

	DATA	CONTEUDO	CH	DIA
1	04/fev	Apresentação da disciplina, Classificação paradigmas	4	2af
2	05/fev	Discussão Cap1 Sebesta - Paradigmas imperativo/declarativo	4	3af
3	06/fev	Paradigmas logico, funcional, concorrente	4	4af
4	07/fev	Paradigmas logico, funcional, concorrente	4	5af
5	08/fev	Prática de programação - Implementações de programas	4	6af
6	09/fev	SÁBADO - Prática de programação - presencial 9h-12h	8	sab
7	11/fev	AVALIAÇÃO - Prova teórica	4	2af
8	12/fev	Elaboração do seminário - não supervisionada	4	3af
9	13/fev	Elaboração do seminário - não supervisionada	4	4af
10	14/fev	Prática de programação - Implementações de programas	4	5af
11	15/fev	Prática de programação - Implementações de programas	4	6af
12	16/fev	SÁBADO - Finalização do seminário - não supervisionada	8	sab
13	18/fev	Prática de programação - Implementações de programas	4	2af
14	19/fev	AVALIAÇÃO - Prova prolog	4	3af
15	20/fev	AVALIACAO - Seminários sobre Linguagens	4	4af
16	21/fev	AVALIACAO - Seminários sobre Linguagens	4	5af
			72	

Atividades da disciplina

M1

Resumo (3) + Prova teórica (7)

M2

Exercícios (3) + Prova prática (7)

M3

Seminário: slides (4) + apresentação (5) + questionamentos (1)

Paradigmas de Programação 4p 19/1

EMENTA

- ▣ Classificação dos paradigmas
- ▣ Paradigmas Declarativos e Imperativos
- ▣ Programação Funcional
- ▣ Programação Lógica
- ▣ Programação Concorrente
- ▣ Prática em programação

Conceitos Preliminares

1. Princípios
2. O que é Paradigma?
3. O que é Paradigma de Programação?
4. Características de Paradigma de Programação
5. Classificação
6. Ranking de linguagens de programação

Princípios

- Linguagem de programação (LP) X linguagem natural
 - Ambas permitem comunicação entre pessoas e computadores
 - LP facilita apenas comunicação de ideias computacionais, porque tem domínio de expressão mais reduzido
 - LP deve satisfazer requisitos diferentes

Princípios de projeto

- Sintaxe: descreve o que constitui um programa estruturalmente correto
- Nomes e tipos: vocabulário de uma LP inclui um conjunto de regras cuidadosamente projetado para nomear entidades (variáveis, funções, classes, parâmetros, etc.). Um sistema de tipos bem especificado permite ao compilador executar uma rigorosa verificação de tipos.
- Semântica: determina o efeito de cada comando sobre os valores das variáveis no programa.

O que é Paradigma?

A definição do dicionário Aurélio para "paradigma":

1. Algo que serve de exemplo geral ou de modelo.
2. Conjunto das formas que servem de modelo de derivação ou de flexão.
3. Conjunto dos termos ou elementos que podem ocorrer na mesma posição ou contexto de uma estrutura.

O que é Paradigma de Programação?

- “(...) padrão de resolução de problemas que se relaciona a um determinado gênero de programas e linguagens.”
Tucker, Noonan, 2009

O paradigma de uma linguagem de programação é a sua identidade.

Corresponde a um conjunto de características que, juntas, definem como ela opera e resolve os problemas. Algumas linguagens, inclusive, possuem mais de um paradigma, são as chamadas multiparadigmas.

O que é Paradigma de Programação?

- Modelo, padrão ou estilo de programação suportado por linguagens que agrupam certas características comuns.
- Uma LP é um método padronizado para expressar instruções para um computador. É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador.
- Um paradigma de programação fornece (e determina) a visão que o programador possui sobre a estruturação e execução do programa.

Características de Paradigmas de Programação

- Gramática e significado bem definidos [sintaxe: gramática (forma); semântica: significado]
- Implementável (executável) com eficiência “aceitável”
- Universal: deve ser possível expressar todo problema computável
- Natural para expressar problemas (em um certo domínio de aplicação)

Classificação – Grau de Abstração

- Linguagem *assembly* de programação de baixo nível, próximas a máquina (1940)
- Linguagem de programação de alto nível, no nível da especificação de algoritmos, como Pascal e Fortran
- Linguagem de programação de muito alto nível, no nível da especificação declarativa, como SQL

(Tucker, Noonan, 2009)

Classificação – Estrutura de tipos

Linguagens fortemente tipadas (ou tipificadas): todas as variáveis tem um tipo específico e seus tipos são importantes para a linguagem. Normalmente também são **estaticamente** tipadas - uma vez que a variável foi declarada com um tipo assim se manterá até o final, e normalmente possuem declaração explícita de tipo. (Java, C#, Fortran, Cobol)

Linguagens fortemente tipadas e dinamicamente tipadas: os tipos das variáveis são importantes, porém, podem ser alterados durante a execução do programa. (Ruby)

Linguagens fracamente tipadas: aquelas que não se importam com o tipo de dados contido em uma variável. Permitem que o programador não tenha de fazer conversões de tipos (cast). (PHP, JavaScript)

Linguagens não tipadas: aquelas em que existe apenas um tipo genérico ou mesmo nenhum tipo de dados. (Assembly, BCPL, Perl)

Classificação – Geração (MacLennan, 99)

- 1a – suas estruturas de controle são aparentemente orientadas a máquina; instruções condicionais dependem fortemente de instruções de desvio incondicional como o *goto*)
- 2a – suas estruturas de controle são estruturadas de minimizando/dispensando uso de instruções *goto*.
- 3a – ênfase em simplicidade e eficiência; suas estruturas de dados mostram um deslocamento da máquina para a aplicação e estruturas de controle são mais simples e eficientes.
- 4a – abstração de dados; enfoque em modularização e no encapsulamento.
- 5a – agrupamento de diversos paradigmas como a orientação a objeto e o paradigma funcional, paradigma lógico.

Classificação – Geração (Bal; Grune,94)

Enfatizando mais o aspecto da aplicação:

- 1ª geração - linguagem de máquina.
- 2ª geração - linguagens de montagem (Assembly).
- 3ª geração - linguagens procedurais (Fortran, C, C++).
- 4ª geração - linguagens aplicativas (SQL).
- 5ª geração - linguagens lógicas (Prolog) e linguagens funcionais (Lisp), voltadas a IA.
- 6ª geração - redes neurais.

Appleby e Vandekopple (1997) consideram apenas as 4 primeiras gerações acima.

Classificação – Paradigmas de Programação

Diferentes linguagens de programação podem ser agrupadas segundo o paradigma que seguem para abordar a sua sintaxe:

- Programação imperativa: programação estruturada e programação orientada a objetos
- Programação declarativa: programação funcional e programação lógica
- Programação concorrente

Além destes, sabe-se que existem outros mais recentes (programação orientada a eventos, aspectos...)

Classificação – Paradigmas de Programação

Os paradigmas de programação são muitas vezes diferenciados pelas técnicas de programação que proíbem ou permitem.

O relacionamento entre paradigmas de programação e linguagens de programação pode ser complexo pelo fato de linguagens de programação **poderem suportar mais de um paradigma.**

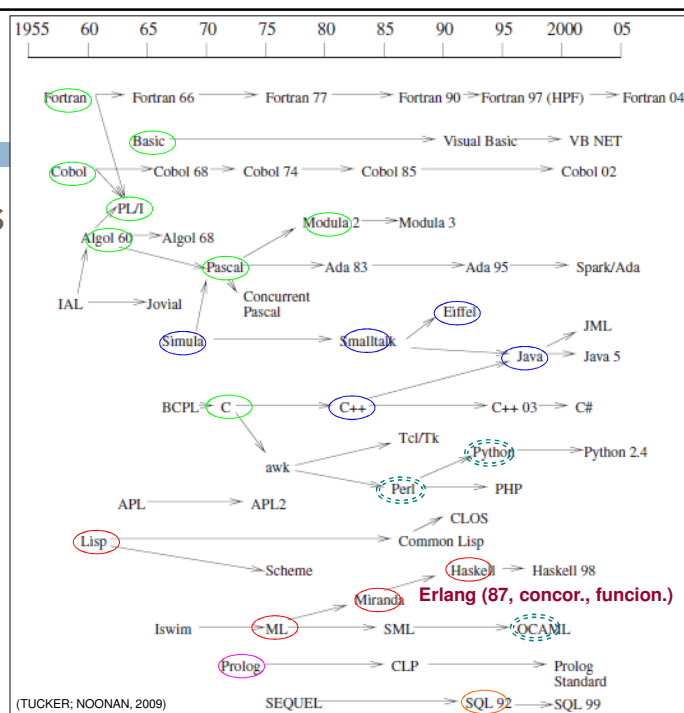
Classificação – Paradigmas de Programação

- A classificação de linguagens em paradigmas é uma consequência de decisões de projeto que impactam radicalmente a forma na qual uma aplicação real é modelada do ponto de vista computacional.
- A relação entre paradigmas e linguagens de programação tende a ficar complexa pois determinada linguagem pode conter vários tipos de paradigmas => confusão aos programadores.
- Mesmo havendo mais de um paradigma, há um modelo principal influenciando fortemente o processo de criação de programas.

RESUMO

História das linguagens de programação

- Imperativo
- Declarativo
- POO
- Funcional
- Lógico
- Multiparadigma



Evolução das linguagens

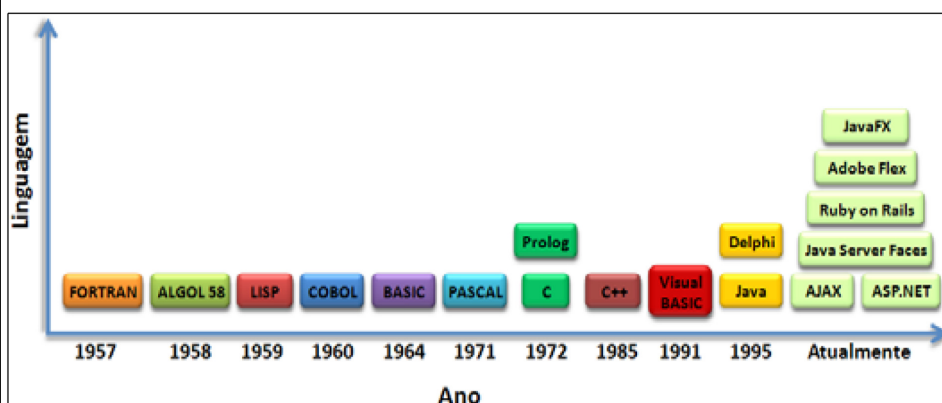
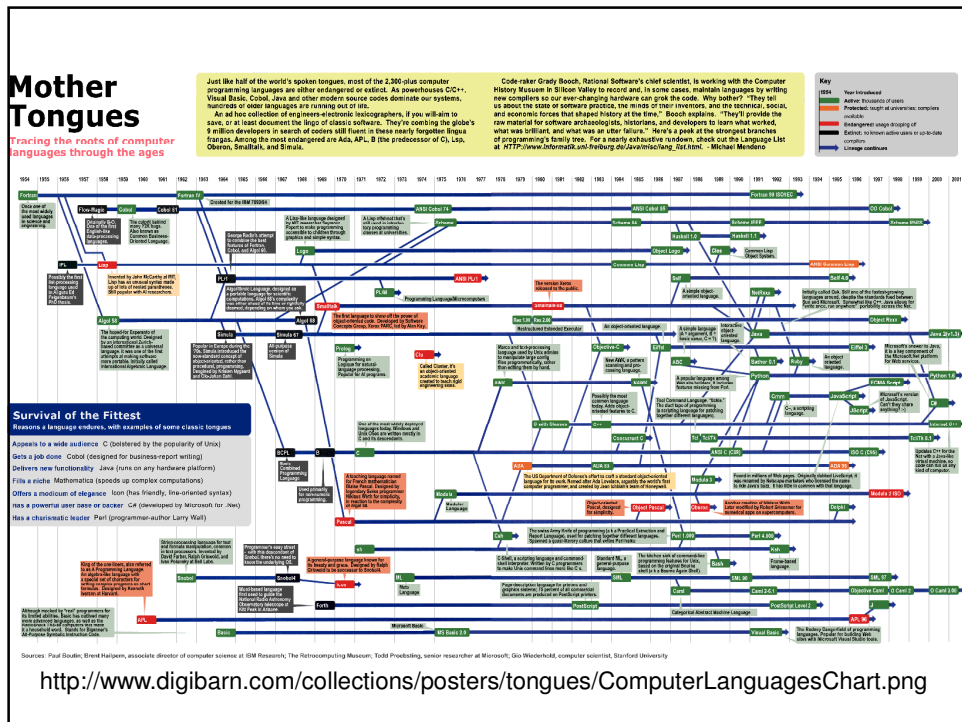


Figura 2 *Ordem cronológica da criação das principais linguagens de programação.*



Jan 2019	Jan 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.904%	+2.69%
2	2		C	13.337%	+2.30%
3	4	▲	Python	8.294%	+3.62%
4	3	▼	C++	8.158%	+2.55%
5	7	▲	Visual Basic .NET	6.459%	+3.20%
6	6		JavaScript	3.302%	-0.16%
7	5	▼	C#	3.284%	-0.47%
8	9	▲	PHP	2.680%	+0.15%
9	-	▲	SQL	2.277%	+2.28%
10	16	▲	Objective-C	1.781%	-0.08%
11	18	▲	MATLAB	1.502%	-0.15%
12	8	▼	R	1.331%	-1.22%
13	10	▼	Perl	1.225%	-1.19%
14	15	▲	Assembly language	1.196%	-0.86%
15	12	▼	Swift	1.187%	-1.19%

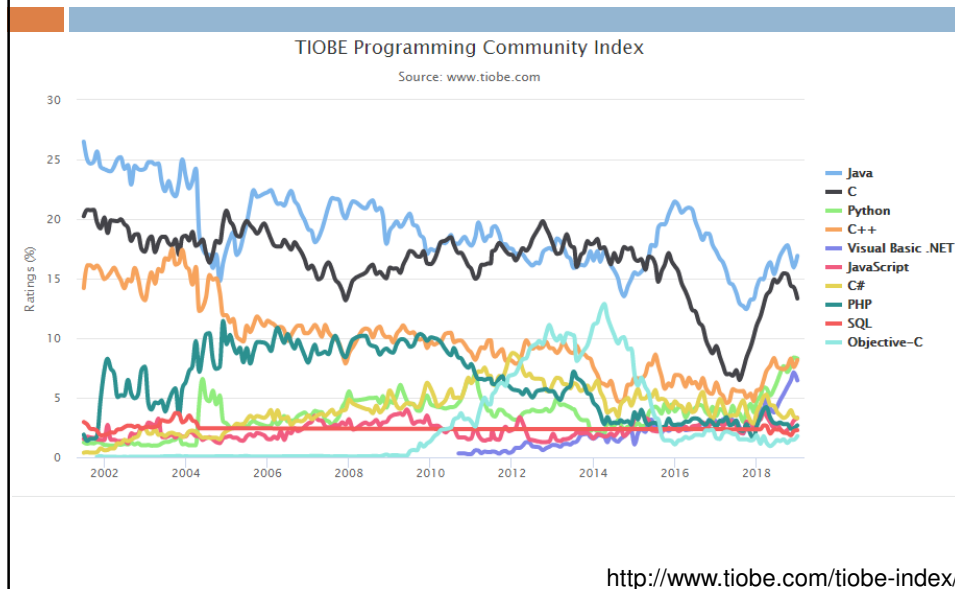
<http://www.tiobe.com/tiobe-index/>

TILOBE,
Jan/2019























Ranking
das
languagens

Ranking das linguagens

TIOBE,
Jan/2019



Recently IEEE spectrum comes with the ranking sheet of top programming language 2019 according to their popularity - Programming Languages Popularity

Language Rank	Types	Spectrum Ranking
1. Python	 	100.0
2. C	  	99.7
3. Java	  	99.5
4. C++	  	97.1
5. C#	  	87.7
6. R		87.7
7. JavaScript	 	85.6
8. PHP		81.2
9. Go	 	75.1
10. Swift	 	73.7

Este ranking é conduzido através da ponderação e combinação de 12 indicadores de 10 fontes de dados (como o IEEE Xplore biblioteca digital, GitHub, e CareerBuilder).

Linguagens para ficar de olho e tendências para 2019

<https://itmidia.com/16-linguagens-de-programacao-para-conquistar-um-bom-emprego-em-2019/>

<https://www.profissionaisti.com.br/2019/01/7-linguagens-de-programacao-para-voce-aprender-em-2019/>

<https://devlooper.blog.br/updates/6-tendencias-do-mercado-de-programacao-para-ficar-de-olho-em-2019/>

Outros links

- Um programa em várias linguagens diferentes:
<http://www.99-bottles-of-beer.net/>
- Benchmarks de linguagens:
<http://benchmarksgame.alioth.debian.org/>
- History of Programming Languages and Their Evolution:
<http://www.scriptol.com/programming/history.php>
- Wikipedia:
https://pt.wikipedia.org/wiki/Hist%C3%B3ria_das_linguagens_de_programa%C3%A7%C3%A3o
http://pt.wikipedia.org/wiki/Anexo:Lista_de_linguagens_de_programa%C3%A7%C3%A3o

Referências

- APPLEBY, Doris; VANDEKOPPLE, Julius J. *Programming Languages. Paradigm and Practice*. 2ª ed. New York: McGraw-Hill, 1997.
- BAL, Henri E.; GRUNE, Dick. *Programming Language Essentials*. Wokingham: Addison-Wesley, 1994.
- MACLENNAN, Bruce J. *Principles of Programming Languages: Design, Evaluation and Implementation*. 3ª ed. Oxford: Oxford University Press, 1999.
- SEBESTA, Robert W. *Conceitos de Linguagens de Programação*. 5ª ed. Bookman, 2003.
- TUCKER, Allen B.; NOONAN, Robert E. *Linguagens de Programação: princípios e paradigmas*. 2ª ed. São Paulo: McGraw-Hill, 2009.