Programação Orientada a Objetos

Aula I – Conceitos de Iniciais de Programação Orientada a Objetos

Prof.: Bruno E. G. Gomes
IFRN

Introdução

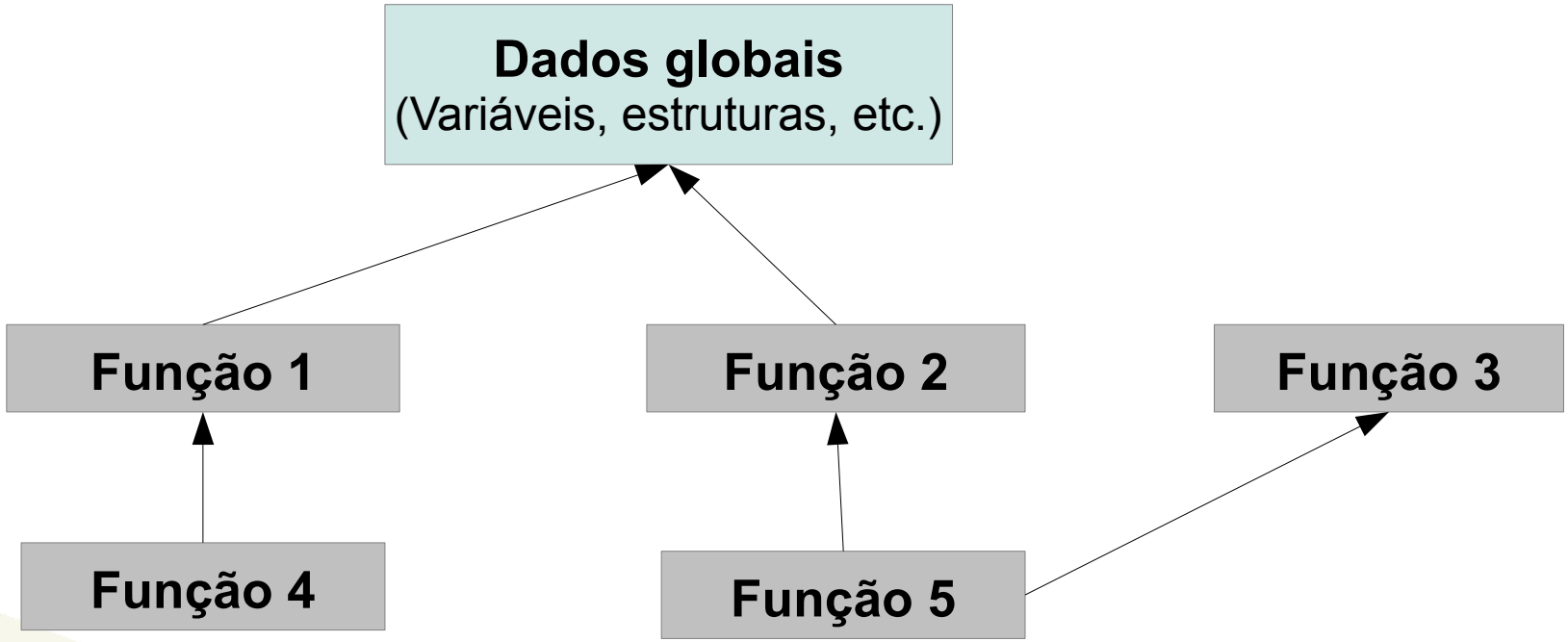
- Na aula de hoje:
  - Revisão de Programação Imperativa
  - Introdução à Programação Orientada a Objetos
  - Conceitos de objetos e classes
  - Relação de classes e objetos com o "mundo real"

- Objetivos:
  - Compreensão dos conceitos iniciais da POO
  - Criação de uma pequena classe e de um objeto dessa classe

Programação Imperativa

- Modelo de programação que utilizamos até o momento
- Em programação imperativa temos:
  - Dados do programa (variáveis)
  - Ações que operam sobre esses dados (funções)
- Normalmente as funções são reunidas em uma ou mais bibliotecas

Programação Imperativa



- Exemplo: mat.cpp, mat.h (biblioteca)

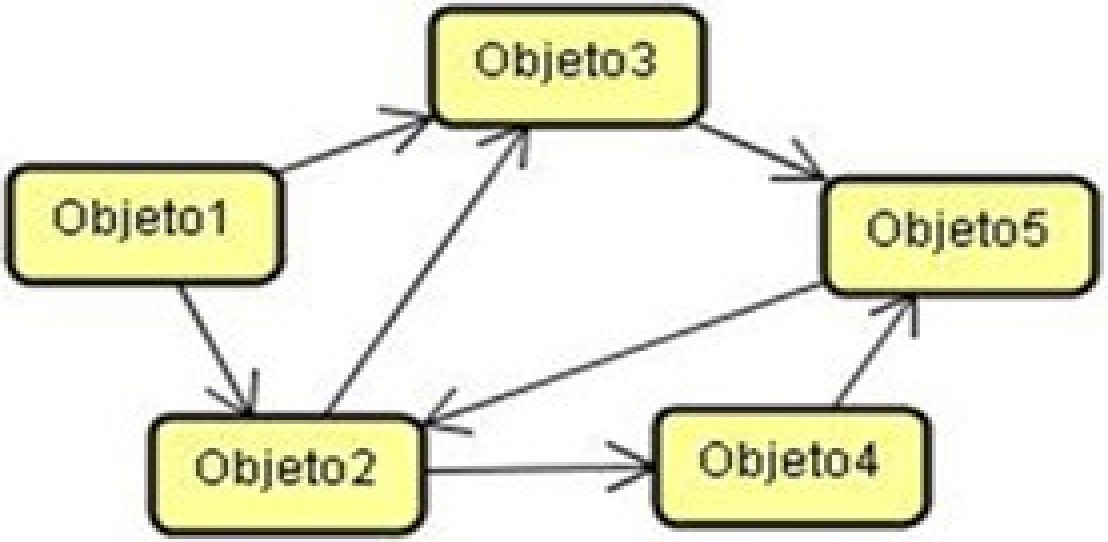
Programação Imperativa

- Dados e funções são independentes
- Dados são fornecidos a funções, que normalmente retornam um resultado
- Há pouco (ou nenhum) controle ao acesso a variáveis e funções
- Ainda hoje esse modelo de programação é utilizado
  - É também base para o modelo de orientação a objetos

Programação Orientada a Objetos

- Forma de pensar em um programa a partir de abstrações da vida real, tais como:
  - Objetos
  - Troca de mensagens
  - Composição
  - Hierarquia, etc.

- O Programa é composto por um conjunto de objetos que se comunicam entre si através da troca de mensagens entre eles



- Dados e funções:
  - São definidos e encontram-se interligados dentro de um objeto
  - podem (e devem) ser protegidos de acesso indevido (conceito de *encapsulamento*)

POO – Princípios básicos

- Abstração
- Classes e Objetos
  - São as entidades básicas de um sistema OO
  - Objetos são criados a partir de classes
- Outros conceitos:
  - Encapsulamento
  - Composição
  - Herança
  - Polimorfismo

Abstração

- Consiste em uma forma de pensar na solução de um problema de modo a gerenciar a sua complexidade

- A partir da descrição de um problema, deve-se focar e “modelar” apenas os aspectos essenciais

- Exemplos no mundo real:
  - Um médico especialista em uma determinada área
  - Um engenheiro que projeta um edifício pesando em cada parte isoladamente (sistema elétrico, hidráulico, etc...).

Objetos

- Representam objetos do mundo real ou computacional
  - Aluno, venda, biblioteca, calculadora, carro, veículo, etc.
- Conjunto de elementos que fazem parte da solução que estamos desenvolvendo
  - Loja de veículos: objetos vendedor, carro, moto, venda, etc.
  - Biblioteca: livro, exemplar, empréstimo, funcionário, etc.
- Possuem *características* e *comportamento*.

Objetos

- Característica (**atributos**, em OO)
  - Ex.: Atributos de “Pessoa”: nome, cpf, identidade, idade, estado civil...
  - Atributos de “Carro”: cor, motor, ano, marca, etc.

- Comportamento (**métodos**, em OO)
  - Ações que podem ser realizadas pelo objeto
  - Comportamentos de um “Carro”: ligar, acelerar, diminuir a velocidade, parar, etc.

Exercício

- Escolha um objeto a sua volta e cite alguns de seus atributos e comportamentos

Classes

- Pode ser vista como um molde ou projeto para se criar "objetos" daquela classe
- Os objetos são representações concretas (instâncias) das classes
- Uma classe criada *define um novo "tipo" de dados.*

“Classes” no mundo real

- No mundo real temos vários exemplos de projetos, por exemplo:
  - Uma planta de engenharia,
  - Um desenho,
  - Uma maquete de um prédio, etc.

- No entanto, uma classe pode ser a representação para criar qualquer tipo de objetos
  - Uma pessoa, um carro, uma caneta, um livro, uma loja, uma nota fiscal, um item de compra, etc.

Exemplos

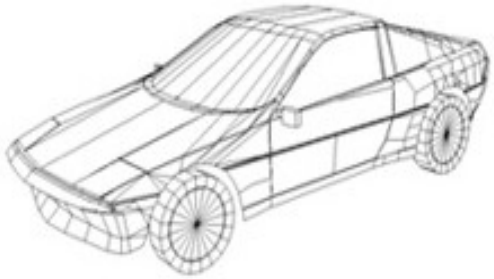
Uma **planta** é um modelo a ser seguido para a criação de uma casa.

Exemplos

- Outra representação de uma casa

Classe carro e objetos do tipo carro

CLASSE →



Tipo: ?

Cor: ?

Placa: ?

Número de Portas: ?

Tipo: Porsche

Cor: Cinza

Placa: MHZ-4345

Número de Portas: 2

← OBJETOS →

Tipo: Ferrari

Cor: Vermelho

Placa: JKL-0001

Número de Portas: 4

- Considerando o exemplo do slide anterior, para as classes **Livro**, **Animal** e **Cachorro**, descreva os atributos e comportamentos das classes e dos objetos correspondentes.

Representação de uma classe (em UM

Exemplo de classe em C++

```
class Bicicleta {
    private:
        float velocidade;
        int marcha;
        int cor;
    public:
        void pedalar(float vel) {
            if (vel > 0 && vel <= 50) {
                velocidade = vel;
            }
        }
        void trocarMarcha(int m) {
            if (m >= 1 && m <= 21) {  marcha = m;  }
        }
};
```