

Gerador de Questionários

Departamento de Eletrónica, Telecomunicações e
Informática

Universidade de Aveiro

Linguagens Formais e Autómatos

(85088) Francisco Martinho, (88930) João
Simões, (93088) Eduardo Cruz, (95228) Lara
Matos, (98003) João Felisberto



Conteúdo

1	Resumo	1
2	Introdução	2
3	Linguagem auxiliar <i>CQuiz</i>	4
4	Linguagem principal <i>Quiz</i>	5
4.1	Intruções	5
4.2	Tipos	5
4.3	Comentários	5
4.4	Palavras reservadas	6
4.5	Operações Aritméticas	7
4.6	Instruções Condicionais	7
4.7	Ciclos	7
4.7.1	While e Do...While	8
4.7.2	For	8
5	Palavras Reservadas	9
6	Conclusão e Resultados	10
7	Contribuição dos Autores	11

Capítulo 1

Resumo

No âmbito da Unidade Curricular de Linguagens Formais e Autómatos, foi proposto o desenvolvimento de um projeto a este grupo de alunos. Esse projeto consiste num Gerador de Questionários.

Neste relatório, serão abordados itens relacionados com o trabalho, tais como os objetivos com este trabalho apresentados no Capítulo 2, a linguagem para a Base de Dados (*CQuiz*), a linguagem para o Desenvolvimento de Questionários (*Quiz*), e as conclusões tiradas com o desenvolvimento deste projeto.

Na pasta designada de "manual" contém um manual de instruções com todas as indicações a ter em consideração para o bom funcionamento por parte do utilizador.

Capítulo 2

Introdução

No nosso dia a dia, é muito frequente depararmo-nos com avaliações e desafios. Para isso, é usual a utilização de questionários para a avaliação de conhecimentos, sendo essas questões de vários tipos, tais como de escolha múltipla, correspondência, resposta direta, entre outros.

Os objetivos deste projeto são:

- Definição de uma linguagem principal para a construção de questionários interativos;
- Contrução de um compilador para transformar a linguagem num programa de linguagem genérica;
- Definição de uma linguagem auxiliar para a construção de bancos de perguntas;
- Extração dos bancos de perguntas através de linguagem principal.

Para iniciar este projeto, surge a linguagem *Quiz* onde se processa toda a estrutura do questionário, com semelhanças a linguagens de alto nível e ferramentas adicionais para desenvolvimento de questionários interativos.

Quiz apresenta a possibilidade de:

- Definir variáveis de diferentes tipos;
- Utilizar ciclos e intruções de condição;
- Conter instruções para a apresentação de questionários e respetiva modulação;
- Processar respostas e sua respetiva recolha das mesmas para análise.

Quando o ficheiro *Quiz.g4* é compilado, usando os comandos de **ANTLR4**, cria-se um ficheiro com a linguagem de destino. Neste caso, será em **Java**.

Sabendo agora da existência da liguagem de desenvolvimento de questionários, é necessária a criação de uma linguagem auxiliar *CQuiz.g4* que serve de

base de dados, ou seja, guarda as perguntas para posterior utilização nos questionários. Esta linguagem é muito mais simples do que a *Quiz*, permitindo assim ao programador descrever ficheiros de perguntas que são carregadas para uma base de dados que tanto pode ser ou não utilizada para o desenvolvimento do(s) questionário(s) que queremos.

Para um melhor entendimento do que está descrito acima, seguem-se capítulos que irão descrever detalhadamente as linguagens *Quiz*, presente no Capítulo 4, e *CQuiz*, no Capítulo 3, com respetivos exemplos. Posteriormente, está também apresentado um Manual de Utilização na pasta "manual", com instruções de compilação dos programas desenvolvidos e como é que cada um deverá correr.

Capítulo 3

Linguagem auxiliar *CQuiz*

Foi criada uma linguagem simples *CQuiz* para gerar uma base de dados de perguntas que o utilizador pode usufruir. Esta linguagem tem como objetivo a descrição de ficheiros que contêm as perguntas que podem ser usadas num determinado questionário.

Cada pergunta é constituída pelas características presentes na seguinte tabela:

ID	identificador (índice da pergunta)
grupos	genérico (tema, dificuldade, ...)
pergunta	string (texto da pergunta)
resposta	string (texto da resposta)
dependências	só se pode responder a uma pergunta se tiver feito as anteriores
<i>validator</i>	valida a resposta

É de salientar que, em cada tema, não é permitido que haja IDs iguais nem perguntas iguais, da mesma forma como também não é permitido respostas iguais numa mesma pergunta.

Assim, ficou mais perceptível como é que cada pergunta é constituída. No seguimento deste raciocínio, a base de dados das perguntas será organizada numa Lista com todas as perguntas que podem ser usadas num determinado questionário originado pelo utilizador.

Exemplo:

```
ID: question {
    [group1, group2, group3, ...];
    [depend1, depend2, ...];
    <multiple|match|numeric|short|long>;
    <possible answers>
    <correct>
}
```

Capítulo 4

Linguagem principal *Quiz*

4.1 Instruções

Em **Quiz** tanto pode haver instruções em bloco como instruções de linha. Nestas últimas, as instruções são sempre terminadas com o carácter ';'.

4.2 Tipos

A linguagem **Quiz** apresenta os seguintes tipos:

- int: representação de um valor inteiro; equivalente ao Integer na linguagem de destino.
- string: representação de strings; equivalente ao String na linguagem de destino;
- question: representação de questões da classe Question;
- number: representação de um número;
- text: representação de texto;
- list: representação de uma estrutura de dados List na linguagem de destino;

4.3 Comentários

Na linguagem *Quiz*, é considerada uma linha em comentário sempre que, no início dessa mesma linha, estiver presente o carácter '#'.

4.4 Palavras reservadas

Aqui estão as palavras reservadas:

- main
- stat
- qdb
- question
- varDecl
- mcall
- qans
- qcorrect
- propertyAccess
- fluxCtrl
- if
- elsif
- do
- while
- for
- foreach
- loopStat
- exp
- list
- tuple
- lstElem
- listIndex
- questionType
- int
- string

4.5 Operações Aritméticas

Na tabela seguinte está o conjunto de todas as operações que são possíveis nesta linguagem:

Símbolo	Função	Tipo
+	Adição ou Sinal Positivo	int
-	Subtração ou Sinal Negativo	int
/	Divisão	int
%	Resto da Divisão Inteira	int
*	Multiplicação	int
==	Igualdade	int, string
!=	Diferença	int, string
<	Menor	int
>	Maior	int
<=	Menor ou Igual	int
>=	Maior ou Igual	int

Para as estruturas condicionais, temos as palavras reservadas **and** e **or** que são, respetivamente, a conjunção e a disjunção lógicas.

4.6 Instruções Condicionais

Na linguagem **Quiz** existe a permissão de utilização de instruções condicionais, que alteram o fluxo do programa a desenvolver, escolhendo assim outros caminhos para o final pretendido.

Para isso, é muito usado um tipo de estrutura com a seguinte sintaxe:

```
if (condition) {...}
else if (condition) {...}
else {...}
```

Quando o programador está a escrever um programa, mais especificamente uma instrução condicional, não pode aparecer mais do que um **if** por cada instrução. É esta a palavra que expressa o início de uma instrução condicional. No entanto, pode-se colocar tantos **if** quantos aqueles que o programador assim o desejar. Quando chegar à última condição desse conjunto de instruções condicionais, terá que ser usado um **else**. A cadeia de instruções condicionais tem que ser expressa obrigatoriamente sempre por esta ordem aqui descrita.

Dentro de cada **condition** do **if** ou **else if**, pode existir uma ou mais condições. Neste último caso, podem ser usadas as palavras reservadas **and** e/ou **or**, que são, respetivamente, a conjunção e a disjunção lógicas.

4.7 Ciclos

4.7.1 While e Do...While

Para usufruir deste tipo de estrutura condicional, temos que escrever a seguinte sintaxe:

```
while (condition) {...}  
OU  
do {...} while (condition)
```

No primeiro caso, a condição do ciclo **while** é definida logo no início desse mesmo ciclo e é verificada sempre que se chega ao final do ciclo.

No caso seguinte, a condição é verificada apenas no final do ciclo.

Para que fique claro, caso entrarmos dentro de um dos ciclos, vai significar que estamos dentro de um novo contexto, isto é, todas as variáveis que forem criadas dentro desse mesmo ciclo, nunca teremos acesso a elas fora do ciclo.

4.7.2 For

Para usufruir das capacidades do ciclo **for**, temos duas possibilidades:

```
for(num i=0; i<10; i++) {...}  
OU  
for each v in l {...}
```

Na primeira sintaxe, é definida uma variável auxiliar para percorrer todos os valores do intervalo que está indicado. Caso a variável não sofra alterações como, por exemplo, uma instrução condicional, podemos presumir que essa mesma variável percorre todos os valores do intervalo sem qualquer problema que a impeça disso.

É possível uma equivalência entre um intervalo escrito entre parêntesis e a sua semelhança quando o mesmo intervalo é escrito em linguagem destino que, no nosso caso, é em **Java**. Um exemplo é o seguinte:

Configuração entre Parêntesis	Equivalência em Java
[0, 10]	(num i=0; i<=10, i++)

Capítulo 5

Palavras Reservadas

As palavras reservadas da linguagem *Quiz* são:

- if
- else
- do
- while
- for
- each
- foreach
- list
- number
- text
- question

Capítulo 6

Conclusão e Resultados

Através dos conhecimentos que adquirimos ao longo deste exaustivo semestre, foi possível elaborar este projeto e distribuir de um modo mais ou menos equilibrado as tarefas que competia cada elemento deste grupo realizar. As aulas práticas e teóricas foram a principal ajuda para adquirir os conhecimentos necessários, bem como toda a pesquisa envolvida.

De uma maneira generalizada, os objetivos foram bem sucedidos, tendo sido concretizado o objetivo principal deste projeto: a realização de questionários interativos através de uma linguagem criada por nós, alunos / futuros engenheiros, para que seja usada por qualquer utilizador.

Um agradecimento aos nossos professores, Miguel Oliveira e Silva e Artur Pereira, pela disponibilidade e por nos ter tirado dúvidas pertinentes durante as aulas práticas presenciais.

Capítulo 7

Contribuição dos Autores