributos, Métodos/Ações, Hierarquia, Herança, Polimorfismo;
- Assim, um programa desenvolvido nesse paradigma é composto com um conjunto de Objetos que se comunicam entre si;
- Neste sentido, os Dados e Funções passam a ser Encapsulados dentro de cada Objeto;
- Esse conceito de Encapsulamento é o que permite que os Dados e Funções estejam protegidos de acessos indevidos;
- No entanto, todo Objeto somente pode ser criado a partir de uma Classe.

Classes

- A criação de Classes e da forma como os objetos oriundos das classes irão se comunicar é o que nos referimos como Abstração;
- A Abstração pressupõe que um determinado problema seja modelado a fim de reduzir sua complexidade. Portanto, apenas os aspectos essenciais do problema são modelados;
- Assim, uma Classe deve ser capaz de modelar Objetos, sejam eles reais ou abstratos. Por exemplo: Aluno, Venda, Biblioteca, Calculadora, Carro, Veículo, Jogador, Animal;
- Supondo que queremos modelar um programa para atuar em uma Loja de Veículos. Podemos ter as seguintes Classes: Vendedor, Carro, Moto;
- Nota-se, portanto, que toda Classe terá suas características (Atributos) e suas ações (Métodos).

Atributos de uma Classe

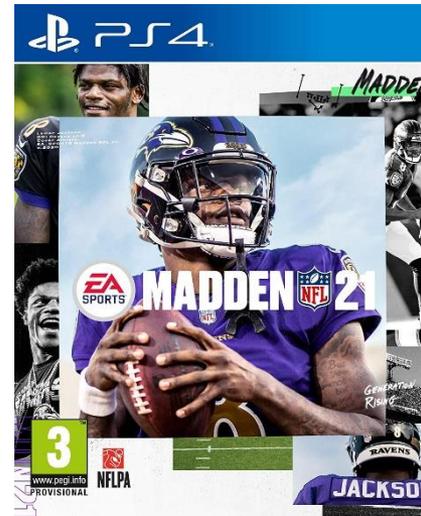
- **Considerando o exemplo da Classe Carro, possíveis Atributos seriam:**
 - **Cor;**
 - **Marca;**
 - **Modelo;**
 - **Quantidade de Portas;**
 - **Tipo do Motor;**
 - **Ano de Fabricação.**

Métodos de uma Classe

- **Considerando o exemplo da Classe Carro, possíveis Métodos seriam:**
 - **Ligar/Desligar;**
 - **Acelerar/Frear;**
 - **Trocar Marcha;**
 - **Abrir/Fechar Porta;**

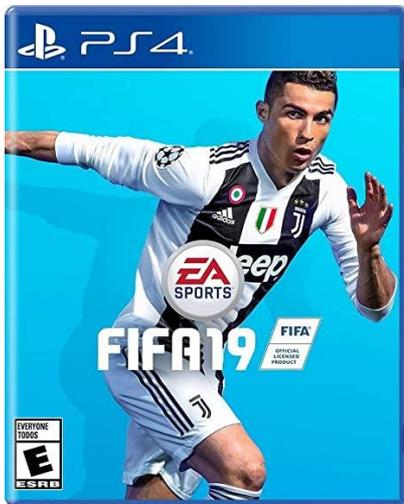
Mas qual a vantagem de se criar Classes?

- A reutilização de código se torna mais fácil, principalmente em grandes equipes de desenvolvimento;
- Ao se criar uma Classe, ela poderá ser facilmente reutilizada em outros projetos de software da mesma empresa;
- Exemplo:



Mas qual a vantagem de se criar Classes?

- Qual seria o custo se a EA Sports contratasse uma grande equipe de programadores para cada jogo?
- E se todo ano, em vez de reutilizar código, tivesse que começar tudo da estaca zero?
- O que realmente sofre alteração de uma versão para outra?



Como funciona a reutilização de Classes?

CLASSE



Nome: ?
Equipe: ?
Altura: ?
Peso: ?

Chutar ()
Cabecear ()
Correr ()

Instanciação

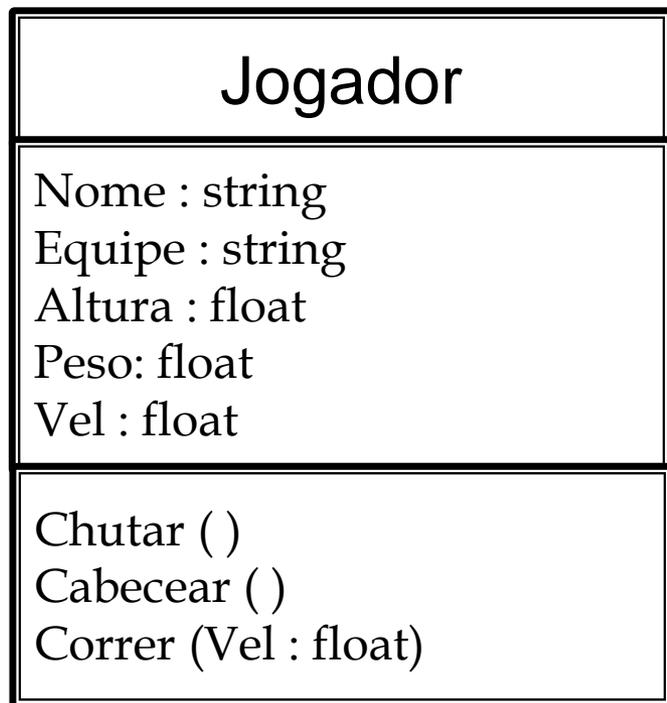


OBJETO 1



OBJETO 2

Representação de uma Classe com seus Atributos e Métodos



- A representação de uma classe é utilizada apenas na fase de Modelagem;
- Representação não é programação;
- Isso ajuda na documentação do programa e permite que toda a equipe tenha acesso;
- Por ser uma forma padronizada, um novo integrante da equipe não terá dificuldades em compreender o programa;
- Permite que o programa se mantenha atualizado, mesmo que a equipe seja alterada, ou seja, o software se mantém independente da equipe.