Linguagens de Programação

Introdução

Carlos Bazilio
carlosbazilio@id.uff.br
http://www.ic.uff.br/~bazilio/cursos/lp

???

Pascal	C
aux := 0 for i:=1 to 10 do aux := aux + i	aux = 0 for (i=1; i<=10; i++) aux = aux + i
10: i = 1 20: if i > 10 goto 60 30: aux = aux + i 40: i = i + 1 50: goto 20 60:	10: i := 1 20: aux := aux + i 30: if i == 10 goto 60 40: i := i + 1 50: goto 20 60:

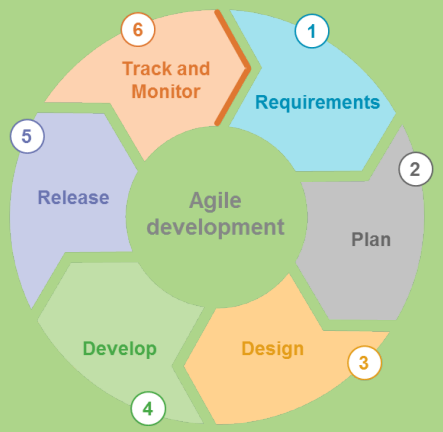
Motivação

- Algumas questões filosóficas
  - Por quê existe mais de uma linguagem de programação?
  - As linguagens de programação são equivalentes em termos de poder de expressão?
  - O que difere uma linguagem de outra?
  - Que impactos podemos causar em nosso sistema na escolha de uma linguagem em detrimento de outra?
  - Linguagens de programação e linguagens naturais são análogas?

Processo de Desenvolvimento de Software Geral

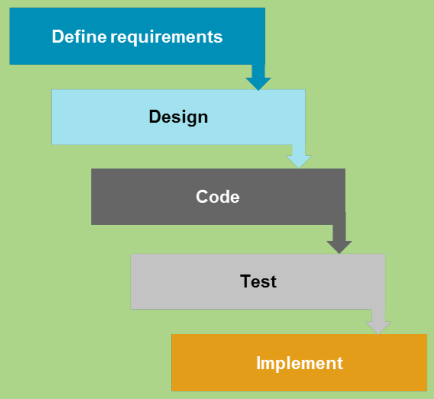
- Especificação de Requisitos
- Projeto (Dentre outras questões, escolha da linguagem)
- Codificação (Uso primário da linguagem)
- Teste (Eventualmente utilizando recursos da própria linguagem)
- Implantação
- Evolução

Agile



- Continuous cycles
- Small, high-functioning, collaborative teams
- Flexible/continuous evolution
- Customer involvement

Waterfall



- Sequential/linear stages
- Upfront planning and in-depth documentation
- Best for simple, unchanging projects
- Close project manager involvement

Processo de Desenvolvimento de Software Geral

- Apesar de termos citado a Linguagem de Programação apenas no Projeto, Codificação e Teste, projetistas de linguagem tem, cada vez mais, depositado esforços no sentido de ampliar esta área de atuação
- Exemplos:
  - JUnit (teste de unidade em Java)
  - Ferramenta de especificação de requisitos que geram artefatos de código

Características Importantes

- Legibilidade: facilidade de leitura e compreensão de programas numa dada linguagem
  - Simplicidade global dos programas, como por ex.:
    - sem sobrecarga de operadores (* em C)
    - instruções que realizam a mesma operação (++ em C)
  - Ortogonalidade, ou seja, uso coerente das construções da linguagem, por ex. (em C):
    - Matrizes e Registros como tipos de retorno de função
    - Uma matriz é sempre passada por referência, diferente de uma variável qualquer
  - Tipos de Dados, Estruturas, Identificadores, ...

Problemas de Legibilidade em Pascal

```
if x>1 then
  if x=2 then
    x:=3
else
    x:=4
```

(a)

```
if x>1 then
  if x=2 then
    x:=3
  else
    x:=4
```

(b)

```
if x>1 then
  if x=2 then
    x:=3
  else
else
  x:=4
```

(c)

Solução do Problema em outras Linguagens

```
if x>1 then
  if x=2 then
    x:=3
  else
    x:=4
  fi
fi
```

Algol-68

```
if x>1 then
  if x=2 then
    x:=3
  else
    x:=4
  end
end
```

Modula-2

```
if x>1 then
  if x=2 then
    x:=3;
  else
    x:=4;
  end if;
end if;
```

Ada

Características Importantes

- Redigibilidade: facilidade de escrita de programas, a qual pode ser contrária à legibilidade
  - Simplicidade e Ortogonalidade
  - Suporte para Abstração (por ex., o uso de subrotinas)
  - Expressividade (operador ++ em C, for em comparação ao while em C e Java)
  - Reuso de código

Características Importantes

- Confiabilidade: a linguagem gera programas cuja execução reflete exatamente o que foi especificado
  - Verificação de Tipos
  - Manipulação de Exceções
  - Uso de sinônimos / apelidos (ponteiros)
  - Compatibilidade entre compiladores

Características Importantes

- Custo: análise do impacto na adoção da linguagem
  - Treinamento de programadores
  - Escrita de programas
  - Compilação destes
  - Execução destes
  - Infra-estrutura necessária
  - Confiabilidade

Outras Características também Importantes

- Eficiência do Programa Compilado
- Características do Processo de Compilação
- Disponibilidade de Ferramentas, Bibliotecas
- Portabilidade do Código Gerado
- Comunidade Ativa
- Atualização Frequente
- *Turbinada com recursos de IA! >8-)

Histórico

- Inicialmente, a programação de computadores se resumia a Assembly ou algo similar
- Na década de 50, diversas linguagens surgiram, as quais são as ancestrais das linguagens que usamos hoje em dia
  - Fortran
  - COBOL
  - LISP
  - Algol 60

Histórico

- Nas décadas de 60 e 70 surgiram outras que definiram a maioria dos diferentes paradigmas de programação que temos hoje em dia
  - C
  - Prolog
  - ML
  - Simula

Histórico

- Na década de 80 os projetistas de linguagens passaram a considerar requisitos de mais alto nível, como desempenho, programação em larga escala e estruturas mais inteligentes
  - C++
  - Ada
  - Eiffel
  - Perl

Histórico

- Na década de 90 entramos na era da internet, onde as linguagens passam a se importar com conexão à rede, integração com navegadores, etc (preocupações da década anterior ainda persistem)
  - Python
  - Java
  - Haskell
  - Ruby
  - PHP
  - C#
  - JavaScript

Histórico

- Anos 2000 em diante, linguagens multiparadigmáticas, especializadas
  - Go
  - Rust
  - Kotlin
  - Swift
  - TypeScript
  - Dart
  - Clojure
  - Elixir

Se Carros fossem Linguagens de Programação ...

- http://www.cs.caltech.edu/~mvanier/hacking/rants/cars.html

Razões para Estudo de Conceitos de Linguagens

- Aumento da capacidade de expressar idéias
- Aumento de *background* para escolha de linguagem apropriada
- Aumento de habilidade para aprender novas linguagens
- Melhor entendimento do significado de implementação
- Melhor uso de linguagens já conhecidas
- Maior entendimento da Computação como um todo

Tarefa

- Cada aluno deve trazer algum programa escrito em 1 linguagem diferente e explicar o seu funcionamento e as características gerais da linguagem