

TRABALHO PRÁTICO (DCC146)
ASPECTOS TEÓRICOS DA COMPUTAÇÃO 2021.1

Professor: GLEIPH GHIOTTO LIMA DE MENEZES

Autores: Débora Izabel Rocha Duarte	Matrícula:	201776029
Fabício de Sousa Guidine	Matrícula:	201776037
Juarez de Paula Campos Júnior	Matrícula:	201676022
Luiz Guilherme Almas Araujo	Matrícula:	201676050

1. DESCRIÇÃO DE SAÍDA DE BUILD

Quando você constrói um projeto de aplicativo Java que tem uma classe principal, a IDE copia automaticamente todos os arquivos JAR no classpath de projetos para a pasta dist / lib de seus projetos. A IDE também adiciona cada um dos arquivos JAR ao elemento Class-Path no aplicativo. Para executar o projeto a partir da linha de comando, vá para a pasta dist e digite o seguinte: `java -jar "TrabalhoDCC146.jar"`.

2. DOCUMENTAÇÃO DAS CLASSES

2.1. CLASSE ARQUIVO

Classe com os métodos que irão manipular os objetos (arquivos) que serão lidos pelo programa e que serão salvos na memória do computador.

Função	Tipo / Retorno	Parâmetros		Descrição
setExpressao	void	String expressão	String caminhoArq	Método que tem como finalidade criar o arquivo na memória do computador.
		Expressão que será salva no arquivo	Caminho do Arquivo	
gravaDivisao	void	String:divisão	String:caminho	Método que tem como finalidade gravar a divisão da entrada em tags em um arquivo
		A divisão da entrada em tags	Caminho do arquivo	

getExpressao	ArrayList <String> Retorna o conteúdo do arquivo	String: caminhoArq	Método que tem como finalidade ler o arquivo que está salvo no computador
		Caminho do arquivo que será lido	
existe	Boolean True se o caminho existe e false se não existe.	String: caminhoArq	Método para ver se um arquivo existe.
		Caminho do arquivo	

1.2. CLASSE GERA AUTOMATO

Função	Tipo / Retorno	Parâmetros	Descrição
AFN geraAFN	pilha	String auxExpressao String nome String expressao Pilha pilhaAFN Pilha operador	Gera um AFN a partir da expressão passada como parâmetro.
setPrioridadeOperandos	boolean	Character segundoOp Char primeiroOp	Lida com a prioridade dos operandos na expressão.
Operacao	void	Pilha operador	Faz a operação necessária de acordo com o símbolo no topo da pilha.

fechoKleene	void	Pilha AFN nfa	Lida com a operação do fecho de kleene.
Concatenacao	void	Pilha AFN afn2 Pilha AFN afn1	Faz a operação de concatenação.
OperacaoOu	void	Pilha AFN afn2 Pilha AFN afn1	Faz a operacao OU
Empilhar	void	Pilha AFN nfa	Coloca o alfabeto na pilha
colocaUmaConcatenacao	String	String expressao	Coloca um símbolo de concatenação '.' quando existe a concatenação entre dois símbolos
eDoAlfabeto	boolean	char simbolo	Método que verifica se o símbolo passado como parâmetro pertence ao alfabeto do autômato.
removeTransicaoLambda	void	Pilha set1 Pilha set2 ArrayList estadosLambda	Remove transições vazias
organizaEstados	void	Character c Set estados Set estados1 ArrayList allStates	Reorganiza os estados com base no símbolo

1.3. CLASSE MENU

Classe responsável por cuidar das entradas e saídas do programa, recebendo os comandos e parâmetros do usuário, identificando e validando esses comandos. Recebe também as expressões e strings a serem divididas. Mantém o programa rodando aguardando os comandos do usuário.

- **private static Scanner leitura = new Scanner(System.in):** faz a leitura das informações passadas através do teclado.
- **private ArrayList<String> tagsMemoria = new ArrayList<String>():** armazena as tags.
- **private String cmd = " ":** comando passado pelo usuário.
- **private String caminho = " ":** caminho para arquivo passado pelo usuário.
- **private String entradaInformada = " ":** entrada a ser dividida em tags.

Função	Tipo / Retorno	Parâmetros		Descrição
menu	void			Função menu que dará início ao programa e cuida de toda a lógica
entrada	boolean true se o comando é válido e false se não.			Função com objetivo de validar o comando passado ou se não for comando, será considerado como uma tag. Divide também a entrada em duas, a fim de separar o comando de seu parâmetro. Guarda as duas partes em um array de Strings.
validadeComando	boolean	String[]: comando		Função que tem o objetivo de reconhecer o comando e atribuir na variável 'cmd' e se caso, o comando for algum que é necessário de um caminho, o mesmo atribui o caminho na variável 'caminho'.
		array de strings de tamanho 2 contendo um comando e seu parâmetro.		
entradaParaDividirArq	void	string caminho Arraylist expressoes int i		Recebe Strings para divisão de um arquivo, cujo caminho foi informado pelo usuário. Salva as expressões em um array, envia cada uma para divisão na classe Automato. Recebe de volta as divisões em TAG, mostra para o usuário e as prepara para gravação em arquivo.

tagsParaAutomato	void	int i	
geraAutomato	void	ArrayList tags GeraAutomato gera	
definicoesDetag	void	String: caminhoTag	Função que tem como objetivo, ler um arquivo passado pelo usuário e verificar se as tags contidas nele, são tags válidas ou não e se caso for, adicionar essas tags na memória de execução do programa (caso alguma tag desse arquivo tiver o mesmo nome de alguma tag já salva na memória de execução do programa, a mesma será desconsiderada).
		caminho do arquivo contendo as definições de Tag.	
salvaDivisaoTags	void	String: caminhoDiv	Salva a divisão da entrada em tags em um arquivo.
		Caminho do arquivo.	
salvarTags	void	String: caminhoTag	Função que tem como objetivo, salvar as tags válidas que estão na memória de execução do programa em um arquivo na memória do computador.
		Caminho do Arquivo	
adicionaTagMemoria	void	String: tag	Função que tem como objetivo, verificar a tag que foi passada pelo usuário é uma tag válida ou não. Se caso for válido, adiciona na memória de execução do programa e se caso não, retorna uma mensagem de erro.
		Tag a ser validada e armazenada.	

1.4 CLASSE RECONHECE TAG

Classe com as operações referentes à validação e leitura de tags e expressões regulares.

Função	Tipo / Retorno	Parâmetros	Descrição
separaString	Void	expressao toCharArray	Método que transforma uma string em um array de char.
validaTag	boolean	Int i Int j	Método que verifica se a tag inserida antes da expressão regular é válida ou não.
verificaTag	boolean	Tag inserida	Verifique se já existe uma tag com o nome inserido.
validaExpressaoRegular	boolean	String escape String comparacao String fechoKleene String concatenacao	Metodo que verifica se a expressao regular em notacao polonesa reversa
validaExpressao	boolean		Se a tag for válida e não existir outra tag com o mesmo nome e a expressão regular for válida, então a expressão completa é válida.

1.5 CLASSE CLASSIFICA STRING

Função	Tipo / Retorno	Parâmetros	Descrição
adAFN	void	ArrayList afns GeraAutomato ga	Faz a classificação das Strings
divideString	ArrayList	String str int percorreStr String s ArrayList afds	
addNome	void	String nNome ArrayList divisoes	Função auxiliar para o método divide entrada, adiciona o "nome" da Tag em um array de divisões que
verifica	boolean	String nNome ArrayList divisoes	Verifica se a TAG que está sendo inserida é igual a última TAG inserida, e, se sim, não insere o nome da TAG no array.
classifica	boolean	AFD afd String s	Método que verifica se uma string (expressão) é aceita pelo AFN passado como parâmetro.

1.6 CLASSE ESTADO

Classe Estado com transições para cada símbolo encontrado na expressão da TAG.

Função	Tipo / Retorno	Parâmetros	Descrição
addTransicao	void	char chave Estado proximo ArrayList lista	Adiciona a transição entre os estados e insere na ArrayList ou cria o ArrayList com base na chave.
getTransicoes	lista	char c	Pega todos os estados de transição com base no símbolo.