Linguagens de Domínio Específico

Fabio Mascarenhas – 2017.1

http://www.dcc.ufrj.br/~fabiom/dsl

Tipos

- Um *tipo* é:
  - Um conjunto de valores
  - Um conjunto de operações sobre esses valores
- Os tipos de uma linguagem podem ser pré-definidos, mas normalmente as linguagens também permitem que o programador defina seus tipos
- Os tipos de uma linguagem podem formar sua própria mini-linguagem, com sua sintaxe e regras de escopo

Sistema de Tipos

- O *sistema de tipos* de uma linguagem especifica a *sintaxe* dos tipos, e quais operações são válidas nesses tipos
- O processador da linguagem usa as regras do sistema de tipos para fazer a *verificação de tipos* do programa
- O objetivo é rejeitar programas que contêm operações inválidas
- Várias linguagens adiam essa verificação até o momento em que o programa está executando – sistemas de tipos estáticos flexíveis podem ser bem complexos!

Compatibilidade entre tipos

- A regra mais simples de compatibilidade entre tipos é a *igualdade*: dois tipos são compatíveis se são idênticos
- O sistema de tipos também pode ter regras de *coerção*: tipos que são incompatíveis, mas que podem ser convertidos explicitamente (com uma operação de *cast*) ou implicitamente (inserção automática de *casts*)
- Como tipos numéricos em Java, ou tipos primitivos de Java com suas respectivas classes
- Finalmente, a compatibilidade de tipos pode ser dada através de uma *relação de subtipagem*

Subtipagem

- O conjunto de valores de um tipo pode ser um subconjunto do conjunto de valores de outro tipo

- Podemos querer expressar isso no sistema de tipos através de uma *relação de subtipagem* ≤

- Em uma linguagem com subtipagem nominal essa relação é declarada pelo programador; em Java ela é dada pelas cláusulas *extends* e *implements*

- A relação de subtipagem é *simétrica* (t ≤ t) e *transitiva* (r ≤ s e s ≤ t implica r ≤ t)

Tipagem nominal vs estrutural

- Se um tipo é identificado unicamente pelo seu nome, independente de quaisquer outros atributos, esse tipo é *nominal*
- Um exemplo são todos os tipos de Java exceto vetores e tipos genéricos
- Se um tipo é identificado unicamente pela sua estrutura, independente de ter um nome associado, ele é *estrutural*
- Tipos nominais são mais simples de entender e implementar, tipos estruturais são mais flexíveis