

Compiladores

Linguagem *TabLan*

Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro

Abril de 2023

Objectivos

O objectivo geral deste trabalho é o desenvolvimento de uma linguagem de programação compilada – i.e. que crie programas equivalentes ao programa a compilar numa linguagem de programação genérica (Java, C++, Python, ...) – para manipulação avançada de tabelas.

Nesta linguagem, as tabelas são definidas como tipos de dados (um pouco à imagem de estruturas da linguagem C, mas em que cada tabela pode ter um qualquer número de linhas). Na versão obrigatória da linguagem, para além das tabelas, vão existir os tipos de dados elementares inteiro (`integer`), real (`real`) e texto (`text`).

Características da solução

Apresentam-se a seguir um conjunto de características que a solução desenvolvida pode ou deve contemplar. Essas características estão classificadas a 3 níveis:

- mínima – característica que a solução tem obrigatoriamente que implementar;
- desejável – característica não obrigatória, mas fortemente desejável que seja implementada pela solução (apenas considerada se as mínimas forem cumpridas);
- avançada – característica adicional apenas considerada para avaliação se as obrigatórias e as desejáveis tiverem sido contempladas na solução.

Características mínimas

Os exemplos `example1.da`, `example2.da` e `example3.da` indicam algum código fonte que tem de ser aceite (e devidamente compilado) pela linguagem a desenvolver.

A linguagem deve implementar:

- Instrução para definir um novo tipo de dados tabela. Nesta definição de um tipo de dados para tabela, as colunas podem ser independentes, ou dependentes. Neste último caso a definição usa uma expressão aritmética para a respectiva definição (o que faz com que o seu valor seja automaticamente definido uma vez definidos os valores dos quais depende).
- Os tipo de de dados inteiro, real e texto.
- Aceitar expressões aritméticas standard para os tipos de dados numéricos. Aceita a operação de concatenação de texto (operador da soma).
- Instrução de escrita no *standard output* (com e sem mudança de linha no fim).
- Instrução de leitura de texto a partir do *standard input*.
- Operadores de conversão entre tipos de dados (por exemplo, `text(10)` para converter para texto; ou `integer("10")` para converter para inteiro).
- Instrução para adicionar linha no fim de uma tabela.
- Instrução para adicionar cabeçalho à tabela ou a cada coluna da tabela.
- Instrução de iteração (loop) das linha de tabelas.
- Verificação semântica do sistema de tipos.

Características desejáveis

Os exemplos `example4.da`, `example5.da` e `example6.da` indicam algum código fonte que se enquadra nas características desejáveis ou avançadas.

- Permitir a definição de expressões booleanas (predicados) contendo, pelo menos relações de ordem e operadores booleanos (conjunção, disjunção, etc.).
- Incluir a instrução condicional (operando sobre expressões booleanas).
- Incluir outras instrução repetitivas (operando sobre expressões booleanas).
- Instrução remoção de linha de tabelas optativamente sujeita a um predicado (isto é, a uma condição booleana).
- Definição de novos tipos de dados assentes em tipos primitivos (integer, real ou text) com a definição de um invariante.
- Definição de tabelas extendendo tabelas existentes (uma variante simplificada de herança).

Características avançadas

- Implementar funções e variáveis locais às mesmas.
- Permitir que a instrução `read` preencha tabelas dentro de tabelas (ver exemplo `example6.t1`).
- ...