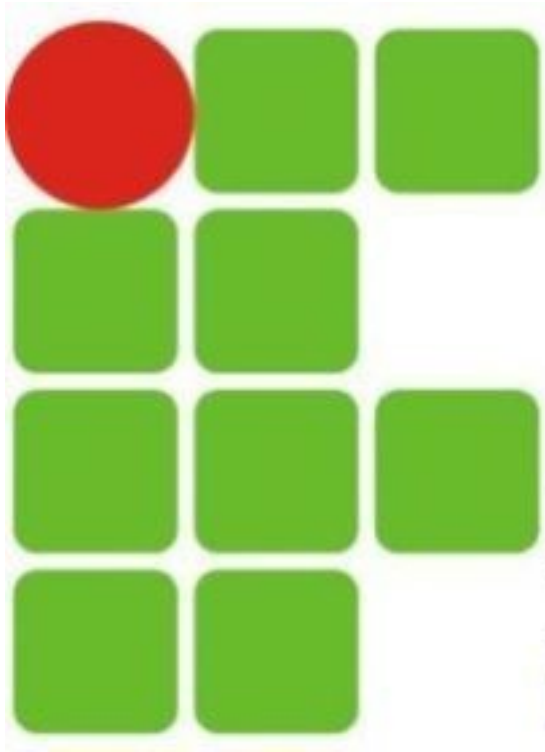


INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

INTRODUÇÃO

INTRODUÇÃO

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Computador cumpre ordens
- Como o computador entende nossa língua?
- O que a programação pode fornecer? (apps, sistemas, impressões, vídeo, etc)

INTRODUÇÃO
COMPUTADOR ENTENDE NOSSA LÍNGUA?

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

Comunicação entre pessoas através da linguagem: consiste num conjunto de **convenções** e **regras** para comunicação de ideias ou troca de informações

INTRODUÇÃO
COMPUTADOR ENTENDE NOSSA LÍNGUA?

- Meio de comunicação entre homem e máquina. Uma LP possui um conjunto de regras de sintaxe, semântica, instruções, operadores e identificadores

LINGUAGENS DE PROGRAMAÇÃO

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

Linguagem Natural

- Elementos envolvidos: homem X homem
- Conteúdo da comunicação palavras
- Sons, sinais
- Possui sintaxe e semântica

Linguagem de Programação

- Elementos envolvidos: homem X máquina
- Conteúdo da comunicação: programa
- Pulsos elétricos
- Possui sintaxe e semântica

LINGUAGENS DE PROGRAMAÇÃO CARACTERÍSTICAS

- Simplicidade: clareza e concisão semântica (linguagem com um mínimo número de conceitos e estruturas), e clareza sintática (sintaxe deve representar cada conceito de uma maneira apenas).
- Capacidade de abstração: representação de elementos;
- Expressividade: refere-se a facilidade com que um objeto pode ser representado;
- Ortogonalidade: um conjunto relativamente pequeno de construções primitivas podem ser combinadas em um número pequeno de maneiras para construir as estruturas de controle e de dados de uma linguagem.
- Suporte à manutenção e portabilidade: habilidade de manter programas que devem ser fáceis de entender e alterar;
- Eficiência: as medidas mais comuns são a eficiência da execução do programa, da tradução do programa, e da criação, teste e uso do programa.

INTRODUÇÃO
BENEFÍCIOS DA PROGRAMAÇÃO / COMPUTAÇÃO

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

Aplicativos

Sites

Serviços

NETFLIX

Conectividade

- Produção de sistemas
- Redução de custos
- Aumento da produtividade
- Redução de retrabalho
- Agilidade

INTRODUÇÃO

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

Programar

- Definir os comandos a serem executados pelo computador
- Arquivo de texto comum contendo a lógica do programa → código fonte

Linguagens de Programação

- Linguagens de programação servem como meio de comunicação entre os humanos e os computadores
- Tipos:
  - Linguagem de máquina
  - Baixo Nível
  - Alto Nível

LINGUAGENS DE PROGRAMAÇÃO
LINGUAGEM DE MÁQUINA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Programação diretamente em binário ou em hexadecimal
- Identificar a função dos códigos não era tarefa fácil
- Específico para cada tipo de computador
- Programar um microprocessador para executar uma determinada tarefa não era uma das coisas mais fáceis que existiam

```
00110001 00000000 00000000
00110001 00000001 00000001
00110011 00000001 00000010
01010001 00001011 00000010
00100010 00000010 00001000
01000011 00000001 00000000
01000001 00000001 00000001
00010000 00000010 00000000
01100010 00000000 00000000
```

LINGUAGENS DE PROGRAMAÇÃO
BAIXO NÍVEL OU LINGUAGEM DE MONTAGEM

- Logo foi criada a primeira linguagem de programação, a linguagem Assembly
  - Cada código que possui um significado especial para o microprocessador – Instrução
  - Muito mais fácil de se memorizar um nome do que um código
- Começa a ser introduzida a abstração
- Necessário uso de montador (Assembler → programa que lê o programa em baixo nível e converte os códigos mnemônicos para opcodes (códigos de operações)

```
mov ax,cs
mov ds,ax
mov ah,9
mov dx, offset Hello
int 21h
xor ax,ax
int 21h
```

LINGUAGENS DE PROGRAMAÇÃO
BAIXO NÍVEL OU LINGUAGEM DE MONTAGEM

Linguagem de Máquina

```
00110001 00000000 00000000
00110001 00000001 00000001
00110011 00000001 00000010
01010001 00001011 00000010
00100010 00000010 00001000
01000011 00000001 00000000
01000001 00000001 00000001
00010000 00000010 00000000
01100010 00000000 00000000
```

Assembly

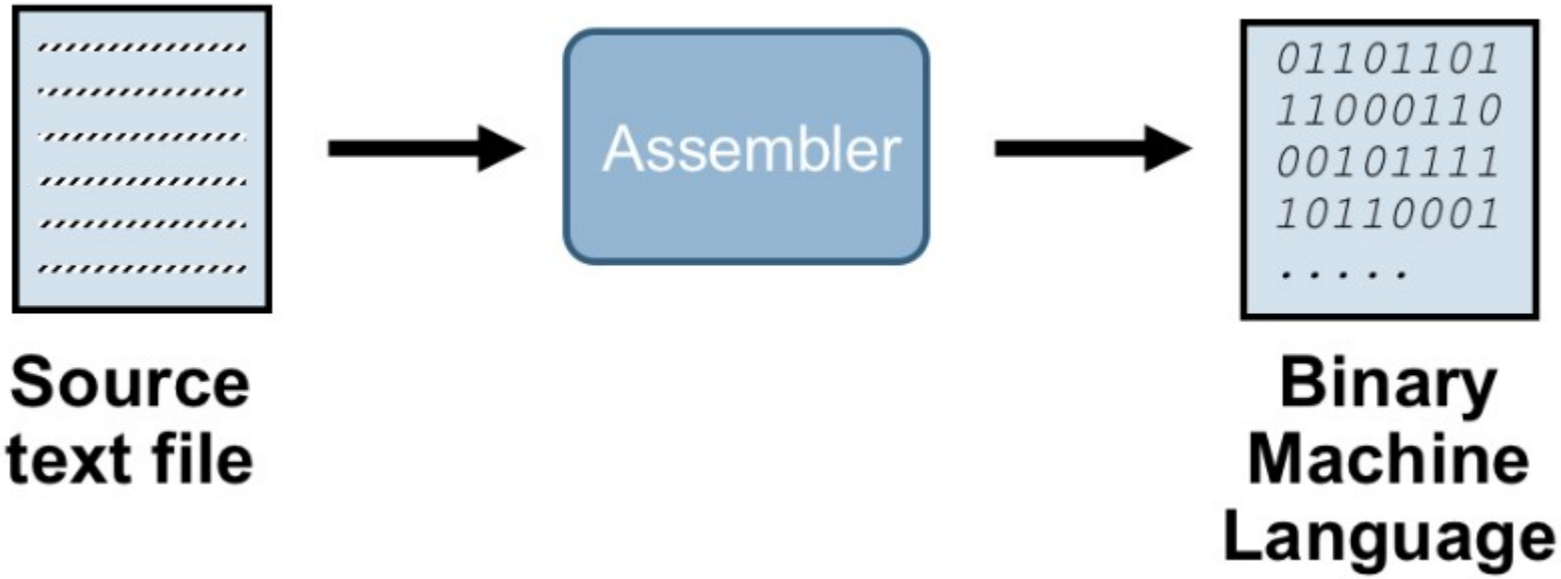
```
mov ax,cs
mov ds,ax
mov ah,9
mov dx, offset Hello
int 21h
xor ax,ax
int 21h
```

Mesmo sem saber o que significam, qual aparenta ser mais legível?

LINGUAGENS DE PROGRAMAÇÃO
BAIXO NÍVEL OU LINGUAGEM DE MONTAGEM

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Mesmo sendo o Assembly uma linguagem de baixo nível, o microprocessador não a entende
- Microprocessador entende somente linguagem de máquina
- O Assembler lê um programa lido em Assembly e converte-o para linguagem de máquina



LINGUAGENS DE PROGRAMAÇÃO
BAIXO NÍVEL OU LINGUAGEM DE MONTAGEM

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

Vantagens

- Execução mais rápida
- Programas menores

Desvantagens

- Dificuldade de compreensão do código
- Dificuldade de correção de erros
- Alto acoplamento com a arquitetura do microprocessador
- Programador deve conhecer o conjunto o funcionamento do microprocessador

LINGUAGENS DE PROGRAMAÇÃO ALTO NÍVEL

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Mais próxima da linguagem do homem
- Maior facilidade de compreensão
- Baixo acoplamento com o microprocessador → não requer conhecimento da arquitetura da máquina
- É portável
- Comandos executam tarefas mais completas
- Exemplos: BASIC, FORTRAN, Pascal, JAVA, PHP e C

LINGUAGENS DE PROGRAMAÇÃO

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Há diversas linguagens

PARADIGMAS DE PROGRAMAÇÃO

- Paradigma refere-se a um padrão de pensamento, modelo, classe de elementos com similaridades
- Um paradigma de programação fornece e determina a visão que o programador possui sobre a estruturação e execução do programa
- A maioria das linguagens de programação podem suportar mais de um paradigma
- Cabe ao programador escolher o paradigma mais adequado ao seu problema

PARADIGMAS DE PROGRAMAÇÃO
POR QUE ESTUDAR ESTE TEMA?

- Variedade de linguagens de programação
- Variedade de problemas
- Variedade de soluções
- Escolha da linguagem que melhor se adequa à necessidade
- Aumenta a eficiência da Programação: programador estará mais apto a usar a linguagem adequada ao problema
- Evitar o Vício de Programação de uma LP: não limitando as soluções às limitações de uma determinada linguagem
- Permite fazer mais com menor esforço
- Facilita o Aprendizado de uma nova LP: conhecimento de uma variedade de construções, técnicas e estruturas comuns
- Permite explorar ao máximo os recursos da LP

PARADIGMAS DE PROGRAMAÇÃO
POR QUE ESTUDAR ESTE TEMA?

- Estudando-se paradigmas de linguagens, é possível se tornar um programador mais eficiente, mais produtivo e que faz uso mais inteligente de LP, explorando ao máximo os recursos oferecidos pela LP.

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

PARADIGMA IMPERATIVO

PARADIGMAS DE PROGRAMAÇÃO
PARADIGMA IMPERATIVO

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Baseado na arquitetura de Von Neumann
- Primeiro paradigma a existir e até hoje é o dominante
- Descreve a computação como ações, enunciados ou comandos que mudam o estado (variáveis) de um programa
- Expressam ordens → programas imperativos são uma sequência de comandos para o computador executar
- Podemos denominá-lo de procedural por incluir sub-rotinas ou procedimentos para estruturação
- Sofrem de uma falta de flexibilidade dadas o caráter sequencial das instruções
- Exemplos: Ada / ALGOL / Assembly / Basic / C / Cobol / Fortran / Pascal / Python / Lua;

PARADIGMAS DE PROGRAMAÇÃO
PARADIGMA IMPERATIVO

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Os principais elementos da programação imperativa são:
  - Definição de tipos de dados (int x)
  - Expressões (x = a and b)
  - Atribuições (variável = valor)
  - Estruturas de controle de fluxo (programação estruturada)
  - Definição de sub-rotinas (programação procedimental)

PARADIGMAS DE PROGRAMAÇÃO
PARADIGMA IMPERATIVO

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

Vantagens

- Eficiência
- paradigma dominante
- bem estabelecido
- muito flexível

Desvantagens

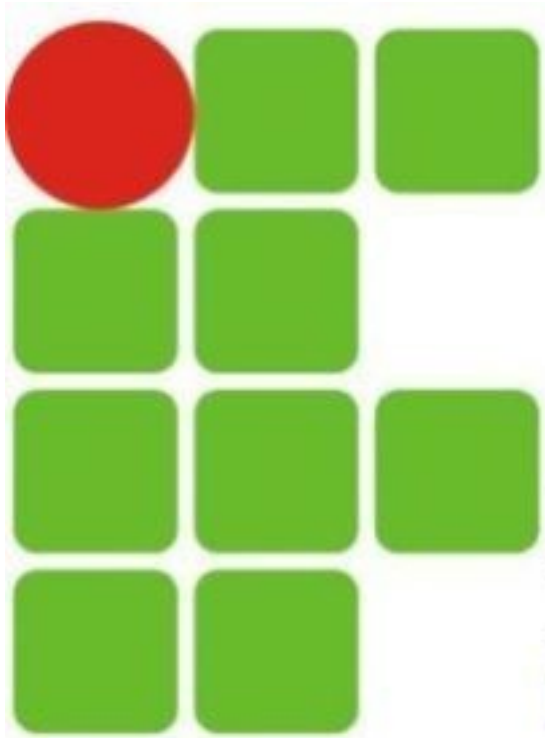
- difícil legibilidade
- Instruções demasiadamente profissional focaliza o "como" e não o "quê“
- Alto acoplamento
- Difícil manutenção

PARADIGMAS DE PROGRAMAÇÃO
PARADIGMA IMPERATIVO

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

```c
#include "stdio.h"

void main() {
    int fib0, fib1, temp, n;

    fib0 = 0;
    fib1 = 1;

    printf("Digite um número: ");
    scanf("%d", &n);

    if(n < 0) {
        printf("Número inválido.
            Digite um número diferente: ");
        scanf("%d", &n);
    }

    printf("Série de Fibonacci:\n");
    printf("%d\n", fib1);

    for(int i = 1; i < n; i++) {
        temp = fib0 + fib1;
        fib0 = fib1;
        fib1 = temp;

        printf("%d\n", temp);
    }
}
```

PROF: JOSÉ COUTO JÚNIOR



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

PARADIGMA ESTRUTURADO

PARADIGMAS DE PROGRAMAÇÃO
PARADIGMA ESTRUTURADO

- Preconiza que todos os programas possíveis podem ser reduzidos a apenas três estruturas:
  - Sequência
  - Decisão
  - Iteração
- Ênfase no uso de subrotinas, laços de repetição, condicionais e estruturas em bloco
- O PE foi o paradigma dominante na escrita de software até a programação orientada a objetos (POO)
- Estimula a criação de estruturas simples
- Evita o uso de instruções de desvio incondicional (goto, break)
- Uso de sub-rotinas e funções
- Exemplos: C / Basic / Pascal / Cobol / PHP;

PARADIGMAS DE PROGRAMAÇÃO
PARADIGMA ESTRUTURADO

- Os programas são vistos como compostos das seguintes estruturas de controle:
  - Sequência: de instruções ou sub-rotinas executadas em sequência (a=4; b=4*5)
  - Seleção/condicional: instruções são executadas ou não conforme o estado do programa (if, else, elif/elseif, endif)
  - iteração/repetição: instruções são executados até que o programa atinja um determinado estado (for, while, repeat, do..until)
  - recursão: instruções executadas com chamadas auto-referenciadas até que certas condições sejam satisfeitas

```
def fatorial(x):
    if x > 1:
        return x*fatorial(x-1)
    return x
```

PARADIGMAS DE PROGRAMAÇÃO
PARADIGMA ESTRUTURADO

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

Vantagens

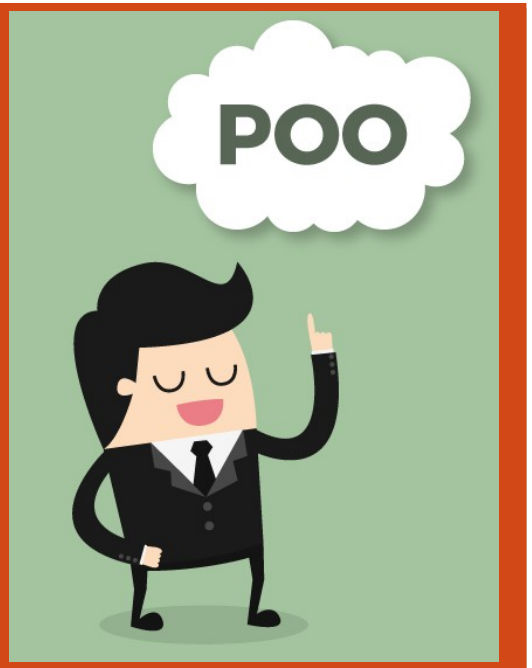
- Problemas podem ser quebrados em vários subproblemas
- Boa Legibilidade
- Fácil compreensão da estrutura

Desvantagens

- Dados separados das funções
- Difícil manutenção

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

PARADIGMA ORIENTADO A OBJETOS



PARADIGMAS DE PROGRAMAÇÃO
PARADIGMA ORIENTADO A OBJETOS

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Baseada na composição e interação de diversas unidades de softwares denominados objetos
- Trata os programas como uma coleção de objetos que se comunicam através de mensagens
- Aproxima-se do mundo real com objetos virtuais que representam objetos reais
- Baixo acoplamento
- Programação profissional
- Fácil manutenção

PROF: JOSÉ COUTO JÚNIOR

PARADIGMAS DE PROGRAMAÇÃO
PARADIGMA ORIENTADO A OBJETOS

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Concentra as responsabilidades nos lugares certos
- Métodos como comportamentos
- Estados como atributos
- Exemplos: Smalltalk / Python / Ruby / C++ / Object Pascal / Java / C# / Oberon / Ada / Eiffel / Simula / .NET

Este é o paradigma mais utilizado atualmente e deve continuar em uso por bastante tempo na indústria de software.

PARADIGMAS DE PROGRAMAÇÃO
PARADIGMA ORIENTADO A OBJETOS

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Vamos imaginar uma escola
- A principal entidade deste sistema será o aluno

Informações:
- Nome
- Matrícula
- Turma

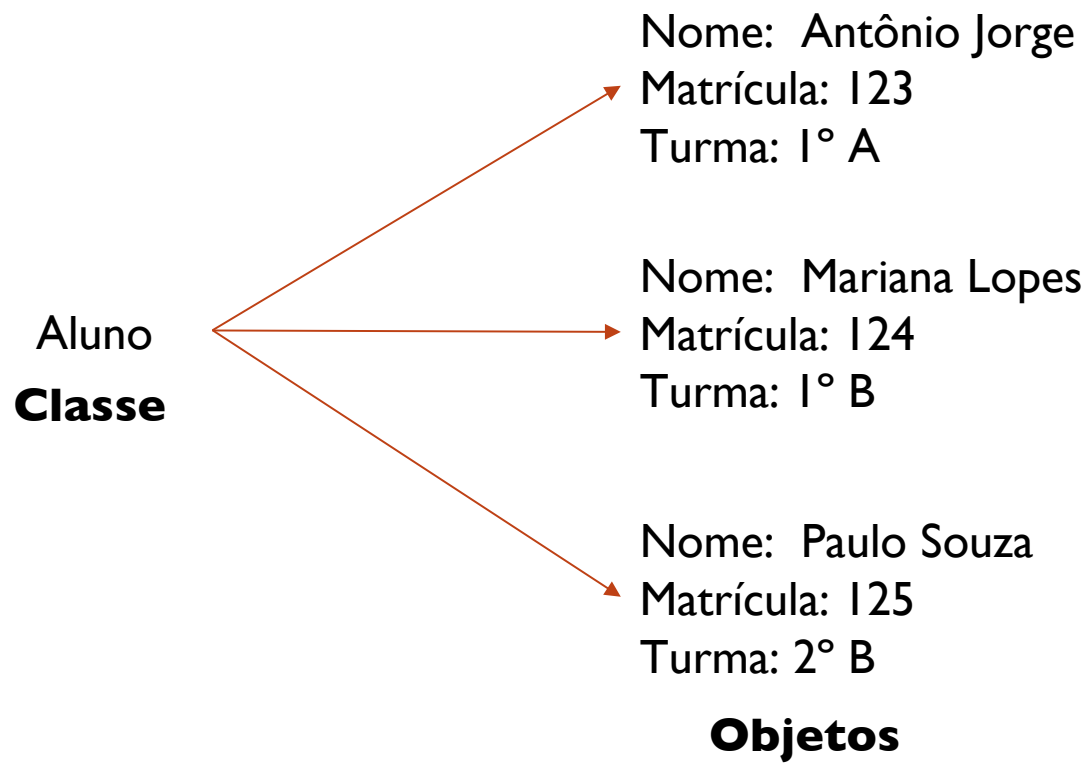
Atributos

Ações:
- Assistir Aula
- Tomar notas das aulas
- Estudar

Métodos

PARADIGMAS DE PROGRAMAÇÃO
PARADIGMA ORIENTADO A OBJETOS

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA



PARADIGMAS DE PROGRAMAÇÃO
PARADIGMA ORIENTADO A OBJETOS

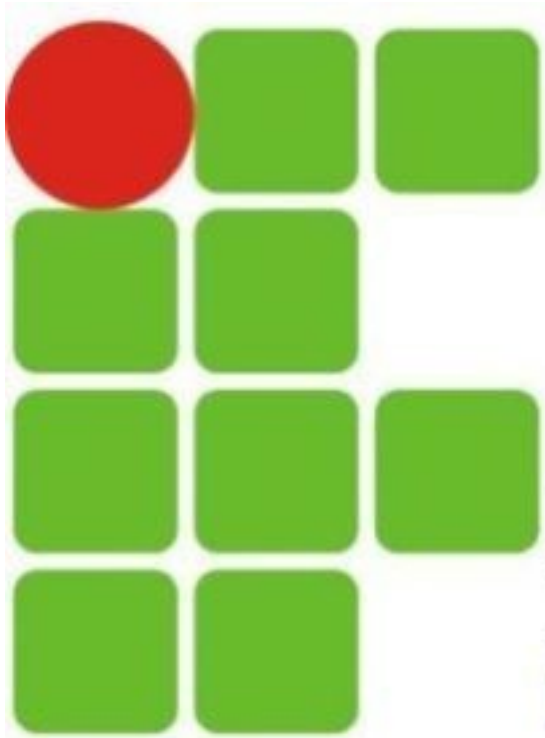
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

Vantagens

- Eficiência
- paradigma dominante
- bem estabelecido
- muito flexível
- a alteração de um módulo não incorre na modificação de outros módulos
- Reutilização do código

Desvantagens

- Exige maior conhecimento



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

PARADIGMA FUNCIONAL

PARADIGMAS DE PROGRAMAÇÃO
PARADIGMA FUNCIONAL

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Trata a computação como uma avaliação de funções matemáticas (entradas geram saídas)
- Não existe noção de estado
- Baseado em funções que buscam se aproximar das funções matemáticas (f(x) = expressão)
- Utiliza recursão
- Exemplos: Lambda / LISP / Haslkell / Miranda

PROF: JOSÉ COUTO JÚNIOR

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

PARADIGMA LÓGICO / DECLARATIVO

PARADIGMAS DE PROGRAMAÇÃO
PARADIGMA LÓGICO

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Programas são relações entre Entrada/Saída
- Programas são mais próximos de uma especificação do que a implementação tradicional
- Inclui características imperativas
- Utiliza lógica simbólica para expressar proposições, relações e inferir novas proposições
- Aplicação em sistema de inteligência artificial
- Define "o que deve ser feito" ao invés de "como deve ser feito"
- Exemplos: Prolog / QLISP / Mercury

PARADIGMAS DE PROGRAMAÇÃO
PARADIGMA LÓGICO - EXEMPLO PROLOG

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- star(sun).
- star(sirius).
- star(betelgeuse).
- orbits(mercury, sun).
- orbits(venus, sun).
- orbits(earth, sun).
- orbits(mars, sun).
- orbits(moon, earth).
- orbits(phobos, mars).
- orbits(deimos, mars).
- planet(X) :- orbits(X,sun).
- satellite(B) :- orbits(B,P), planet(P).
- ?-
- satellite(phobos).
- yes

REFERÊNCIAS

- https://www.infoescola.com/informatica/historia-da-programacao/
- https://brasilescola.uol.com.br/informatica/introducao-a-programacao.htm
- https://fit.faccat.br/~guto/artigos/Artigo_Paradigmas_de_Programacao.pdf
- https://slideplayer.com.br/slide/376609/
- https://sites.google.com/site/proffernandodesiqueira/disciplinas/paradigmas-de-linguagens-de-programacao/aula-1
- https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_imperativa

ATIVIDADE

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Qual a principal função das linguagens de programação?
- Qual a função do tradutor?
- Marque A para características de alto nível e B para características de linguagens de baixo nível:
  - ( ) Programas menores
  - ( ) Baixo acoplamento com a arquitetura do computador
  - ( ) Mais próxima da linguagem do homem
  - ( ) Alto acoplamento com a arquitetura do computador
  - ( ) Comandos executam tarefas complexas
  - ( ) Maior abstração
  - ( ) Comandos complexos

- A qual paradigma se refere as seguintes proposições:
  - ( ) Comandos executados na ordem que aparecem na memória
  - ( ) Baseado na arquitetura de Von Neumann
  - ( ) No coração desse paradigma, temos a ideia da atribuição
  - ( ) comunicação através de mensagens
  - ( ) elementos como objetos
  - ( ) Utiliza-se do conceito de classes e objetos
  - ( ) Atributos como características e métodos como comportamentos
  - ( ) Computação é vista como uma função matemática
  - ( ) Funções são o coração do paradigma
  - ( ) Permite interações através de recursão
  - ( ) A computação é a avaliação de funções matemáticas
  - ( ) Explora a recursão
  - ( ) Aplicação em sistemas de inteligência artificial