UNIP

UNIVERSIDADE PAULISTA

Paradigmas de Linguagens

Aula 1: Introdução e Conceitos Básicos

Professora Sheila Cáceres

O que é um paradigma???

O que é um paradigma???

O que é um paradigma???

O que é um paradigma?

A palavra paradigma tem origem numa palavra grega que significa padrão ou exemplo.

- Modelo
- Padrão

O que é um paradigma de programação?

Modelo, padrão ou estilo de programação suportado por linguagens que agrupam certas características comuns.

Conjunto de características que servem para categorizar um grupo de linguagens.

Linguagem de programação (LP) & Programa

- **Linguagem de programação:**

  Ferramenta utilizada para escrever programas.

- **Programa:**

  Conjunto de instruções a serem seguidas pelo computador para realizar um determinado processo

O que é um Algoritmo

- Um algoritmo é uma sequência de passos para realizar uma tarefa ou resolver um problema.
- Em nosso dia a dia utilizamos algoritmos para realizar nossas atividades, definindo a sequência de atividades que devemos fazer para atingir um objectivo.
- Poderia se dizer:
  - Um **algoritmo** é um **programa abstrato.**
  - Um **programa** é um **algoritmo concretizado**

Algo de história

- Antigamente só usavam-se linguagens muito simples (poucas instruções que realizavam ações muito elementares apenas para um tipo de computador específico)-> Linguagens de baixo nível
- Com o avanço da computação as aplicações tornavam-se mais complexas (as linguagens de baixo nível reduziam a produtividade dos programadores).
- Assim, surgiram linguagens de programação de alto nível (conjunto mais amplo de instruções, não apenas para um tipo de computador).

Porque existem tantas linguagens de programação?

Por que existem tantas linguagens de programação?
Resposta: Para corrigir as falhas das anteriores
Resposta certa ou errada?

Qual é a melhor linguagem?
Resposta: A que tem menos falhas.
Resposta certa ou errada?

Porque existem tantas linguagens de programação?

- A maioria surgiu para sanar uma necessidade diferente numa determinada área numa época específica. (video)

Exemplos

- Java (video)
  - Nasceu para substituir C / C++?
  - Nasceu pela orientação a objetos?

    NÃO, nasceu para oferecer **Portabilidade**
- Smalltalk: programação intuitiva
- PERL: linguagem poderosa e prática

LPs

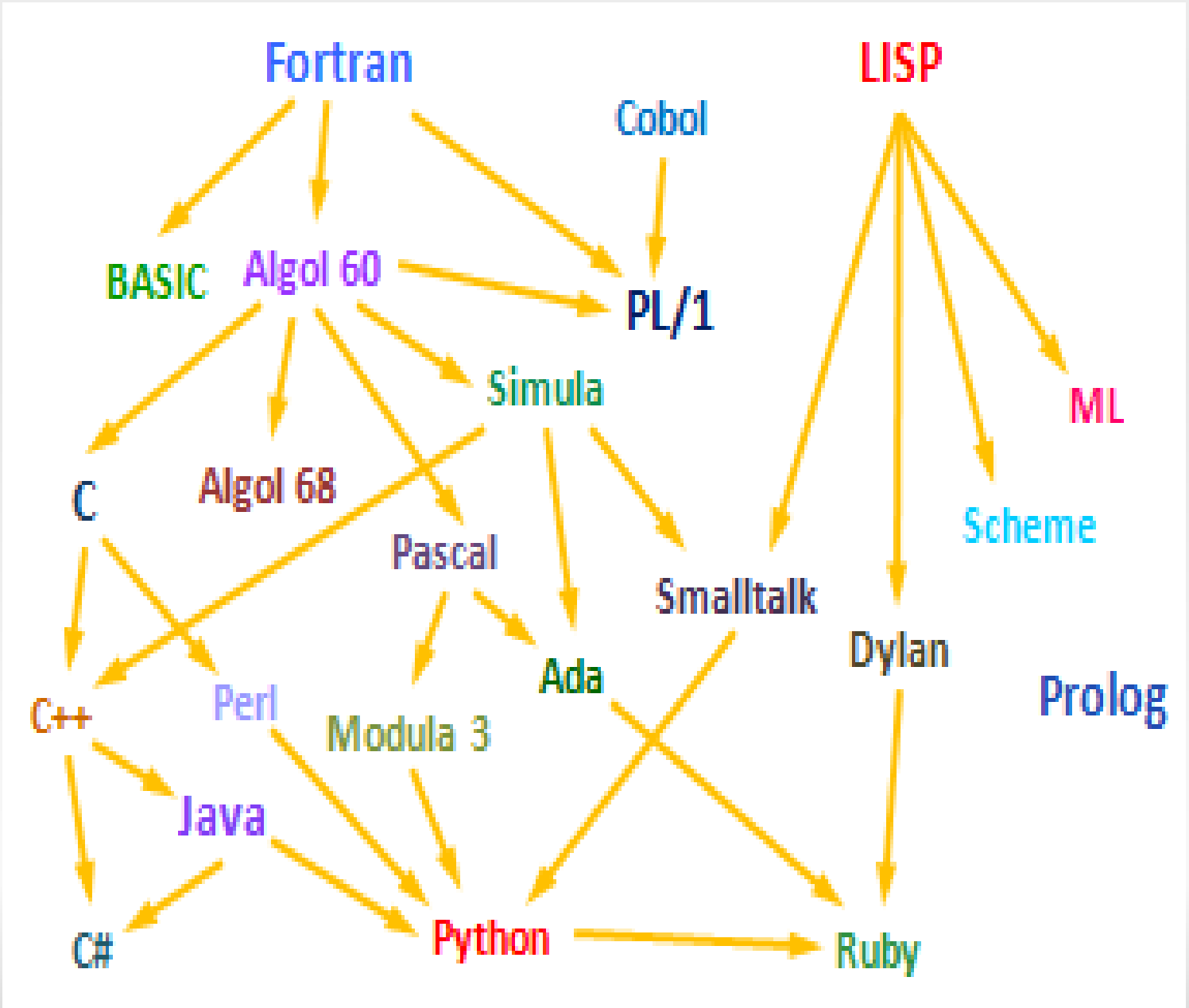
- Algumas outras linguagens foram criadas para competir.
  - **Exemplo:**
    - C#, veio para competir com Java
- Simula nasceu pela necessidade de realizar simulações de eventos discretos.
- MUMPS nasceu pela necessidade de desenvolver aplicações baseadas em bancos de dados para um hospital.

LPs

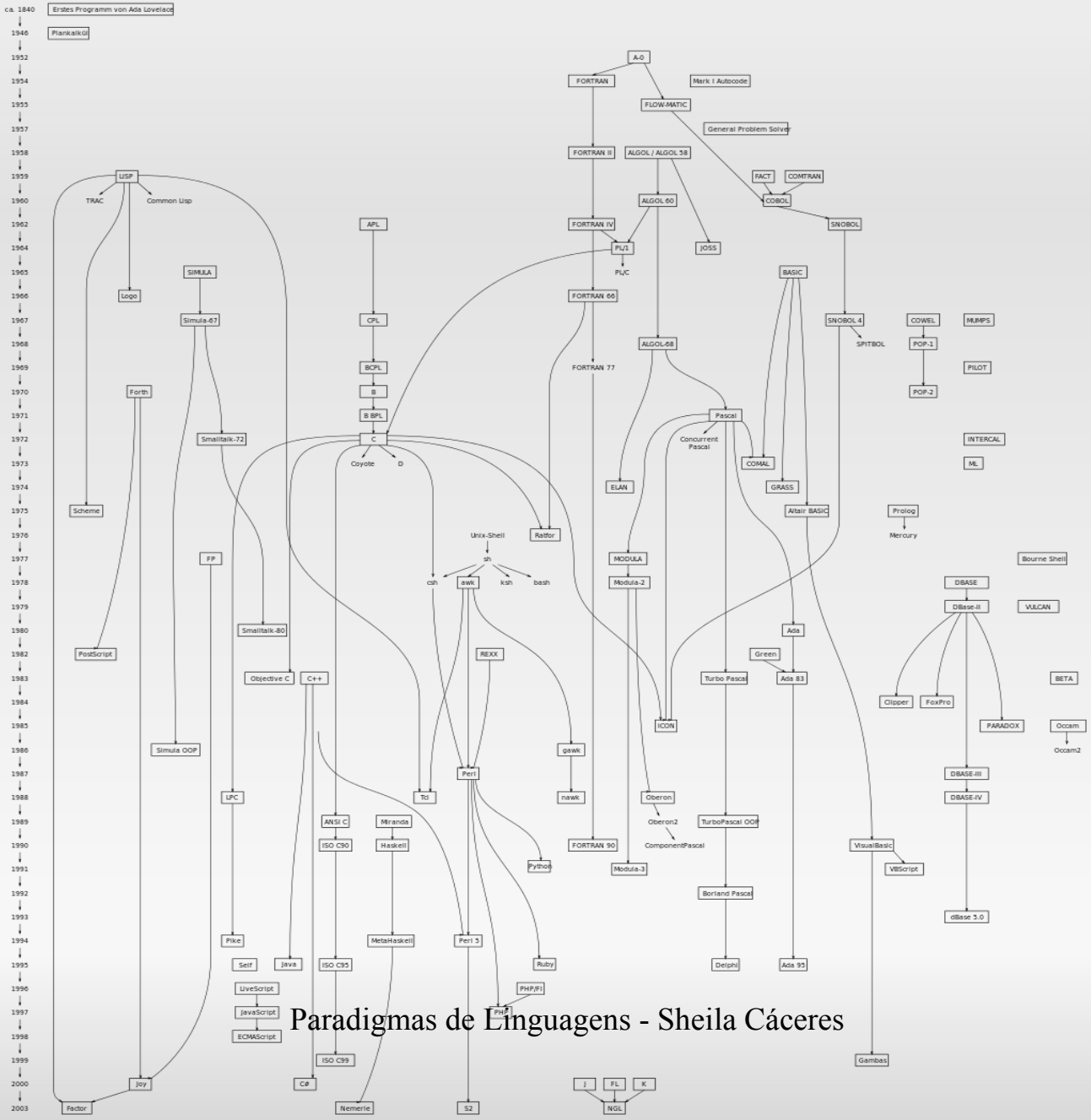
- Cada linguagem tem pontos fortes e fracos.
- Não ha a melhor linguagem.
  - Senão:

Uma linguagem é a melhor para determinado problema numa determinada situação

LPs


Fortran
LISP
Cobol
BASIC
Algol 60
PL/1
Simula
ML
C
Algol 68
Scheme
Pascal
Smalltalk
Dylan
Ada
Prolog
C++
Perl
Modula 3
Java
C#
Python
Ruby

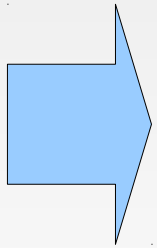
LPs


ca. 1840
Erstes Programm von Ada Lovelace
1946
Plankalkül
1952
1954
1955
1957
1958
1959
1960
1962
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2003
A-0
FORTRAN
Mark I Autocode
FLOW-MATIC
General Problem Solver
FORTRAN II
ALGOL / ALGOL 58
FACT
COMTRAN
COBOL
LISP
TRAC
Common Lisp
ALGOL 60
FORTRAN IV
SNOBOL
APL
PL/1
JOSS
PL/C
SIMULA
BASIC
Logo
FORTRAN 66
Simula-67
CPL
SNOBOL 4
COWEL
MUMPS
ALGOL-68
SPITBOL
POP-1
BCPL
FORTRAN 77
PILOT
Forth
B
POP-2
B BPL
Pascal
Smalltalk-72
C
Concurrent Pascal
INTERCAL
Coyote
D
COMAL
ML
ELAN
GRASS
Scheme
Altair BASIC
Prolog
Mercury
Unix-Shell
Ratfor
FP
sh
MODULA
Bourne Shell
csh
awk
ksh
bash
Modula-2
DBASE
DBase-II
VULCAN
Smalltalk-80
Ada
PostScript
REXX
Green
Objective C
C++
Turbo Pascal
Ada 83
BETA
Clipper
FoxPro
ICON
PARADOX
Occam
Occam2
Simula OOP
gawk
DBASE-III
Perl
LPC
Tcl
nawk
Oberon
DBASE-IV
ANSI C
Miranda
Oberon2
TurboPascal OOP
ISO C90
Haskell
FORTRAN 90
VisualBasic
Python
Modula-3
VBScript
ComponentPascal
Borland Pascal
dBase 3.0
Pike
MetaHaskell
Perl 5
Ruby
Self
Java
ISO C95
Delphi
Ada 95
UveScript
PHP/FI
JavaScript
PHP
ECMAScript
ISO C99
Gambas
C#
Joy
Factor
Nemerle
S2
J
FL
K
NGL

- As inúmeras Linguagens de Programação tem seu próprio conjunto de símbolos e regras.
- Podem ser estudadas tendo em conta os principais conceitos que lhes são comuns (paradigmas comuns).
- E para que estudá-las???

  Linguagens de programação são usadas em:

  - Banco de dados
  - Sistemas operacionais
  - Descrição de hardware
  - Linguagem natural
  - Processamento de imagens
  - Etc, etc, etc



Fundamental para a computação

E para que estudar LPs?

Estudando as LPs

- Maior capacidade para desenvolver soluções computacionais para problemas
- Maior habilidad para usar uma LP
- Maior capacidade para escolher LPs apropriadas
- Maior habilidade para aprender novas LPs.
- Maior habilidade para projetar novas LPs

Exemplos

Olá Mundo... arquivo anexo

A conteúdo a seguir deste arquivo não entra para a NP1. Somente será tomado em conta somente para a NP2, sub, exame.

LPs no processo de Desenvolvimento de Software

- O objetivo das LPs é tornar mais efetivo o processo de desenvolvimento de software.
- Esse processo existe para tornar mais produtiva a geração e manutenção de software e para garantir que ele seja produzido atendendo a padrões de qualidade.
- Como as LPs podem apoiar esse processo?

  Apoiando as propriedades que identificam um software de qualidade.

Propriedades requeridas em um software de qualidade

- **Confiabilidade**:
  - atendimento adequado da especificação funcional, garantia de segurança contra erros, e integridade dos dados manipulados pelo software.

Propriedades requeridas em um software de qualidade

- Como uma LP pode promover a confiabilidade?
  - Facilitando a existência de ferramentas computacionais que verifiquem a ocorrência de erros nos programas.
  - ***Exemplo***: Em C, a declaração de variáveis é obrigatória Caso um usuário digite um nome incorreto, um verificador de erros pode identificá-lo porque não foi declarado.

Propriedades requeridas em um software de qualidade

- **Manutenibilidade**:
  - Facilidade de alteração do software.

Propriedades requeridas em um software de qualidade

- Como uma LP pode promover a manutenibilidade?
  - Fornecendo mecanismos que permitam a sua adaptação a diferentes contextos.
  - ***Exemplo***: Em Java (ver exemplo), caso um programa utilize uma constante para definir o tamanho máximo de um vetor, basta modificar essa constante para adaptar todo o programa a um aumento no tamanho máximo do vetor.

Propriedades requeridas em um software de qualidade

- **Eficiência**:
  - Uso otimizado dos recursos computacionais em termos de tempo de execução, espaço de memoria utilizado, uso de dispositivos perifericos.

Propriedades requeridas em um software de qualidade

- Como uma LP pode promover a eficiencia?
  - Incentivando o uso de mecanismos computacionalmente eficientes.
  - ***Exemplo***: A linguagem FORTRAN não permite o usso de recursão para tornar mais eficiente o processamento e o consumo de memoria.

Processo de desenvolvimento de Software

- Segundo PRESSMAN é considerado como um processo com 5 etapas
  - Especificação de requisitos
  - Projeto de software
  - Implementação
  - Validação
  - Manutenção

Especificação de requisitos

- Identificação da funcionalidade que é requerida
- Estudo de viabilidade e custo do sftware.

- LPs tem pouca influencia nessa etapa,
- O conhecimento sobre LPs pode ser usado no estudo de viabilidade para ajudar a responder se é possível desenvolver o software no período de tempo desejado.

Projeto de Software

- Projetar o sistema de programação.
- Identificação dos módulos que compõem a arequitetura do sistema, as estruturas de dados de cada módulo, as interfaces de comunicação entre módulos, etc.

- LPs oferecem suporte ao paradigma usado.
- Exemplos: Java pode ser adequada quando o estilo é orientado a objetos.

Implementação

- Programação dos módulos do software.

- Lps são essenciais nessa etapa pois os programas devem ser escritos em uma linguagem.
- A tapa é a mais atendida por ferramentas como editores de texto que destacam os vocabulos da linguagem idntam automaticamente o texto, analisadores lexicos, sintáticos e semanticos de programas e bibliotecas de subprogramas e modulos

Validação

- Verificar se o sistema satisfaz as exigencias das especificações de requisitos.

- As Lps podem auxiliar na validação em diversas formas.
- Exemplo: Algumas linguagens facilitam a construção de depuradores de erros.

Manutenção

- É necessário que o software:
  - seja capaz de facilitar a correção de erros residuais.

    Erro residual: Erros descobertos após a sua liberação para o usuário.
  - se adapte a mudanças no seu contxto de aplicação
  - Atenda novas demandas.
- LPs com recursos de modularização tendem a gerar programas mais fáceis de serem mantidos (pois alterações em um módulo não interferem nos outros módulos).

Propriedades desejáveis em uma Linguagem de Programação

- Aproveitamento do tempo do programador é vital no desenvolvimento de software (Lps devem enfatizar esse aspecto).
- Algumas propriedades são:
  - Legibilidade
  - Redigibilidade
  - Confiabilidade
  - Eficiência
  - Ortogonalidade
  - Reusabilidade
  - Modificabilidade
  - Portabilidade

Legibilidade: Facilidade para se ler e entender um programa

- Exemplo: A não obrigatoriedade de um marcador especifico em C pode ser dificil de ser entendido (Não legivel).

```
if( x>1)
    if( x==2 )
                x=3;
else
    x=4;
```

- **Redigibilidade:** Possibilita ao programador se concentrar nos algoritmos centrais do programa, sem se preocupar com aspectos não relevantes.
  - Exemplo: Em Linguagem máquina o programador com frequencia se preocupa com detalhes de implementação (não redigivel).
- **Confiabilidade:** (mencionado anteriormente).

- **Eficiência**: (mencionado anteriormente) deacordo com as demandas para o tipo de aplicação. Capacidade da LP de fornecer meios adequados para atingir o objetivo.
  - Ex: aplicações de automação em tempo real normalmente requerem o uso de Lps que minimizem o tempo de execução e de aceso aos dispositivos periféricos.
- **Facilidade de aprendizado**:
  - **Exemplo:** Linguagem C versus linguagem Assembler

- **Ortogonalidade**: Capacidade de que a LP permita combinar conceitos básicos sem que se produzam efeitos anômalos.
- O programador pode prever com segurança o comportamento de uma determinada combinação de conceitos.
  - Exemplo

```
iny x, y = 2, z = 3;
x = x + y;
byte a, b = 2, c = 3;
a = b + c;
```

- **Reusabilidade**: Possibilidade de reutilizar o mesmo código para diversas aplicações
- **Modificabilidade**: Possibilita alterar o programa em função de novos requisitos (ver ex).
- **Portabilidade**: Capacidade de um programa se comportar da mesma maneira independente da arquitetura computacional (hardware ou sistema operacional) sobre a qual estão sendo executados.

Referências Bibliográficas

* VAREJÃO, Flavio. Linguagens de Programação. Campus, 2004.

* http://www.vidageek.net/2008/08/11/linguagens-de-programacao/