



**UNIVERSIDADE FEDERAL DE SERGIPE
DEPARTAMENTO DE SISTEMAS DE INFORMAÇÃO - DSI
LINGUAGENS FORMAIS E TRADUTORES**

Abraão Pereira Alves e Igor Bruno dos Santos

Documentação Léxica da Linguagem de Programação Ada

Itabaiana, 31 de outubro de 2018

1. Introdução

Ada é uma Linguagem de programação estruturada, de tipagem estática, ~~é uma linguagem~~ imperativa, orientada a objetos ~~e é uma linguagem de alto nível~~, originada de Pascal, Simula e outras linguagens. O nome Ada vem de Ada Lovelace, a mulher conhecida por ter escrito o primeiro algoritmo para ser processado por uma máquina na **história da computação**. Linguagem de programação patrocinada pelo Departamento de Defesa dos EUA, que teve uma base de Cobol e Basic e que anos mais tarde serviria de base para o Ruby. ADA surgiu em 1982, e é o resultado do mais extensivo e dispendioso esforço de projeto de uma linguagem já realizado. Ela foi criada com o propósito de resolver o problema do Departamento de Defesa dos Estados Unidos que em 1974 era composta de sistemas **incorporados**, os custos de software elevavam-se rapidamente; primeiro, por causa da crescente complexidade dos sistemas. Mais de 450 diferentes linguagens de programação estavam em uso para projetos do Departamento de Defesa, e nenhuma delas era padronizada. Então o Departamento criou um grupo que era liderado por Jean Ichbiah da CII Honeywell Bull. No Mesmo ano que ela surgiu, foi publicado seu primeiro padrão ISO (Ada 83). Suas aplicações são mais significativas ~~são~~ na área de sistemas de tempo real e de segurança.

2. Elementos léxicos de Ada

2.1 Palavras Reservadas

Palavras reservadas em uma linguagem de programação são palavras que tem um significado especial e portanto não podem ser utilizadas para outra finalidade a não ser para as atividades de sua própria natureza.

A linguagem de programação Ada possui ao todo 73 palavras reservadas de acordo com o Ada Reference Manual. Dentre elas foram selecionadas 22, são elas:

with	use	procedure	is	begin
end	if	then	else	endif
for	loop	while	exit	when
string	integer	float	begin	function
type	in			

2.2 Operadores e Delimitadores

Delimitadores são símbolos que determinam onde algo começa e/ou termina dentro de um programa construído na linguagem Ada, já os operadores são

elementos que são aplicados a um ou mais operandos. Segue abaixo operadores e delimitadores simples da linguagem Ada:

&	'	()	*	+	,	-
!	/	:	;	<	=	>	!

Operadores e delimitadores compostos:

=>	..	**	:=	/=
>=	<=	<<	>>	<>

2.3. Identificadores

Identificadores são usados como nome. Um identificador é qualquer objeto em Ada e deve ser formado seguindo algumas regras bastante rígidas. Um identificador tem que começar com uma letra do alfabeto e pode ser formado de letras, números e underlines, contanto que não hajam underlines duplos ou triplos e assim por diante e que ele não seja o último caracter. Ada não é “case-sensitive”. Não podem ser usados espaços em branco ou caracteres especiais como parte de um identificador. É desejável que os identificadores sejam significativos.

2.4 Literais Numéricos

Existe dois tipos de numerico_literal, literal real e literal inteiro. Um literal real é um numerico_literal que inclui o ponto; Um literal inteiro é um numerico_literal que não inclui o ponto.

2.5 Literais decimais

Um decimal_literal é um numerico_literal na notação decimal convencional (ou seja, a base é dez).

Exemplos: 123 --- inteiro, 3.14159 -- real.

2.6 Caracteres Literais

Um caractere_literal é formado por um caractere entre um delimitador gráfico, dois caracteres de apóstrofo.

Exemplos: 'A', '*', '"', ' '.

2.7 Strings Literais

Uma `String_literal` é uma sequência de caracteres entre aspas duplas
Exemplos: `"Message of the day:", "`, `"", "`.

2.8 Comentários

Inicia com dois traços em sequência.

Exemplos: `-- Comentário, -- the last sentence above echoes the Algol 68 report
end;`.

2.9 Variáveis

Para declarar uma variável, ela primeiramente tem que ser declarada entre `Procedure nome_do_programa is` e `begin`. Coloca-se o nome da variável seguido de `:` e o tipo.

Exemplo:

```
Procedure nome_do_programa is
    numero : integer;
begin.
```

3. Referências

Ada-Europe, Ada Reference Manual - Language and Standard Libraries, ISO/IEC 8652:2012(E), -jan.2016.