

Revisão

Paradigmas de Linguagens

- Inicialmente
 - os computadores eram programados através de código binário
 - representava a sequência de controles alimentadas à CPU
 - Tal processo era difícil e propenso a erros
 - um paradigma de programação de muito baixo nível

Bug

MOV AL 60h

```
010110100001011000101000000100101
010010101000000101001001010001011
0001010000101010001001000010100001
0010100011000100000010100001010010
100001101000101010101100110001011
101000001010000010010001001010101
100100001010110000100110010111001
001011010010010010010100101010101
101001010000101000100101010101010
110001001010000001010111010010101
100010100001001010110001010010100
0100010110000101110101010000101010
100010110101001110001100101010101
```



FIGURA 2 - Representação esquemática da figura 1 na linguagem dos computadores.

Paradigmas de Linguagens

- Para facilitar a programação foram criadas as **linguagens de montagem (Assembly)**
 - um paradigma da "segunda geração" das linguagens
 - substituíam as funções do código de máquina por mnemônicos, endereços de memória absolutos por identificadores
 - linguagem de montagem ainda é considerada de baixo nível

Paradigmas de Linguagens

- Por exemplo:
 - Um computador sabe o que a instrução-máquina IA-21 (10110000 01100001) faz
 - Para os programadores é mais fácil recordar a representação equivalente em instruções mnemônicas

MOV AL, 61h


- Tal instrução ordena que o valor hexadecimal 61 (97, em decimal) seja movido para o registrador 'AL'

Paradigmas de Linguagens

- O próximo avanço foi o desenvolvimento das **linguagens procedurais**
 - descritas como de alto nível, ou linguagens da terceira geração
 - descrevem, passo a passo, o procedimento a ser seguido para resolver certo problema
 - Por exemplo, COBOL, FORTRAN, ALGOL

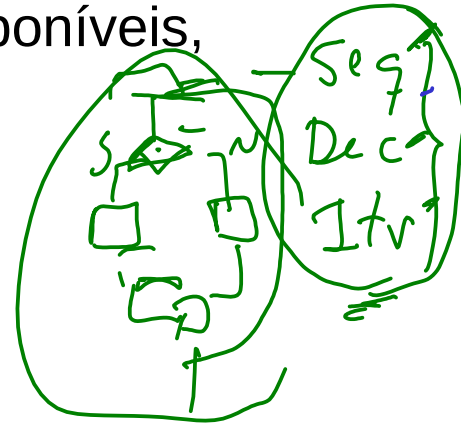
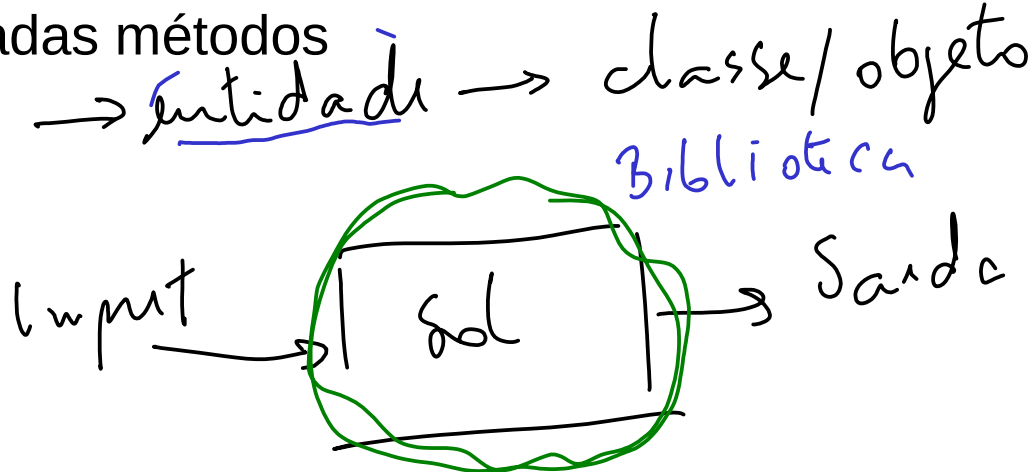
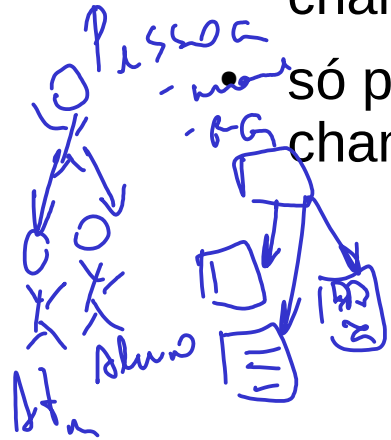
↳ Banco

Paradigmas de Linguagens

- Mais recente → **Programação Estruturada**
 - desenvolvida por Michael A. Jackson no seu livro "Principles of Program Design" de 1975
 - todos os programas possíveis podem ser reduzidos a apenas três estruturas:
 - Sequência, Decisão e Interação 
 - Por exemplo: Ada, Basic, C, Pascal, PHP e Perl

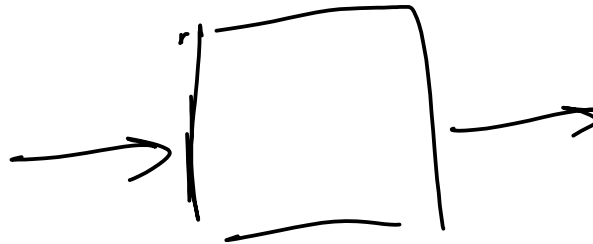
Paradigmas de Linguagens

- Posteriormente, **linguagens orientadas a objeto** foram criadas
 - os dados e as rotinas para manipulá-los são mantidos numa unidade chamada objeto
 - só pode acessar os dados através das subrotinas disponíveis, chamadas métodos



Paradigmas de Linguagens

- Independente do ramo das linguagens imperativas:
- **Programação orientada a eventos** \Rightarrow App Inventor
 - programas tradicionais que seguem um fluxo de controle padronizado
 - o controle de fluxo de programas orientados a evento são guiados por indicações externas, chamadas eventos



Paradigmas de Linguagens

- **Programação científica**
- Visa compreender fenômenos por meio da análise (automatizada) de dados → **Ciência de Dados**
- Envolve:
 - Método científico
 - Estatística / Matemática
 - Computação
- Precisa ser prática e eficiente → Big Data

Big Data

Paradigmas de Linguagens

- **Paradigmas de programação declarativa**

- Nessas linguagens se descreve o problema ao computador, não como resolvê-lo
- Dado um banco de dados ou um conjunto de regras, o computador tenta encontrar a solução ao casar todas as propriedades desejadas

- Lisp, Prolog

- Linguagens funcionais
- Linguagens lógicas

← Lisp

Prolog → Fatos
Regras

① → Prog Estructuradas ← Hoje

② → OO

③ → O. = Eventos

④ → Prog Client.

⑤ → Log / Funcional

