

Conceitos Básicos de Orientação a Objetos

Programação: abordagem tradicional

- Programa: sequência de instruções
- Ao longo do tempo, programas vão se tornando cada vez mais complexos
- Trechos repetitivos do código vão sendo agrupados em funções (sub-rotinas, procedimentos, etc)
- Modularização!

Orientação a Objetos: história

- A modularização é levada ao próximo nível
- Início: cerca de 50 anos
 - Anos 60: Simula-67, introduz alguns conceitos básicos da OO
 - Anos 70: Smalltalk
 - Anos 80: C++
 - Anos 90: Java
- Ficou mais “popular” há uns 20 anos

Por que usar Orientação a Objetos ?

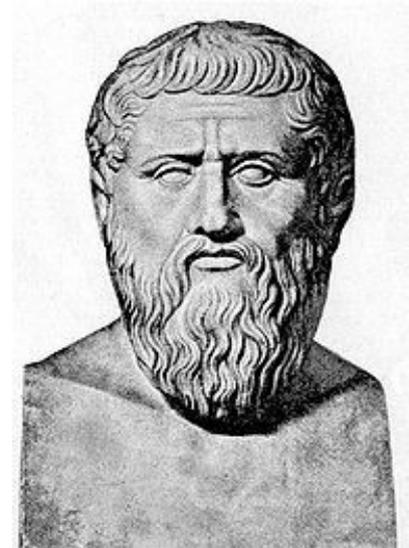
- Mais fácil desenvolver sistemas complexos
- Mais fácil reutilizar código
- Nível mais elevado de abstração

Ou seja

- O objetivo dessa disciplina é facilitar nossa vida, e não complicar.

Orientação a Objetos: Platão

- 428 a.C. – 348 a.C.
- Teoria das Ideias ou Teoria das Formas



Classe

- Classes são a definição de um novo tipo de dados que associa dados e funções em uma só estrutura” (N.N.C. Menezes)
- Ou seja, uma classe define um novo tipo de dados especificando:
 - Dados, também chamados de *atributos*
 - Funções que operam sobre esses dados, também chamadas de *métodos*

Objetos

- “Um objeto pode ser entendido como uma variável cujo tipo é uma classe, ou seja, um objeto é uma instância de uma classe.”
(N.N.C. Menezes)
- Em programas orientados a objetos são criados vários objetos, que interagem entre si.
- Podemos entender uma classe como um “molde” ou uma “fôrma” para a construção de objetos

Classes e objetos: reformulando

- Na orientação a objetos, os programas são organizados em classes e objetos
- As classes servem como fôrma para criar os objetos
- Os objetos agrupam os dados e as funções que operam sobre esses dados (metáfora: pense que está tudo sendo colocado numa “cápsula”)
- São criados vários objetos, que interagem entre si (metáfora: pense que os objetos estão “trocando mensagens” entre si).

Exemplo de classe e objetos

Classe Carro
- marca - modelo - ano - cor - combustivel - velocidade
+ acelerar () + freiar () + abastecer (x litros) + sobe_marcha () + desce_marcha ()

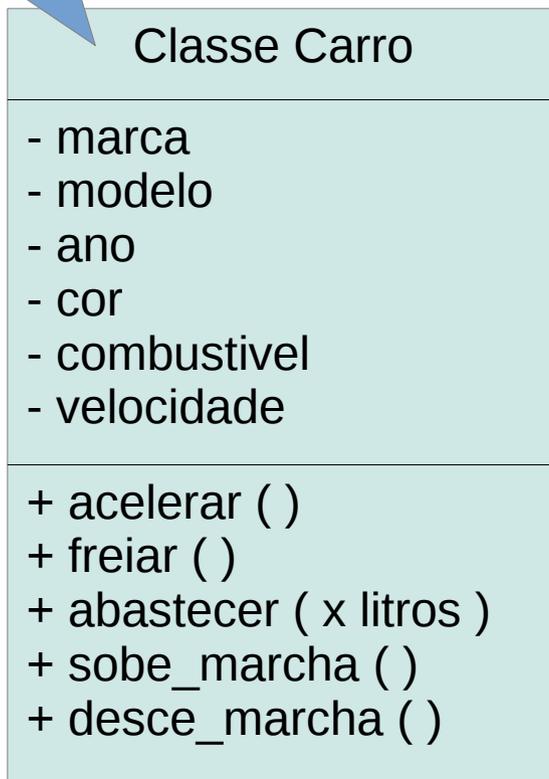
<u>Carro: carro1</u>
- marca: "Ford" - modelo: "Mustang GT Shelby" - ano: 2012 - cor: "vermelho" - combustível: 20 - velocidade: 180

<u>Carro: carro2</u>
- marca: "Chevrolet" - modelo: "Camaro SS" - ano: 2012 - cor: "amarelo" - combustível: 23 - velocidade: 164

UML

- No slide anterior, já começamos a utilizar a notação da UML (Unified Modeling Language)
- Especificamente, usamos um “diagrama de classes” e um “diagrama de objetos”
- Há muitos diagramas diferentes na UML, mas esses dois serão suficientes em nosso curso.

DIAGRAMA DE CLASSES



UML

DIAGRAMA DE OBJETOS



UML

