



Universidade de Évora

Disciplina de Programação II



Autores:

Filipe Alfaiate nº43315

Vasco Barnabé nº42819

Professores:

Salvador Abreu

Francisco Coelho

Índice

ÍNDICE	2
2 - DESCRIÇÃO DO TRABALHO	4
2.1 – COMPILER.....	4
2.2 – COMPILERRULE.....	4
2.3 – FORWARD / LEAP / PENUP / PENDOWN / RESTORE / SAVE / TURN	4
2.4 – RULE	5
2.5 – TARTARUGA	5
2.6 – TURTELPOSITION	5
2.7 – TURTELSTATEMENT	5
2.8 – PANEL	5
2.8.1 – <i>UI</i>	5
2.8.2 – <i>uiCompilacaoRulePanel</i>	6
2.8.3 – <i>uiInicioPanel</i>	6
2.8.4 – <i>uilteracoesPanel</i>	6
2.8.5 – <i>uiRulePanel</i>	6
2.9 – WORDGENERATOR.....	6
2.10 – MAIN.....	6
2.11 – INTERFACES.....	7
2.11.1 – <i>Interpreter</i>	7
2.11.2 – <i>LSystem</i>	7
3 – CONCLUSÃO	8



1 - Introdução

Neste trabalho tivemos como objetivo implementar um conjunto de classes Java que permita definir Sistemas-L, gerar palavras por esses sistemas e visualizar graficamente essas palavras.

Foi criado um menu inicial onde o utilizador tens 2 opções: escolher uma palavra já predefinida e visualizá-la, de imediato, graficamente, ou então pode criar as suas próprias regras e construir palavras segundo a sua “vontade”.

2 - Descrição do trabalho

Inicialmente quando o trabalho estava a ser estruturado entendeu-se que o tipo de Tartaruga utilizada deveria ser o mesmo utilizado nas aulas (Tartaruga da pasta “galapagos”), apesar de existirem outras.

Não foi fácil, a princípio, entender à primeira o rumo do trabalho, pois parecia um pouco confuso. Começámos então por tentar entender passo a passo, percebendo o que era pretendido em cada algoritmo fornecido no enunciado, e até mesmo que classes seriam necessárias implementar.

2.1 – Compiler

Nesta classe é onde cada letra/símbolo é substituída(o) pela instrução correspondente.

2.2 – CompilerRule

Nesta classe cria-se o construtor para a letra e a palavra correspondente. É ainda nesta classe onde estão os *getters* e os *setters* destas duas variáveis (letra = *letter*, palavra = *statement*).

2.3 – Forward / Leap / PenUp / PenDown / Restore / Save / Turn

Cada letra ou cada símbolo teria que ter uma ação associada, portanto foram implementadas as classes *Forward*, *Turn*, *PenUp*, *PenDown*, *Leap*, *Save* e *Restore*, que estendem da *TurtleStatement*. A classe *TurtleStatement* serve para representar instruções dos Programas Tartaruga e contem o método *run*, que serve para ligar a instrução a um interpretador. Assim todas as classes que estendem da *TurtleStatement* terão o método *run* para ligar a instrução correspondente ao interpreter.

A classe *Forward* dá instruções à tartaruga para esta se movimentar de forma retilínea segundo a sua orientação atual.

A classe *Turn* dá instruções à tartaruga para rotacionar segundo o ângulo que lhe é concedido.

A classe *PenUP* dá instruções à “caneta” da tartaruga para levantar, ou seja, não desenha.

A classe *PenDown* dá instruções à “caneta” da tartaruga para baixar, ou seja, começa a desenhá-la nessa posição.

A classe *Leap* dá instruções à tartaruga para saltar para uma posição, usando, por esta ordem, as classes, *PenUp*, *Forward* e *PenDown*.

A classe *Save* dá instruções para gravar a posição atual da tartaruga (x,y).

A classe *Restore* dá instruções para devolver os valores da posição salvas no *Save*.

2.4 – Rule

Nesta classe cria-se o construtor para a letra inicial e para a regra associada. É ainda nesta classe onde estão os *getters* e os *setters* destas duas variáveis (letra inicial = *startLetter*, regra = *rule*).

2.5 – Tartaruga

É nesta classe que, ao fim ao cabo, são feitas todas as alterações da tartaruga, como a velocidade, a cor, o ajustamento da origem do referencial e onde todas as instruções estão referidas para quando for criada um novo objeto da classe *Tartaruga* esta possa ter acesso a todas as instruções.

2.6 – TurtlePosition

Esta classe é onde se encontra o construtor que permite obter a posição atual da tartaruga e a sua orientação. É ainda nesta classe onde estão os *getters* e os *setters* das três variáveis (abscissa da tartaruga = *x*; ordenada = *y*; orientação = *angle*)

2.7 – TurtleStatement

Esta classe serve para representar instruções dos *Programas Tartaruga* e o método *run* serve para ligar a instrução a um interpretador.

2.8 – Panel

Dentro deste tópico estão as classes relacionadas com a parte gráfica do trabalho.

2.8.1 – UI

Nesta classe, que estende da classe *JFrame*, é onde se cria todos os menus e submenus do menu principal. É inserido o título da janela (menu principal), o título do submenu onde se encontram todos os desenhos pré-definidos, as divisões entre alguns destes desenhos pré-definidos. Além disso inicializa e adiciona os campos das regras (coloca-os também no *UserInterface*) e também das regras de compilação (colocando também estas no *UserInterface*). Por último, cria-se o botão *Executar* que aplica todas as regras inseridas pelo utilizador, consequentemente, cria o desenho correspondente, e o botão *Limpar* que clareia todos os sectores caso seja essa a vontade do utilizador.

2.8.2 – uiCompilacaoRulePanel

Nesta classe ajusta-se grande parte do Layout desta parte do menu, das regras de compilação, onde tem de ser inseridos todas as instruções que o utilizador pode atribuir as letras e/ou aos símbolos. São também recolhidas as informações relativamente as regras que o utilizador escolheu. É inserida ainda uma regra que faz com que se utilizador não inserir nenhum valor nas caixas da direita, este é assumido como 0 (zero). Além disso quando o botão *Limpar* é selecionado, clareira o que foi escrito, cajo o utilizador assim o desejar.

2.8.3 – uiInicioPanel

Nesta classe cria-se a caixa onde vai ser inserido no menu principal a expressão inicial, optem-se o texto inserido pelo utilizador, e ainda limpa a caixa quando o botão *Limpar* é selecionado.

2.8.4 – uilteracoesPanel

Nesta classe cria-se a caixa onde vai ser inserido o número de iterações que o utilizador pretende que a tartaruga desenhe. À semelhança da classe anterior, obtém-se o numero de iterações inseridas e ainda limpa a caixa quando o botão *Limpar* é selecionado.

2.8.5 – uiRulePanel

Nesta classe cria-se conjuntos de caixas em paralelo, onde, nas caixas da esquerda, o utilizador insere uma das letras da expressão inicial, e nas caixas da direita, o utilizador insere a palavra em que essa letra se vai transformar.

2.9 – WordGenerator

Nesta classe é onde as letras são substituídas pelas palavras correspondentes.

2.10 – Main

Na classe principal é criada uma interface gráfica, do tipo UI (classe esta criada e modificada pelo programador). É aqui que estão definidas as regras e criadas as tartarugas para cada um dos desenhos pré-definidos. Além disso foram criadas exceções, para o caso de o utilizador não preencher certos campos obrigatórios e consequentemente aparecer a mensagem de aviso para que estes sejam preenchidos.

2.11 – Interfaces

Neste tópico estão englobadas as interfaces *Interpreter* e *LSystem*.

2.11.1 – Interpreter

Esta interface define interpretadores de PTs. São especificados métodos (todos com o nome *run*) para ligar cada interpretador à instrução a ser executada.

2.11.2 - LSystem

Esta interface define Sistemas-L. Representa estes Sistemas e gere palavras dos mesmos.

3 – Conclusão

Neste trabalho foi possível alargarmos o nosso conhecimento sobre esta linguagem de programação, *JAVA*, e mais especificamente sobre *Programas Tartaruga*.

Inicialmente a nossa maior dificuldade foi interpretar corretamente o enunciado, pois este era muito extenso e um pouco confuso, a nosso ver. Uma vez que era necessário criar muitas classes e cada uma delas necessitava das outras (a maioria), foi complicado manter o raciocínio ao objetivo final deste trabalho, que era criar um menu gráfico onde o utilizador pudesse escolher desenhos pré-definidos ou criar os seus próprios desenhos segundo as suas próprias regras.

Contudo, todos os obstáculos foram superados, terminando assim o que foi pedido. Com o desenvolvimento do nosso conhecimento nas linguagens de programação, tornamo-nos melhores a cada dia que praticamos.