

Programação I

Licenciatura em Engenharia Informática

2015-2016

Vitor Beires Nogueira

Escola de Ciências e Tecnologia
Universidade de Évora

- Concepção de métodos para resolver problemas usando computadores
- Análise e comparação de métodos diferentes
- Combinação de várias competências de
 - ▶ **matemática:** linguagens formais para especificar ideias
 - ▶ **engenharia:** projectar, unir componentes para formar um sistema, avaliar prós/contras de alternativas
 - ▶ **ciências naturais** observar comportamento de sistemas complexos, tecer hipóteses, testar previsões

Linguagens formais

- As linguagens formais tem
 - ▶ **sintaxe:** regras "gramaticais"
 - ▶ **semântica:** significado ou acção associada
- Existem linguagens formais já conhecidas:
 - ▶ expressões aritméticas: $3 + 3 = 6$
 - ▶ estrutura molecular: H_2O
- **Exercício:** Qual a semântica dos exemplos acima? Ilustre casos de sintaxe incorrecta.

Definição (Linguagens de Programação)

Linguagens formais que foram concebidas para exprimir computações

Linguagens de Baixo Nível

■ Linguagem/Código máquina

- ▶ Exemplo: 01010101
- ▶ linguagem mais baixa e nativa dos computadores
- ▶ única linguagem directamente executável pelo computador
- ▶ *difícil* compreensão
- ▶ "Looking at a program written in machine language is vaguely comparable to looking at a DNA molecule atom by atom." (Hofstadter)

■ Assembly

- ▶ Exemplo: push, ...
- ▶ Utiliza mnemónicas (texto) para representar código máquina
- ▶ Existem programas (assembladores) para traduzir para código máquina
- ▶ Próximo da linguagem máquina

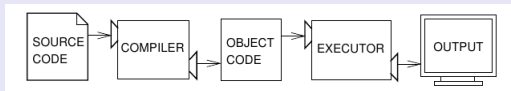
Linguagens de Alto Nível

- Exemplo: C, Java, Prolog, Python, ...
- mais próximas da formulação matemática dos problemas
- mais "fáceis"
- "portáteis"
- Traduzidas para código máquina por interpretadores ou compiladores

Interpretador



Compilador



Permitem diferentes níveis de abstracção

- Alto nível: mais próximo da formulação do problemas; facilita a programação, detecção e correcção de erros
- Baixo nível: mais próximo da maquina; possivelmente mais eficiente

Para problemas diferentes

- Cálculo numéricos: Fortran
- Raciocínio: Prolog
- Scripting: Perl, Python

Diferentes paradigmas

- imperativo: C, Pascal
- funcional: Haskell, Caml
- lógico: Prolog
- orientado a objectos: Java

- decompor um programa em pequenos módulos
- módulos podem ser re-utilizados
- podem-se testar e/ou provar que são correctos
- fáceis de modificar
- programas devem ser escritos para serem lidos por humanos!
Comentários, estrutura, nomes das “coisas”, ...
- correcto - simples - eficiente

Características

- Site principal: <http://python.org/>
- Alto nível
- Sintaxe simples
- Fácil aprendizagem
- Existe para a maioria dos sistemas operativos: Linux, Mac OS, M\$, ...
- É livre!
- Utilizada com sucesso em muitos locais:
<http://python.org/about/success/>

Interpretada ...ou quase

Interpretação “híbrida”:

- compilador traduz Python (ficheiros com extensão py) para um código intermédio “byte-code” (ficheiros com extensão pyc)
- execução é feita por um interpretador de “byte-code”

Vantagens/Desvantagens

- + Desenvolvimento rápido
- + Mais eficiente que um interpretador clássico
 - Menos eficiente do que linguagens compiladas (por exemplo C)

Interpretador de Python

- Em Linux temos o comando python
 - ▶ Interactivo: executa uma instrução de cada vez e mostra o resultado
 - ▶ Batch: executa todos os comandos existentes num ficheiro
- Existem muitos ambientes de desenvolvimento: IDLE, PyDev, Emacs, ...

Programa : "Hello, World" (versão 2.*)

```
print 'Hello , _World! '
```

Programa : "Hello, World" (versão 3.*)

```
print( 'Hello , _World! ')
```

- "Errare humanum est"
- A programação está sujeita a (muitos) erros
- Erros de programação são denominados de **bugs**
- O processo de encontrar os erros é denominado por **debugging**
- Tipos de erros:
 - ▶ sintático: Por exemplo

```
print 'Hello , World!
```
 - ▶ runtime: indicam que algo excepcional (e normalmente mau) aconteceu
 - ▶ semântico: Exemplo?
- Debugging como ciência experimental
 - ▶ Debugging é semelhante ao trabalho de um detective
 - ▶ Quando temos uma ideia sobre o que está errado no programa, alteramos e experimentamos.
 - ▶ "When you have eliminated the impossible, whatever remains, however improbable, must be the truth." (A. Conan Doyle, The Sign of Four)