

nfografo tem como objetivo esboçar os paradigmas das rentes linguagens de programação de computadores.

# PARADIGMAS DE PROGRAMAÇÃO





# O QUE SÃO PARADIGMAS DE PROGRAMAÇÃO?

Paradigma de programação é um meio de se classificar as linguagens de programação baseado em suas funcionalidades. Um paradigma de programação fornece e determina a visão que o programador possui sobre a estruturação e execução do programa.

# TIPOS DE PARADIGMAS DE PROGRAMAÇÃO

# **PARADIGMA IMPERATIVO**

LINGUAGENS DE PROGRAMAÇÃO QUE BASEIAM-SE NO MODELO IMPERATIVO

O Paradigma Imperativo é baseado na arquitetura de Von Neumann. É o primeiro paradigma a existir e até hoje é o dominante.

Esse paradigma segue o conceito de um estado e de ações que manipulam esse estado, nele encontramos procedimentos que servem de mecanismos de estruturação. Podemos denominá-lo de procedural por incluir subrotinas ou procedimentos para estruturação.

ADA \* ALGOL \* ASSEMBLER \*
BASIC \* C \* COBOL \* FORTRAN
\* PASCAL \* PYTHON \* LUA

## **VANTAGENS**

As vantagens desse paradigma são: eficiência (porque embute o modelo de Von Neumann); modelagem "natural" de aplicações do mundo real; paradigma dominante e bem estabelecido; e também muito flexível.

## **DESVANTAGENS**

As desvantagens são: difícil legibilidade; as instruções são centradas no como e não no o que.

## PARADIGMA ESTRUTURADO

LINGUAGENS DE PROGRAMAÇÃO QUE BASEIAM-SE NO MODELO ESTRUTURADO

Este paradigma preconiza que todos os programas possíveis podem ser reduzidos a apenas três estruturas: seqüência, decisão e iteração. Tendo, na prática, sido transformada na Programação modular, a Programação estruturada orienta os programadores para a criação de estruturas simples em seus programas, usando as subrotinas e as funções. Foi a forma dominante na criação de software entre a programação linear e a programação

C \* BASIC \* PASCAL \* COBOL

## **VANTAGENS**

Os problemas podem ser quebrados em vários subproblemas, a boa legibilidade e a boa compreensão da estrutura deste paradigma motivam os programadores a iniciarem a programação pelo modelo estruturado.

## **DESVANTAGENS**

Os dados são separados das funções; Mudanças na estrutura dos dados acarreta alteração em todas as funções relacionadas. Gera sistemas difíceis de serem mantidos;





# PARADIGMA ORIENTADO **A OBJETOS**

LINGUAGENS DE PROGRAMAÇÃO **OUE BASEIAM-SE NO MODELO** ORIENTADO A OBJECTOS

Conceito A programação Orientada a Objetos é baseada na composição e interação de diversas unidades de softwares denominados objetos. O funcionamento de um software orientado a objetos se dá através relacionamento e troca de mensagens entre esses objetos. Esses objetos são classes, e nessas classes os comportamentos chamados de métodos e os estados possíveis da classe são chamados de atributos.

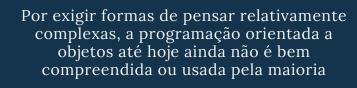
**SMALLTALK \* PYTHON \* RUBY** C++ \* OBJECT PASCAL \* JAVA C# \* OBERON \* ADA \* EIFFEL \* **SIMULA \* NET** 

# VANTAGENS



### **DESVANTAGENS**

Esse paradigma possui todas as vantagens do paradigma imperativo entre outras: a alteração de um módulo não incorre na modificação de outros módulos; quanto mais um módulo for independente, maior a chance dele poder ser reutilizado em outra aplicação.



# PARADIGMA FUNCIONAL

LINGUAGENS DE PROGRAMAÇÃO **OUE BASEIAM-SE NO MODELO FUNCIONAL** 

Este paradigma trata a computação como uma avaliação de funções matemáticas. Este método enfatiza a aplicação de funções, as quais são tratadas como valores de primeira importância, ou seja, funções podem ser parâmetros ou valores de entrada para outras funções e podem ser os valores de retorno ou saída de uma função.

LAMBDA \* LISP \* SCHEME \* ML \* **MIRANDA \* HASKELL** 

## **VANTAGENS**

#### Devido ao processo automático de alocação de memória, então efeitos colaterais no cálculo da função são eliminados. Sem estes efeitos, a linguagem assegura que o resultado da função será o mesmo para um dado conjunto de parâmetros não importando onde, ou quando, seja avaliada e é empregado em computações independentes para execução paralela. A recursividade em programação funcional pode assumir várias formas e é em geral uma técnica mais poderosa que o uso de laços do paradigma imperativo.

### **DESVANTAGENS**



Na programação funcional parecem faltar diversas construções frequentemente (embora incorretamente) consideradas essenciais em linguagens imperativas, como C. Por exemplo, não há alocação explícita de memória nem de variáveis.

# PARADIGMA LÓGICO

LINGUAGENS DE PROGRAMAÇÃO QUE BASEIAM-SE NO MODELO LÓGICO

Nesse paradigma programas são relações entre Entrada/Saída. Possui estilo declarativo. paradigma funcional. características imperativas, por questões de eficiência. Aplicações em prototipação em geral, sistemas especialistas, bancos de dados, etc.

POPLER \* CONNIVER \* QLISP \* PLANNER \* PROLOG \* MERCURY **OZ \* FRILL** 

### **VANTAGENS**



Possui a princípio todas as vantagens do paradigma funcional. E permite concepção da aplicação em um alto nível de abstração (através de associações entre E/S).

Variáveis de programa não possuem tipos, nem são de alta ordem.

**DESVANTAGENS** 



# REFERÊNCIAS

SAMPAIO, A. (2008) "PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO", HTTP://WWW.CIN.UFPE.BR/~IN1007/TRANSPARENCIAS/AULAINTRODUCAOPLP.PPT, AGOSTO.

PAULA, A. (2008) "PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO MOTION CAPTURE WHITE PAPER", HTTP://HTTP://WWW.INF.UNISINOS.BR/~ANAPAULA/DISCIPLINAS/60023/, AGOSTO.

FERNANDES, E., CARVALHO, K., VILLAR, L., GETIRANA, N. E GAUDÊNCIO, V. (2008) "PARADIGMAS DE LINGUAGEM DE PROGRAMAÇÃO MOTION CAPTURE WHITE PAPER", HTTP://HTTP://WWW.INF.UNISINOS.BR/~ANAPAULA/DISCIPLINAS/60023/, AGOSTO.

PROGRAMAÇÃO FUNCIONAL HTTP://PT.WIKIPEDIA.ORG/WIKI/PROGRAMA%C3%A7%C3%A3O\_FUNCIONAL

PROGRAMAÇÃO ESTRUTURADA HTTP://LABES.INF.UFES.BR/VSOUZA/SITES/DEFAULT/FILES/CURSOOOSLIDES03.PDF