

# Tema 7

## Concepção de Linguagens

### Cr terios de qualidade

Linguagens Formais e Aut matos, 2  semestre 2017-2018

Miguel Oliveira e Silva, DETI, Universidade de Aveiro

### Conte do

<b>1</b>	<b>Programa���</b>	<b>1</b>
<b>2</b>	<b>Qualidade de programas</b>	<b>1</b>
2.1	Cr�terios externos de qualidade . . . . .	2
2.2	Cr�terios internos de qualidade . . . . .	2
<b>3</b>	<b>Qualidade de linguagens</b>	<b>3</b>
3.1	Cr�terios de qualidade de linguagens . . . . .	3

### 1 Programa   

- O objectivo  ltimo da programa      resolver problemas com “m quinas”.
- Com esse fim, desenvolvem-se programas em linguagens que sejam, imediata ou ap s tradu    autom tica, execut veis nessas “m quinas”.
- Mas, numa perspectiva de engenharia, como podemos avalia    a qualidade desses programas?
- Isto  , o que   que distingue diferentes programas para o mesmo problema, fazendo com que uns sejam “melhores” do que outros?
- Estando os programas expressos em linguagens de programa   , ser  que podemos t mb m definir cr terios de qualidade para estas?
- Nesta  ltima semana, vamos brevemente abordar estes temas.

### 2 Qualidade de programas

- Os factores de qualidade de programas podem dividir-se em dois grupos: *factores externos* e *factores internos*.
- Os factores externos expressam as qualidades vis veis para os utilizadores externos de programas.
  - Por exemplo: a fiabilidade, a facilidade de utiliza    e o desempenho.
- Os factores internos referem-se  s qualidades vis veis apenas para os programadores.
  - Por exemplo: a modularidade e a legibilidade.
-   evidente que em rela    ao produto final, s  ir o interessar as suas qualidades externas.
  - Pouco importa se um programa de defesa militar   modular e de f cil compreens    se um erro na entrada acciona um m ssil.
- No entanto, apesar desta constata   , a chave para se obterem boas qualidades externas reside precisamente na qualidade dos factores internos.

## 2.1 Critérios externos de qualidade

**Correcção:** Correcção é a capacidade do programa efectuar as suas funções exactamente como definido nas suas especificações.

**Robustez:** Robustez é a capacidade dos programas funcionarem mesmo em situações anormais.

**Fiabilidade:** Fiabilidade é a capacidade de um programa ser correcto e robusto.

**Extensibilidade:** Extensibilidade expressa a facilidade com que os programas se adaptam a mudanças de especificações.

**Reutilização:** Reutilização é a capacidade dos programas serem utilizados em parte, ou na sua totalidade, para novas aplicações.

**Eficiência:** Eficiência expressa a capacidade de usar de uma forma óptima os recursos do hardware (CPU, memória, etc.).

**Verificabilidade:** Verificabilidade é a capacidade de facilmente elaborar procedimentos e dados de teste para detectar erros e falhas.

## 2.2 Critérios internos de qualidade

**Legibilidade:** A legibilidade expressa a facilidade com que se apreende e se compreende a estrutura e código de programas.

**Expressividade:** A linguagem deve expressar com clareza e simplicidade todas as abstracções e mecanismos de programação que pretende suportar.

### Modularidade

- Não é fácil uma definição precisa de modularidade.
- Podemos dizer que é uma forma particular da separação de interesses, em que a divisão do problema se faz por unidades (módulos) individualizadas e coerentes, com valor e significado por si só.

**Módulos:** Excertos de programas (blocos) independentes com os quais se podem construir novos programas.

- Dizemos que um bloco de um programa é modular se for:
  1. Facilmente *separável* de outros blocos;
  2. Facilmente *combinável* com outros blocos;
  3. Fácil de ser *compreendido* isoladamente;
  4. *Continuidade*: pequenas modificações num módulo apenas o afectam a ele ou eventualmente os seus clientes directos;
  5. *Auto-protégido*: dados internos protegidos contra utilizações abusivas.

## Modularidade: Vantagens

- Cada módulo pode ser desenvolvido, analisado e testado de forma *independente*:
  - cada módulo pode ser da responsabilidade de entidades (pessoas) distintas.
- Mais fácil de maximizar a *correção* e a robustez;
- *Reduz a complexidade* do programa global:
  - implementação de mecanismos de abstracção para facilitar tarefas.
- *Reutilização* de código:
  - ao desenvolvermos um módulo especializado numa tarefa/funcionalidade, podemos facilmente reutilizá-lo noutro programa com as mesmas necessidades.

## 3 Qualidade de linguagens

- As linguagens de programação são a mais importante das ferramentas para o desenvolvimento de programas.
- Muitos dos factores de qualidade – em particular, a correção – podem depender fortemente da linguagem de programação.
- Isto é, a maior (ou menor) facilidade no desenvolvimento de programas com qualidade depende em primeira linha das qualidades das linguagens de programação utilizadas.
- Apesar dessa reconhecida importância, são relativamente raras abordagens objectivas e sistemáticas ao problema da qualidade de linguagens.
- Vamos tentar fazer alguma sistematização também aqui.

### 3.1 Critérios de qualidade de linguagens

#### Critérios de qualidade de linguagens: Expressividade

No projecto de programas, o primeiro desafio essencial colocado a uma linguagem de programação consiste na facilidade com que a linguagem expressa os mecanismos e abstracções relevantes para o método (ou métodos) de programação que essa linguagem pretende suportar.

**Expressividade:** A linguagem deve expressar com clareza e simplicidade todas as abstracções e mecanismos de programação que pretende suportar.

#### Critérios de qualidade de linguagens: Abstracção

Desde o aparecimento das primeiras linguagens de programação, que a tendência tem sido de um distanciamento progressivo relativamente ao hardware do sistema computacional, e um aumento da abstracção das linguagens.

Há toda a vantagem em separar a forma como os programas são expressos e construídos, da forma como são realizados e implementados no sistema computacional.

Dito de outra forma, os programas devem ser explícitos quanto ao comportamento que deles se espera e não necessariamente à forma como esse comportamento é traduzido nas linguagens de baixo nível utilizadas pelas unidades de processamento dos computadores.

**Abstracção:** A semântica da linguagem deve ser expressa relativamente aos aspectos importantes dos seus mecanismos, e não aos detalhes de eventuais possíveis realizações.

### Critérios de qualidade de linguagens: Compreensibilidade

A documentação de programas é um dos aspectos que tende a ser menos considerado em linguagens de programação – aparte do suporte para o uso de comentários – levando a que os respectivos programas sejam difíceis de compreender, de depurar e de modificar.

A documentação deve ser encarada como sendo uma parte integral, não só processo de desenvolvimento de programas, mas também do próprio programa.

**Compreensibilidade:** A linguagem deve encorajar e facilitar a escrita de programas legíveis e auto-documentados.

Embora a facilidade na escrita e a facilidade na leitura de um programa não sejam dois objectivos antagónicos (antes pelo contrário), a legibilidade é muito mais importante do que a facilidade de escrita.

### Critérios de qualidade de linguagens: Segurança

No desenvolvimento de programas, a depuração tende a ser a fase mais demorada, difícil e menos motivadora para os programadores.

No entanto, o que dela resultar afecta directamente o critério de qualidade mais importante de todos: a correcção; pelo que se tornam extremamente importantes todas as ajudas que a linguagem possa dar nesse sentido. Essas ajudas podem existir em basicamente duas áreas: na detecção e na localização de erros.

O subconjunto de erros relativamente aos quais as linguagens de programação mais têm a obrigação de detectar, são os que se relacionam com utilizações incorrectas dos seus próprios mecanismos e respectivas abstrações.

**Segurança:** Uma linguagem diz-se segura se os seus mecanismos e abstrações não produzirem resultados sem sentido.

Outra definição interessante para o mesmo conceito é:

Uma linguagem diz-se segura se proteger as suas próprias abstrações.

Assim, um mecanismo será seguro se a sua utilização num programa só for aceite se houver a garantia de que nenhum resultado sem sentido advirá dessa sua utilização.

A segurança pode ser garantida antes dos programas serem executados (em tempo de compilação ou estaticamente), ou testada enquanto estes são executados (em tempo de execução ou dinamicamente).

Se possível, uma aproximação estática é desejável já que garante a inexistência de determinados erros em tempo de execução.

A opção de projecto de linguagens mais importante em termos de garantir a segurança das linguagens tem a ver com o chamado sistema de tipos da linguagem.

### Critérios de qualidade de linguagens: Sinergia

Um aspecto de qualidade de linguagens pouco referido (mas com certeza facilmente reconhecido), é não só o grau de integração e coesão dos vários mecanismos da linguagem entre si, mas também quando desse facto resultam mais valias com o aparecimento de novas funcionalidades, emergentes do uso conjunto desses mecanismos.

Ou seja, essa propriedade avalia a possibilidade das funcionalidades do conjunto de determinados mecanismos serem mais do que a soma individual das funcionalidades dos mecanismos envolvidos.

**Sinergia:** Quando possível, os mecanismos e abstrações das linguagens devem ser construídos por forma a que, quando utilizados em conjunto, gerem novas funcionalidades desde que estas sejam consentâneas com as respectivas semânticas individuais.

### Critérios de qualidade de linguagens: Ortogonalidade

Levando mais longe esta perspectiva de se analisar as propriedades resultantes do uso conjunto de mecanismos, temos que quando o todo tem um valor (em termos de funcionalidades) inferior à soma das partes (cada um dos mecanismos vistos isoladamente), estamos provavelmente na presença de problemas de segurança; quando esse valor for superior à soma das partes, temos qualidades sinérgicas; e quando for igual, estamos na presença de mecanismos independentes ou ortogonais.

Assim, sendo que a segurança deve ser sempre garantida, temos apenas duas opções no funcionamento conjunto de mecanismos: ou devem ser sinérgicos ou ortogonais.

**Ortogonalidade:** Os mecanismos e abstrações da linguagem são ortogonais, se funcionarem de uma forma independente.

### Critérios de qualidade de linguagens: Outros critérios

**Realizabilidade:** Um mecanismo de uma linguagem de programação será realizável se existir pelo menos uma implementação, computável no sistema de compilação, que permita a geração do código executável apropriado no sistema de suporte à execução do programa.

**Eficiência de programas:** A linguagem deve permitir que o respectivo sistema de compilação (ou, se for o caso, de interpretação) tenha a possibilidade de gerar programas eficientes.

Existe ainda outro aspecto de eficiência aplicável às linguagens de programação: a eficiência na compilação.

**Eficiência de compilação:** A linguagem deve permitir uma compilação eficiente.

**Extensibilidade da linguagem:** Extensibilidade de linguagens de programação expressa a facilidade com que se lhe adicionam novos mecanismos.

**Unicidade:** As linguagens de programação devem fornecer uma boa forma de expressar cada operação de interesse; devem evitar fornecer duas.

**Consistência:** As linguagens de programação devem assentar num conjunto pequeno de ideias fundamentais e completas, devendo depois as realizar consistentemente até às últimas consequências.

