

Aula 3

Prof. Daniel Cavalcanti Jeronymo

Fundamentos de Programação

ET43I

Orientada a Objetos

Abstração em computação,
paradigmas de linguagens de
programação. Conceitos de
orientação a objetos.

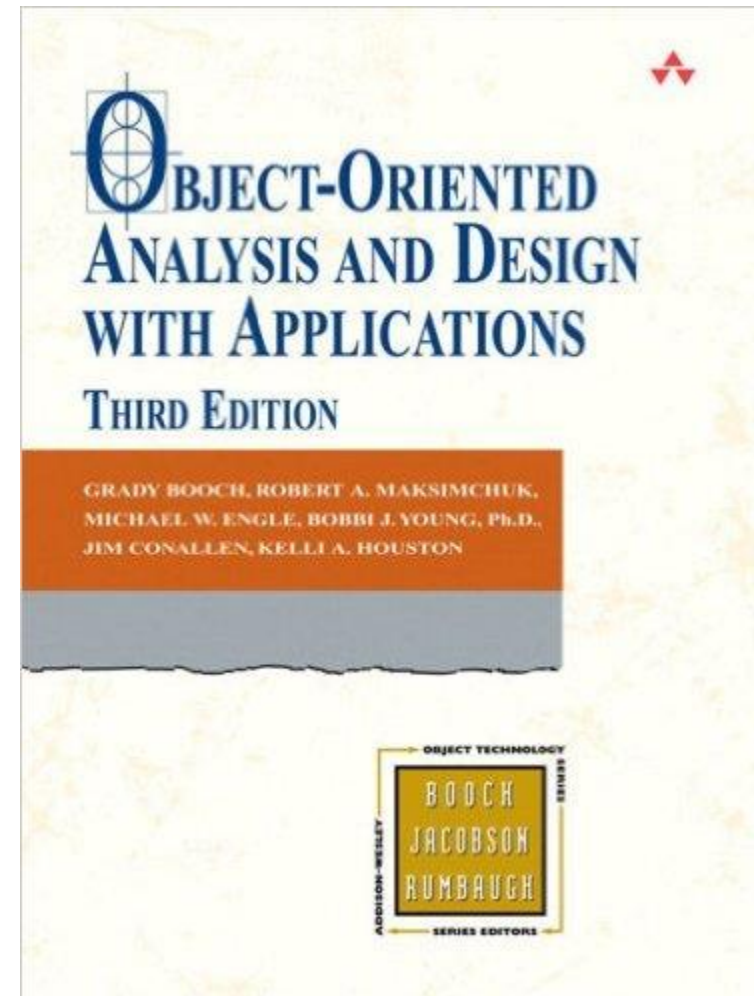
Universidade Tecnológica Federal do Paraná (UTFPR)
Engenharia Eletrônica – 3º Período

- Abstração
- Paradigmas de linguagens de programação
- Conceitos de orientação a objetos

Abstração

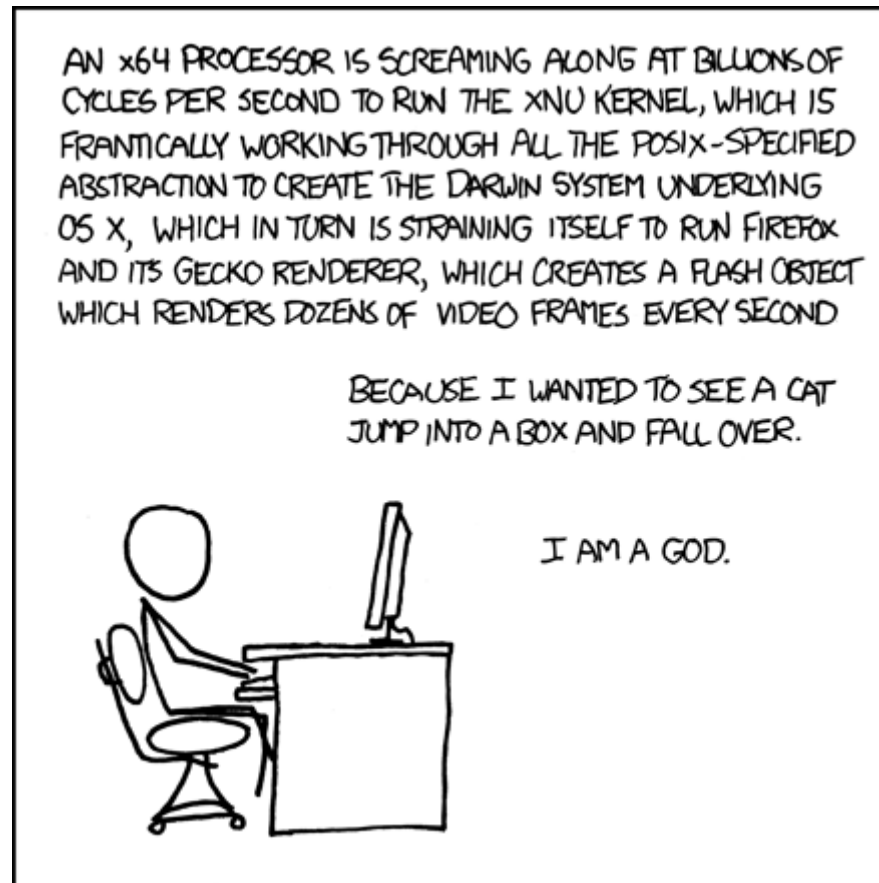
- “Uma abstração indica as características essenciais de um objeto que o distingue de todos os outros tipos de objetos e então proporciona fronteiras conceituais nitidamente definidas, relativas a perspectiva do espectador”

– G. Booch



Abstração

- Foco no **essencial** e ignora-se o **irrelevante**



Abstração

- Vamos pensar numa **MESA**



Abstração

- Não foi especificado o tipo da mesa



- *Madeira? Vidro? 4 pernas? 3 pernas? Pequena? Grande? Coluna central? Pernas distribuídas nos cantos?*
- Mesmo assim todos entendem o conceito e conseguem visualizar uma

Abstração

- Abstração de uma mesa
- **Modelo abstrato:**
 - Um objeto apoiado no chão
 - Exibe uma superfície plana para apoio de outros objetos
- O conceito é separado e independente de qualquer exemplo específico

Abstração

- O **essencial** e o **irrelevante** dependem de contexto
 - Logo, a **abstração depende de contexto**
- Abstração de uma mesa **em gravidade zero**?



Abstração

Conceito amplamente utilizado para gerenciar complexidade

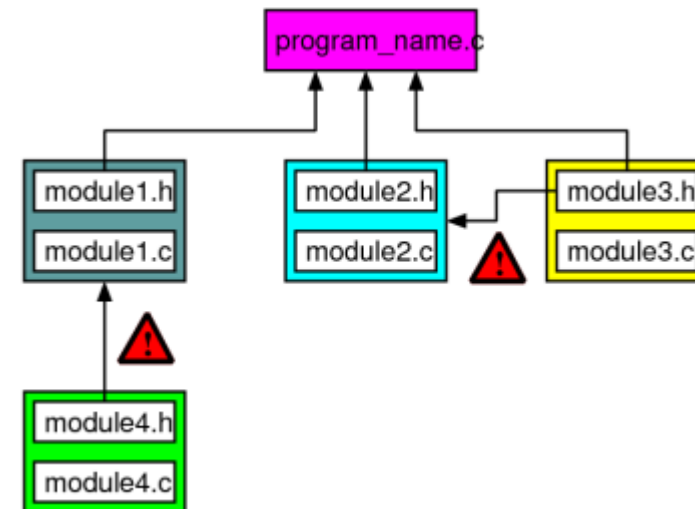
- Sistemas complexos são decompostos em componentes menores



Abstração

Cada componente deve ter as seguintes características bem definidas:

- Comportamento (**faz** A)
- Responsabilidade (faz **apenas** A e **apenas ele modifica** B)
- Comunicação (para fazer A ou modificar B, deve ser **acessado** pela *interface* C)
- Estado interno (B)
- *Interface* (C)



Interface: ponto onde dois sistemas interagem

Tipos Abstratos de Dados (*Abstract Data Types*)

- Modelos para tipos de dados definidos pelos seus comportamentos

- Exemplo:

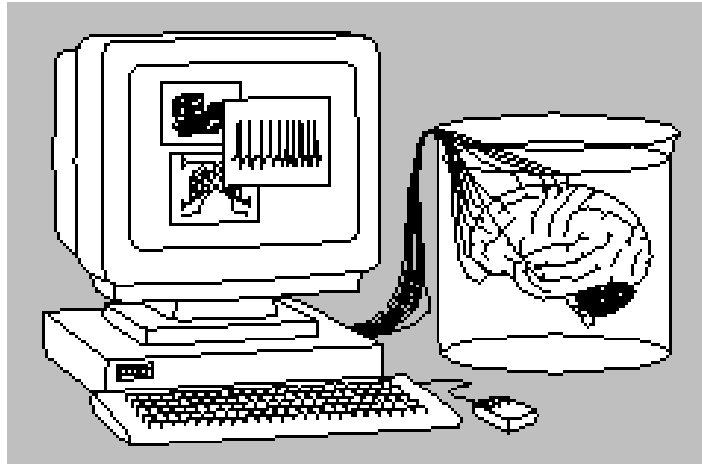
```
int A;  
A = 10;  
A += 21;
```

Sabemos o que faz (semântica)
mas não como é implementado

- Não somente primitivas mas também tipos complexos: lista, pilha, árvore, fila, etc.

Abstração

- Permite obter um modelo computacional

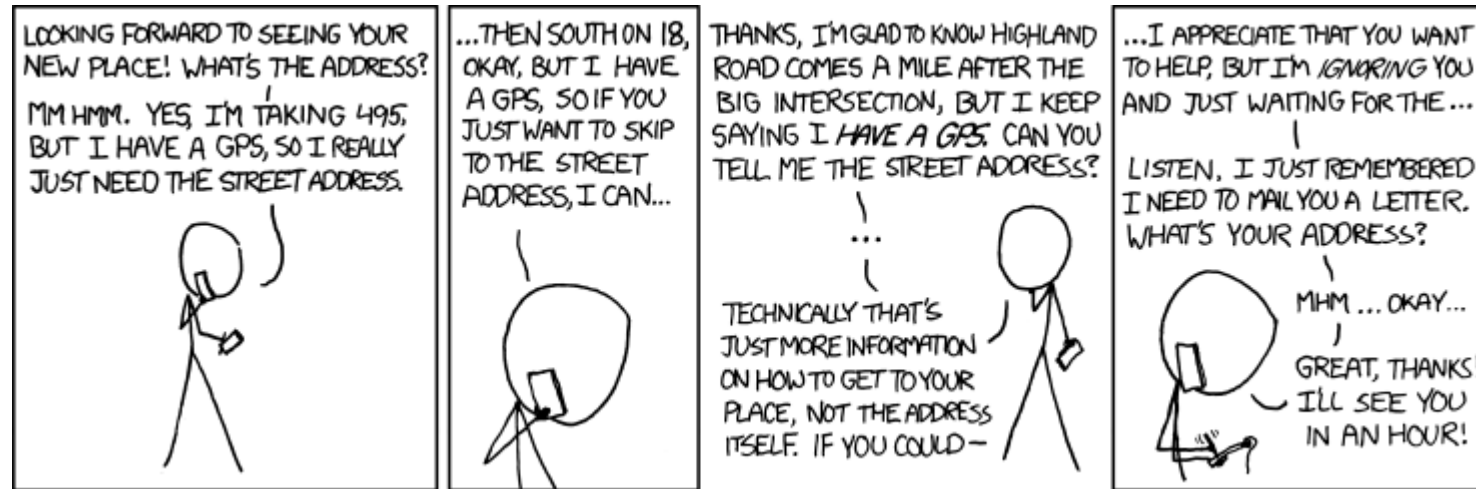
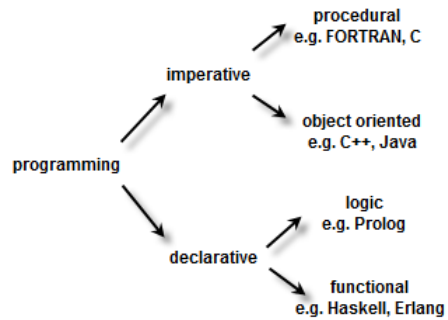


- **Modelo**: descrição de um sistema usando conceitos e linguagem de uma solução
- A solução computacional depende do **paradigma de linguagem de programação**

Paradigmas de Programação

- Paradigmas principais:

- Declarativa (*what*) x Imperativa (*how*)



- Não-estruturada x Estruturada

The Evolution of Expression

A structured while loop is easier to read than the convoluted goto approach.

Instead of writing ...

```

i = 1
START: if i = 4
  then goto END
  print(i)
  i = i + 1
  goto START
END:
  
```

we write ...

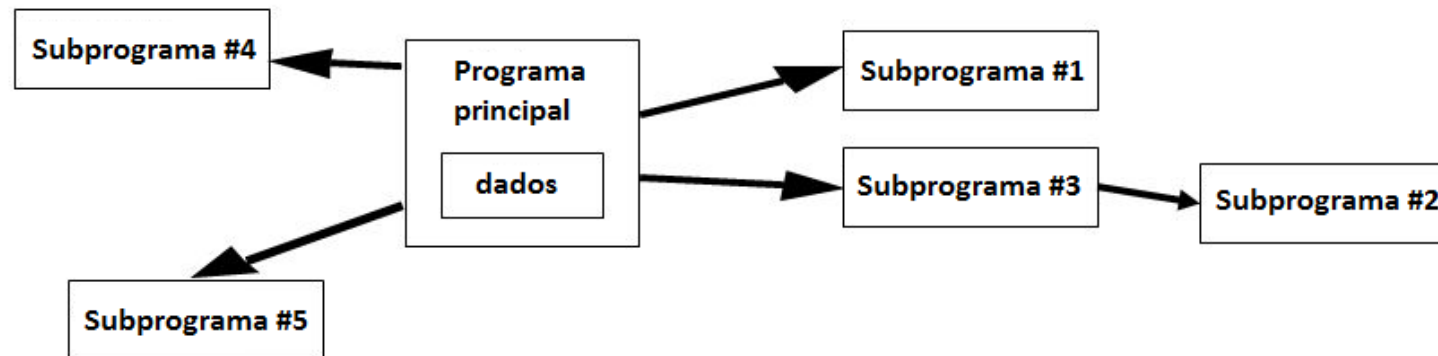
```

i = 1;
while (i < 4) {
  print(i);
  i = i + 1;
}
  
```

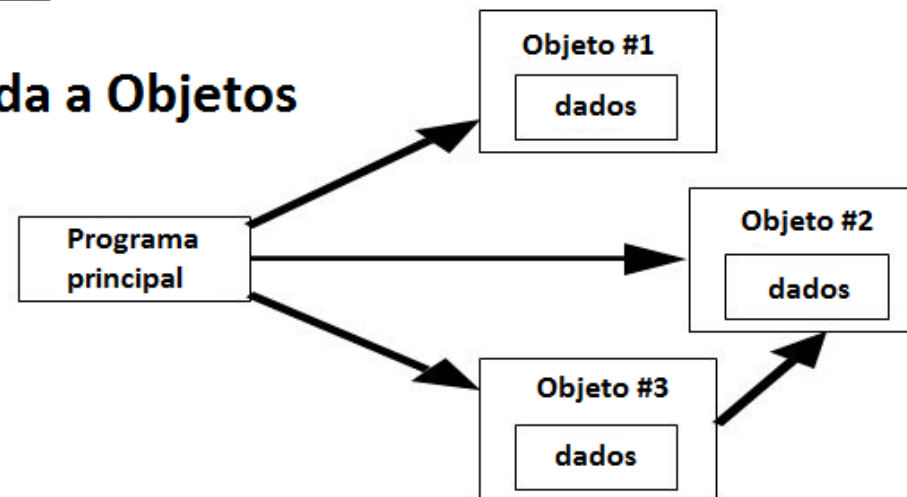
Paradigmas de Programação

- Paradigmas principais:
 - Procedural x Orientada a Objetos

Procedural



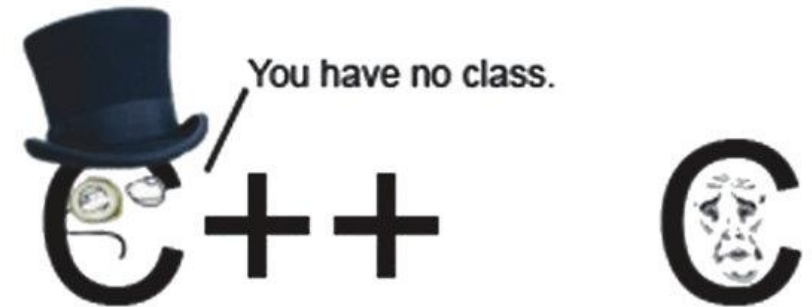
Orientada a Objetos



Paradigmas de Programação

- **Procedural**

- O problema é separado em variáveis, estruturas de dados e subrotinas
- As subrotinas **transformam** dados



- **Orientada a Objetos**

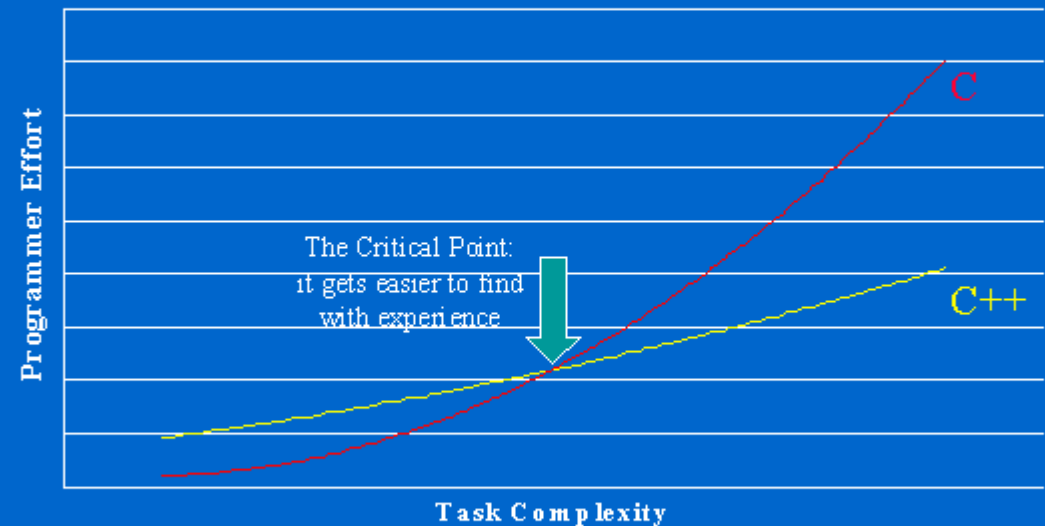
- O problema é modelado utilizando **objetos** que exibem comportamento (métodos) e dados (atributos).
- Os métodos combinam subrotinas e dados, transformando a si mesmos.

Paradigmas de Programação

- Procedural x Orientada a Objetos – **Quando Usar?**



C vs. C++: When To Use Them



The location of the Critical Point depends on many factors such as number of programmers, lifetime of the project, complexity of the problem, etc.

Conceitos OOP

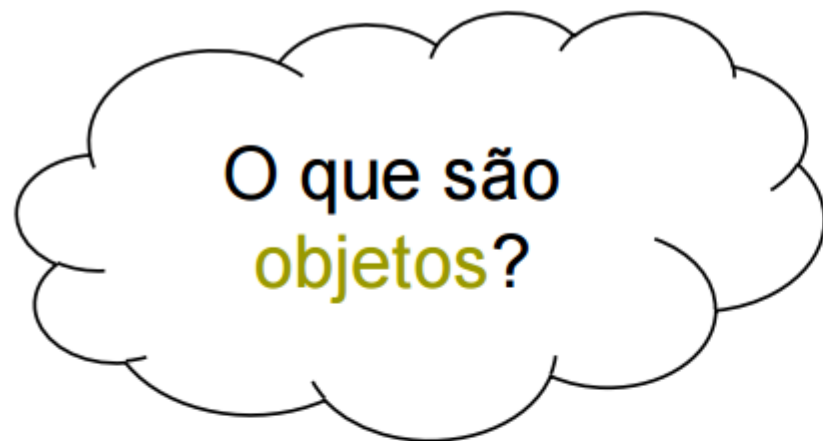
- **Evolução do processo de abstração nas linguagens de programação**
- Assembly – abstração da linguagem de máquina
- Linguagens procedurais (C, FORTRAN, PASCAL, etc) – abstrações da linguagem Assembly
 - ❑ Nas linguagens procedurais a solução de um problema é modelada como uma **série de operações que transformam dados**

Conceitos OOP

- **Evolução do processo de abstração nas linguagens de programação**
- Linguagens orientadas a objetos (C++, Java, C#, etc) – abstrações **do problema**
 - ❑ Nas linguagens OO a solução de um problema é modelada em **termos do próprio problema**
 - ❑ **Objetos** modelam pequenas partes do problema: possuem **estado e operações**



Conceitos OOP



Conceitos OOP

- **Structs**

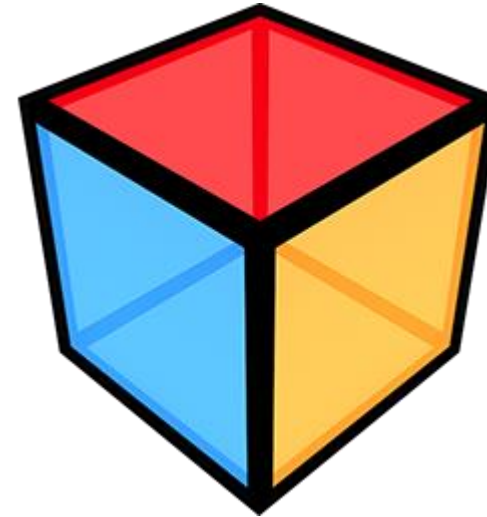
- Modelam propriedades pelo armazenamento de dados
- Como modelar um carro por estrutura?

- **Objetos**

- Mais completos que **structs**
- Modelam comportamentos
- Armazenam atributos (dados)
- Possuem métodos (funções)
- Como modelar um carro por objeto?

Objetos

- São variáveis
- Possuem um tipo – sua **classe**
- Comunicam-se por **mensagens** (chamadas de funções)
- Objeto cubo – atributos e métodos?



Objetos

❑ Classe: `Cubo`

❑ Atributos:

```
enum Cor lado_esquerdo;
```

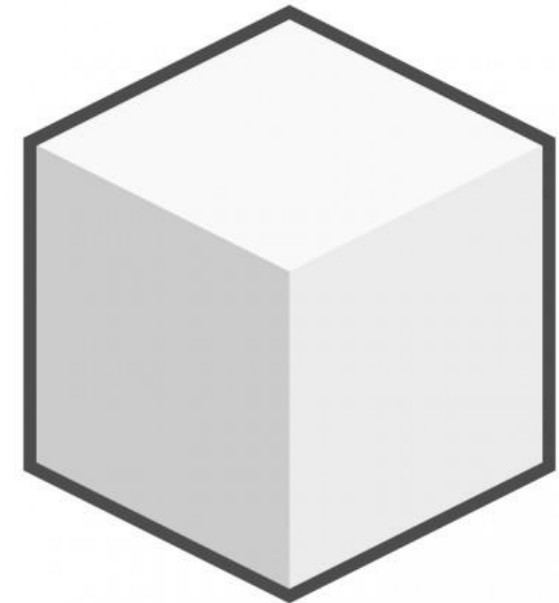
```
enum Cor lado_direito;
```

```
enum Cor lado_topo;
```

❑ Métodos:

```
girar_esquerda();
```

```
girar_direita();
```



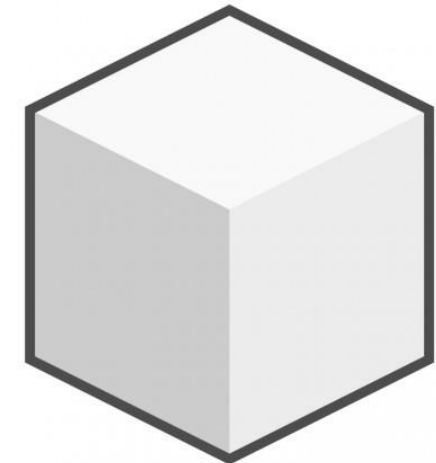
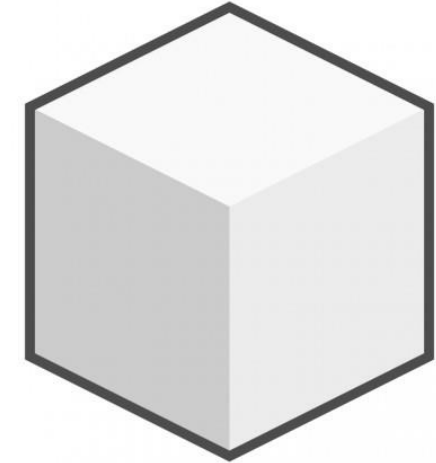
Conceitos OOP

Objetos

❑ Instanciando a classe Cubo obtemos os objetos:

```
meu_cubo1 = Cubo ();
```

```
meu_cubo2 = Cubo ();
```



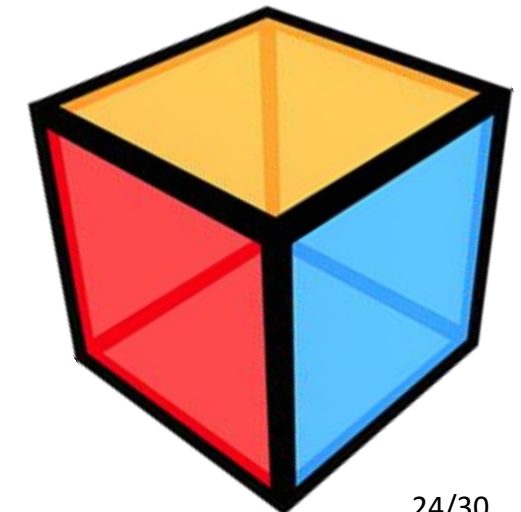
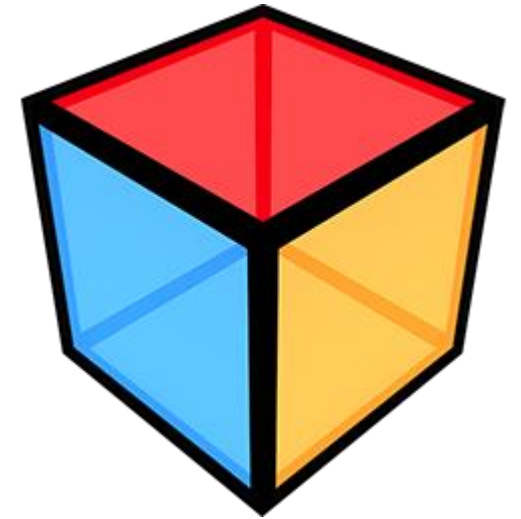
Conceitos OOP

Objetos

❑ Cada objeto é distinto:

```
meu_cubo1.lado_esquerdo = azul;  
meu_cubo1.lado_direito = laranja;  
meu_cubo1.lado_topo = vermelho;
```

```
meu_cubo2.lado_esquerdo = vermelho;  
meu_cubo2.lado_direito = azul;  
meu_cubo2.lado_topo = laranja;
```

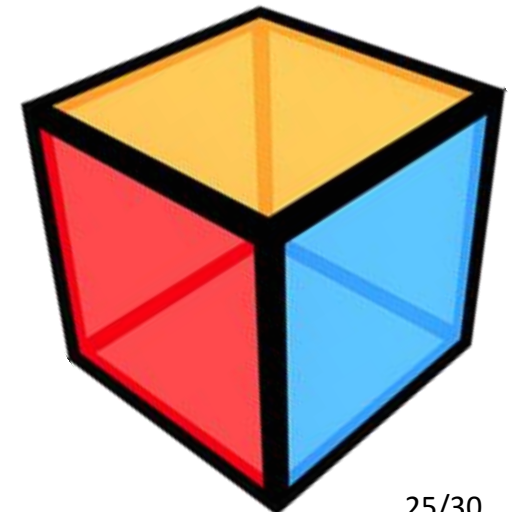
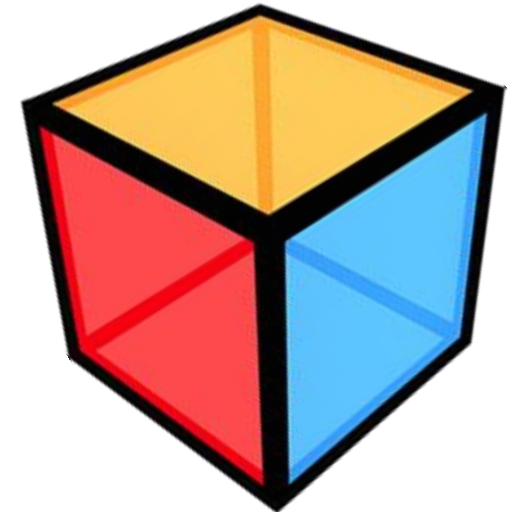


Conceitos OOP

Objetos

- ❑ Métodos alteram apenas aquele objeto:

```
meu_cubo1.girar_esquerda();
```



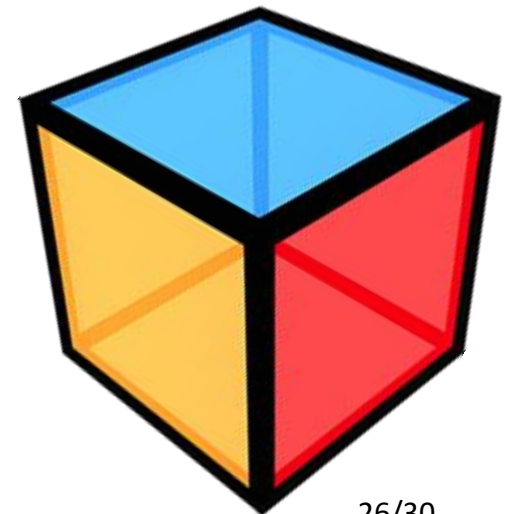
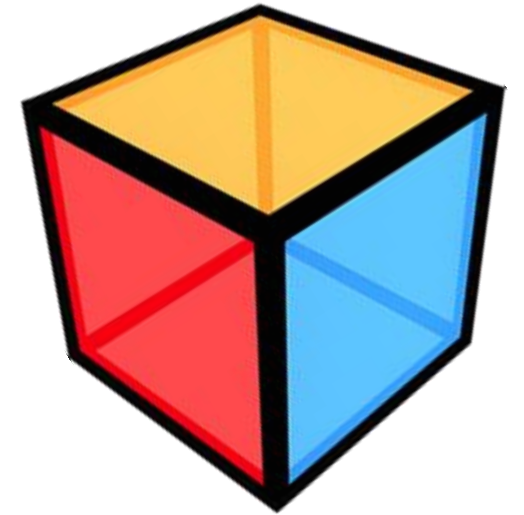
Conceitos OOP

Objetos

- ❑ Métodos alteram apenas aquele objeto:

```
meu_cubo1.girar_esquerda();
```

```
meu_cubo2.girar_esquerda();
```



Objetos

Objeto Lâmpada

Características



Potência: 40w

Tensão: 12V

Família: incandescentes

Cor: clara

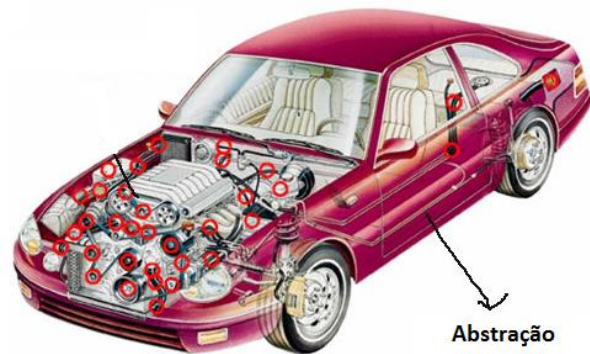
Comportamento

ligar, desligar, aumentaLuminosidade, diminuiLuminosidade, etc.

Conceitos OOP

- Conceitos principais

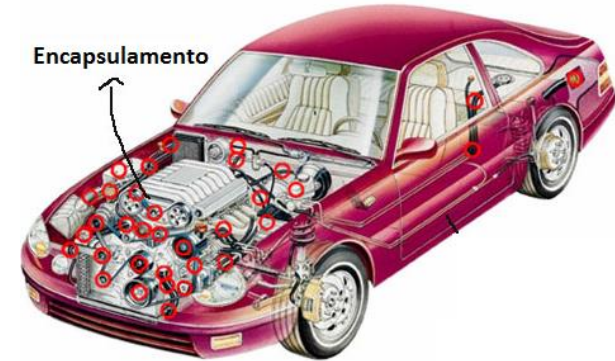
Abstração



Herança



Encapsulamento



Polimorfismo



Conceitos OOP

- Orientação a objetos permite a aplicação de diversas regras
- **CUIDADO!**
- **Teorema Fundamental da Engenharia de Software:**
“Todo problema pode ser resolvido pela introdução de um nível extra de indireção”
- **Corolário de Henney** “... exceto o problema de muitos níveis de indireção”

SIMPLY EXPLAINED



Conceitos OOP

- Como realizar uma lista simplesmente encadeada em C?
- Usando OOP – É possível?

