

Licenciatura em Engenharia Informática – DEI/ISEP
Linguagens e Programação 2021/2022
Aula Prática-Laboratorial

Ficha PL 4

ANTLR

Objetivos:

- Implementação de analisadores sintáticos com ANTLR

2. Exercícios Propostos

- 1) Crie o programa “Hello World” com o ANTLR. Os tokens existentes são HELLO e WORLD. Sempre que o texto a analisar tiver os dois tokens pela ordem certa, deve imprimir a frase “Hello World!!!”.

- 2) Crie um analisador usando o ANTLR, que reconheça frases constituídas por dois inteiros separados por um operador relacional (=, <, >, <=, >=, <>). O analisador deve indicar se a frase está de acordo com a sintaxe, e se a comparação é verdadeira ou falsa.

10 <= 20 '\n' - verdadeiro

5 = 10 '\n' -falso

120 <> 130 '\n' - verdadeiro

> 10 '\n' - erro de sintaxe

- 3) Reescreva a gramática da alínea anterior, de modo a:

i. aceitar inteiros e letras (a-z e A-Z);

ii. aceitar várias comparações na mesma linha;

iii. testar a incompatibilidade de tipos entre inteiros e letras;

iv. fazer a recuperação dos erros ocorridos.

10 <= 20 = 20 '\n' - verdadeiro verdadeiro

5 < 10 >= 5 < 2 '\n' - verdadeiro verdadeiro falso

120 <> A '\n' - incompatível

z <> A '\n' - verdadeiro

> 10 '\n' - erro de sintaxe

- 4) Implemente um analisador sintático para reconhecimento duma expressão aritmética, utilizando o ANTLR. A gramática é a seguinte:

$$S \rightarrow ID \mid '=' E \mid E$$

$$E \rightarrow E \mid '+' E \mid E \mid '-' E \mid E \mid '*' E \mid E \mid '/' E \mid '(' E ')' \mid ID \mid INT \mid REAL$$

Em que ID é um identificador (letra de 'a' a 'z'), INT um número inteiro e REAL um número real.

O *parser* deve analisar múltiplas expressões e obter resultados, apresentando-os. Sempre que haja uma atribuição esse valor deve ser guardado, para ser utilizado com o identificador respetivo em outras expressões. Como tabela de símbolos, utilize um vetor com uma posição para cada letra.

- i. Implemente este exercício sem usar precedências de operadores;
 - ii. Implemente este exercício usando precedências de operadores.
- 5) Considere um simulador de uma máquina de venda automática que dispõe de um conjunto de produtos e aceita moedas em euros (€0.01, €0.02, €0.05, €0.10, €0.20, €0.50, €1.00, €2.00). O objetivo é selecionar um produto, introduzir o respetivo valor, receber o troco (se existir) e receber o produto. Considere os seguintes produtos: café (€0.35), pingó (€0.35), chá (€0.35), chocolate (€0.40), copo (€0.05) e leite (€0.30).

O formato de entrada de dados deve obedecer à seguinte regra:

<produto>, <moeda1>, ... <moedan>.

O formato de saída deve obedecer à seguinte regra:

<produto>, <moeda1>, ... <moedan> | "dinheiro insuficiente".

Exemplo:

Entrada - café, €0.01, €0.10, €0.05, €0.20

Saída - café, €0.01

Defina a gramática de modo a que a máquina funcione ininterruptamente e implemente-a utilizando o ANTLR.

- 6) Suponha um simulador para cálculo das notas finais da disciplina de Linguagens de Programação. Pretende-se verificar, apenas, se um determinado aluno obteve aproveitamento ou não. Assim, o simulador deve estar constantemente a receber dados traduzidos por frases do tipo:

*<tipo> (<exame época normal> | <exame recurso>
| <exame Setembro>) <trabalho prático>
<nº aluno> <turma>*

Os formatos de cada um dos campos é o seguinte:

<nº aluno> - 7 algarismos

<tipo> - N ou D conforme o aluno esteja inscrito no regime noturno ou diurno

<turma> - 1 algarismo e 2 letras

<exame época normal> - $0 \leq \text{inteiro} \leq 20$

<exame recurso> - $0 \leq \text{inteiro} \leq 20$

<exame Setembro> - $0 \leq \text{inteiro} \leq 20$

<trabalho prático> - $0 \leq \text{inteiro} \leq 20$

Considere definida a função procura(N) que procura num ficheiro de alunos o nome do aluno com o número N e retorna uma cadeia com 60 caracteres contendo o nome do aluno.

Pretende-se que o simulador retorne informação sobre o aproveitamento do aluno, indicando todos os dados do mesmo (incluindo o nome), bem como a classificação final da disciplina (CF), obtida de acordo com a seguinte fórmula (NFREQ é a nota do trabalho prático e PE é a nota do exame):

$$CF = \frac{xNFREQ + yPE}{x + y} \begin{cases} x = 45\% \\ y = 55\% \\ \min NFREQ = 8.0 \\ \min PE = 8.0 \end{cases}$$

Pode utilizar o formato de saída que considerar mais adequado, tendo em atenção que se não forem atingidas as notas mínimas indicadas a classificação final deverá ser SM.

Defina a gramática de modo a que a máquina funcione ininterruptamente e implemente-a utilizando o ANTLR.

Exercícios complementares

1. O gabinete de e-learning fazOSCursos, pretende realizar a análise da quantidade de alunos e tipo de cursos lecionados por cada formador. Num ficheiro está reunida a informação de todos os cursos lecionados. O registo de cada formador é constituído por uma linha com a informação do formador (código e nome), seguido de linhas com a informação dos cursos lecionados (código, nome, ano curricular, alunos inscritos e carga horária). A informação está registada da seguinte forma, no formato EBNF (campos entre [] são opcionais e os campos entre { } repetem-se 0 ou mais vezes):

```
<cod_formador> [<nome_formador>] '\n'
{<cod_curso> [<nome_curso>] <ano_curricular> <alunos_inscritos> <carga_horaria> '\n'}
```

Em que o conteúdo de cada campo é o seguinte:

- <cod_formador> – Sigla com 3 letras minúsculas seguido de 2 algarismos
- <nome_formador> – String entre aspas
- <cod_curso> – Sigla com 3 letras maiúsculas seguida de 2 algarismos
- <ano_curricular> – Inteiro maior que 0 e menor que 10
- <nome_curso> – String entre aspas
- <alunos_inscritos> – Inteiro maior que 0
- <carga_horaria> – Inteiro maior que 0 e menor que 20

Defina a gramática para o ficheiro anteriormente descrito, e crie utilizando o ANTLR um programa que:

1. Reconheça a validade do ficheiro;
2. Indique para cada um dos formadores, o código do formador, o código do curso e o número de alunos inscritos, com o maior número de alunos;

3. A quantidade de alunos de cada formador;
4. Gere um erro sempre que encontre um erro sintático/léxico no ficheiro.

O programa só deve parar no final do ficheiro, independentemente dos erros encontrados. O programa deve validar o ficheiro de dados e imprimir os resultados como no seguinte exemplo:

Input:	Output:
ans23 "António Silva"	ans23 MAT33 445 aluno(s)
AAA32 3 125 12	Total: 582 aluno(s)
MAT33 "Matemática I" 1 445 8	Erro: registo incompleto
ANF42 "Análise Financeira" 5 12 4	abc34 NHL96 178 aluno(s)
abc34	Total: 255 alunos
ABC33 2 55 12	
XYZ "Materiais" 125	
NHL96 "Circuitos Elétricos" 2 178 9	
FMI69 "Teoria de Empréstimos" 5 22 3	

2. Crie e defina uma gramática que reconheça endereços URL e implemente-a utilizando ANTLR. Para além disso, o reconhecimento de um endereço deve ser seguido pela indicação dos seus componentes.

Exemplo para: `http://www.dei.isep.ipp.pt/nova/index.html`, a resposta deverá ser:

endereço válido

protocolo: http

máquina: www.dei.isep.ipp.pt

caminho: nova

página: index.html

Exemplo 2: para `mailto:dei@isep.ipp.pt`, a resposta deverá ser:

endereço válido

protocolo: mailto

utilizador: dei

domínio: isep.ipp.pt

Considere como válidos os seguintes protocolos: `http`; `https`; `ftp`; `ftps`; `mailto`. Lembre-se que quer o caminho, quer a página podem não existir, caso em que a sua indicação deve ser ignorada e que a máquina pode ser substituída pelo respectivo endereço IP. Considere ainda que os nomes das máquinas e dos ficheiros apenas são constituídos pelos caracteres de *a* a *Z*.