UFMG
UNIVERSIDADE FEDERAL DE MINAS GERAIS

Programação e Desenvolvimento de Software 2

Introdução à orientação a objetos

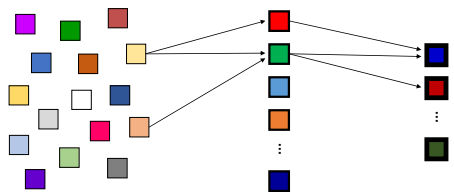
Prof. Douglas G. Macharet
douglas.macharet@dcc.ufmg.br

DCC
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Introdução

- Paradigmas de programação
  - Visão que o programador possui sobre a estruturação e execução do programa
  - Diferentes linguagens → Diferentes paradigmas
- Tipos
  - Imperativo, Estruturado, Procedural
  - Funcional
  - Lógico
  - Orientado a objetos ⬅

Introdução

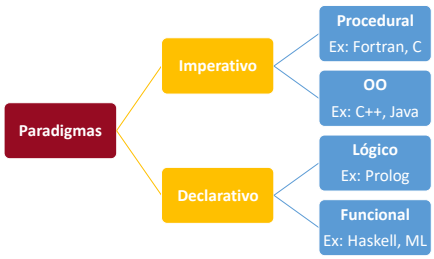


Cada linguagem realiza um ou mais paradigmas.

Cada paradigma consiste em um conjunto de conceitos.

Linguagens ➡ **Paradigmas** ➡ **Conceitos**

Introdução

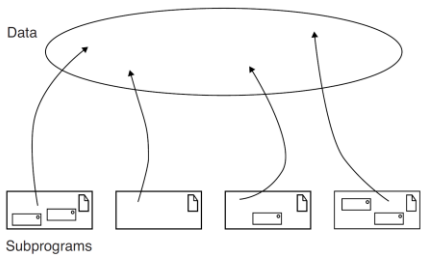


Introdução

- Programação Estruturada
  - Instruções que mudam o estado do programa
  - Programas imperativos (ações)
- Programação Orientada a Objetos
  - Dados e procedimentos encapsulados (TADs)
  - Composto por diversos objetos
  - Interação/comunicação entre os objetos

Introdução

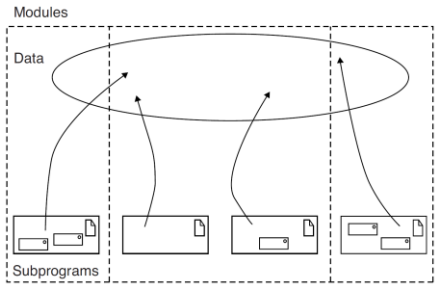
Programação Estruturada/Procedural



Introdução
Programação Estruturada/Procedural

- Como resolver problemas muito grandes?
  - Construí-lo a partir de partes menores
  - "Dividir para conquistar"
- Módulos compiláveis
  - Solucionam uma parte do problema
  - Dados x Manipulação
    - Abstração fraca para problemas mais complexos

Introdução
Programação Estruturada/Procedural



Introdução
Crise

- Prazos raramente cumpridos
- Custos acima dos previstos
- Não atendimento dos requisitos
- Elevado custo de manutenção

https://en.wikipedia.org/wiki/Software_crisis
https://www.youtube.com/watch?v=0b5vp4Z2PKE
https://www.youtube.com/watch?v=Cd3TrUK8axU

Programação Orientada a Objetos

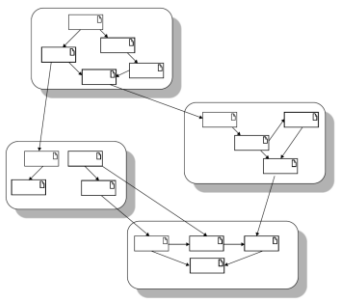
- Sistemas maiores e mais complexos
  - Aumentar a produtividade no desenvolvimento
  - Diminuir a chance de problemas
  - Facilitar a manutenção/extensão
- Programação Orientada a Objetos
  - Tem apresentado bons resultados
  - Não é uma bala de prata!

https://en.wikipedia.org/wiki/Object-oriented_programming#Criticism

Programação Orientada a Objetos
História

- Desenvolvimento de hardware
  - Pedaços simples de hardware (chips) unidos para se montar um hardware mais complexo
- Amadurecimento dos conceitos
  - Simula (60's)
  - Smalltalk (70's)
  - C++ (80's)

Programação Orientada a Objetos



Programação Orientada a Objetos
PE vs. POO

- Programação Estruturada
  - Procedimentos implementados em blocos
  - Comunicação pela passagem de dados
  - Execução → Acionamento de procedimentos
- Programação Orientada a Objetos
  - Dados e procedimentos encapsulados
  - Execução → Comunicação entre objetos

Programação Orientada a Objetos
Novo paradigma de programação

- Programação Estruturada
  - Dados acessados via funções
  - Representação de tipos complexos com struct
- Programação Orientada a Objetos
  - Dados são dotados de certa inteligência
  - Sabem realizar operações sobre si mesmos
  - É preciso conhecer a implementação?

O que o TAD representa e faz é mais importante do que como ele faz!

Programação Orientada a Objetos
"Humanização" dos dados

- O dado tem características humanas
  - Conhecimento/ação sobre si próprio!
- Exemplos
  - Uma *circunferência* sabe determinar sua área
  - Uma *lista* sabe dizer quantos elementos tem
  - Um *estudante* sabe se matricular em um *curso*
- Programação Estruturada
  - Funções externas fazem isso pelas estruturas

Programação Orientada a Objetos
Benefícios

- Maior confiabilidade
- Maior reaproveitamento de código
- Facilidade de manutenção
- Melhor gerenciamento
- Maior robustez
- ...

Classes vs. Objetos

- Classe
  - Representa uma unidade de compilação
  - Conceito, ideia, abstração (representação)
  - Um módulo, um tipo → Struct turbinado!
- Um TAD é implementado como uma Classe
  - TAD → Classe → Objeto
- Objeto é uma instância de uma Classe
  - Existe em tempo de execução (memória)

Classes vs. Objetos

Memória	Classes	Objetos
Massa	Fôrma	Biscoitos

Objetos

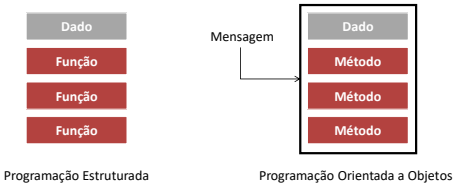
- Propriedades
  - Identidade (referência única)
  - Estado
  - Comportamento
- Possui alocação própria na memória
- Deve ser criado explicitamente
- Em algum momento, será destruído
  - Explicitamente x Implicitamente

Objetos

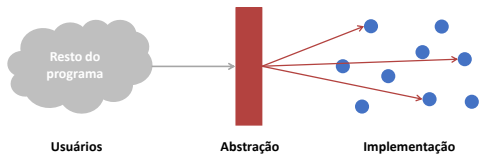
- Identidade
  - Propriedade que o difere dos demais objetos
- Estado
  - Valores que suas propriedades estão assumindo
- Comportamento
  - Relacionamento com os demais objetos
  - Como trata mudanças de estado

Objetos

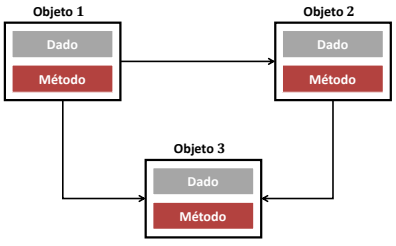
- Dados ocultos do "mundo externo"
- Acessíveis somente via métodos internos



Objetos



Objetos

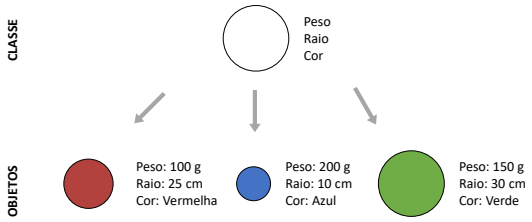


Classes vs. Objetos

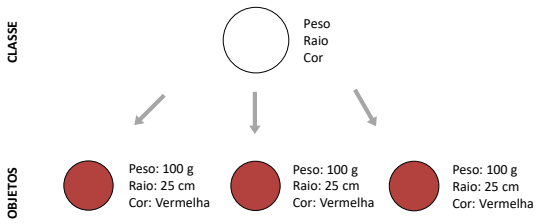
- Classe
  - Descrição de propriedades em comum de um grupo de objetos (conjunto)
  - Um conceito
  - Faz parte de um programa
  - Exemplo: Pessoa
  - Exemplo: Carro
- Objeto
  - Representação das propriedades de uma única instância (elemento)
  - Um fenômeno (ocorrência)
  - Faz parte de uma execução
  - Exemplo: João da Silva
  - Exemplo: Ferrari

Classes vs. Objetos

Classes vs. Objetos



Classes vs. Objetos

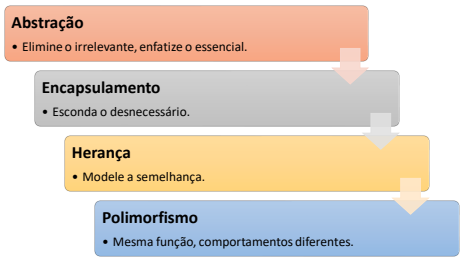


Programação Orientada a Objetos
Princípios

- Abstração
- Encapsulamento
- Herança
- Polimorfismo
- Modularidade
- Mensagens

Princípios fundamentais

Programação Orientada a Objetos
Princípios



Programação Orientada a Objetos
Princípios – Abstração

- Modelagem de um domínio
  - Identificar artefatos de software
  - Ignorar aspectos não relevantes
  - Representação de detalhes relevantes do domínio do problema na linguagem de solução
- Classes são abstrações de conceitos

Programação Orientada a Objetos
Princípios – Encapsulamento

- Agrupamento dos dados e procedimentos correlacionados em uma mesma entidade
- Um sistema orientado a objetos baseia-se no contrato e não na implementação interna
- Proteção da estrutura interna (integridade)

Programação Orientada a Objetos
Princípios – Herança

- Permite a hierarquização das classes
- Classe especializada (subclasse, filha)
  - Herda as propriedades (atributos e métodos)
  - Pode sobrescrever/estender o comportamento
- Auxilia no reuso de código

Programação Orientada a Objetos
Princípios – Polimorfismo

- Tratar tipos diferentes de forma homogênea
  - Classes distintas com métodos homônimos
  - Diferentes níveis na mesma hierarquia
- Um método assume "diferentes formas"
  - Apresenta diferentes comportamentos

Programação Orientada a Objetos
Princípios – Modularidade

- Separação em conjuntos de módulos
  - Classes com independência de funcionamento
- Separação de responsabilidades
  - Limites lógicos entre os componentes
  - Melhora a manutenibilidade
  - Aumenta a estabilidade (↓ efeitos colaterais)

Programação Orientada a Objetos
Princípios – Mensagens

- Comunicação entre objetos
  - Envio/recebimento de mensagens
  - Forma de invocar um comportamento
- Informação contida na mensagem
  - Utiliza o contrato firmado entre as partes