

# Linguagem de Programação: Pascal

Departamento de Sistemas de Informação

Universidade Federal de Sergipe (UFS) - Itabaiana, SE - Brasil

# Sumário

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Lexemas de Pascal</b>	<b>3</b>
2.1	Identificadores . . . . .	3
2.1.1	Declaração de variáveis (Var) . . . . .	4
2.1.2	Declaração de Constantes (Const) . . . . .	5
2.1.3	Variáveis de atribuição . . . . .	5
2.2	Comentários . . . . .	5
2.3	Palavras Reservadas . . . . .	6
2.4	Operadores e Delimitadores . . . . .	6
2.4.1	Aritméticos . . . . .	6
2.4.2	Relacionais . . . . .	6
2.4.3	Lógicos . . . . .	6
2.5	Literais Numéricos . . . . .	6
2.5.1	Números inteiros . . . . .	6
2.5.2	Números reais . . . . .	7
2.6	Literal Booleano . . . . .	7
2.7	Literal String . . . . .	7
2.8	Literal Char . . . . .	8
2.9	Tipo array (matriz) . . . . .	8
2.10	Bloco de um programa . . . . .	8

# 1 Introdução

Pascal é uma linguagem de programação que recebeu o seu nome em homenagem ao matemático Blaise Pascal. A linguagem Pascal foi criada em 1970 pelo suíço Niklaus Wirth que tinha em mente encorajar o uso do código estruturado. O próprio Niklaus Wirth diz que Pascal foi criada simultaneamente para ensinar programação estruturada e para ser utilizada no desenvolvimento de software.

A linguagem de programação Pascal foi inventada por Niklaus Wirth e a sua versão original encontra-se no “PASCAL USER MANUAL REPORT” de Kathleen Jensen e Niklaus Wirth.

Existem várias versões de do Pascal. Desenvolvendo-se vários dialectos do Pascal conhecidos como Turbo Pascal. As várias versões do Turbo Pascal procuram sempre melhorar as versões anteriores.

Compromisso entre as características de uma larga variedade de linguagens com o objetivo de uso por parte de estudantes e eficiência na compilação em um só passo. Como linguagem de ensino teve tremendo sucesso e é largamente empregada a despeito de suas deficiências para a implementação de sistemas de grande porte.

Em 1986 foi divulgado o Object Pascal que é a extensão da linguagem Pascal para a orientação a objetos.

Em 1987 a empresa Borland lançou o ambiente Turbo Pascal 4.0 que foi um grande sucesso pela facilidade de edição, compilação e execução que, pela primeira vez, foram integrados no ambiente de microcomputadores. O Pascal v7.0 da Borland (para o DOS, Windows 16-bit e a modalidade protegida) foi liberado em 1992 sendo a última versão do Pascal Turbo a ser nomeado desta forma.

## 2 Lexemas de Pascal

Linguagem parecida com a linguagem algorítmica. Utiliza o paradigma estruturado, onde qualquer problema pode ser resolvido utilizando três estruturas:

- Sequencial;
- Condicional;
- Iterativa (repetição).

### 2.1 Identificadores

Identificadores são utilizados para denominar programas, rótulos, constantes, variáveis, tipos, procedimentos, funções e units criados pelo programador.

São regras, em Pascal para se constituir corretamente um identificador:

1. Um identificador não pode ser igual a uma palavra reservada;
2. Um identificador deve ter no máximo 255 caracteres;
3. Os símbolos que constituem um identificador podem ser:
  - letras: ‘A’ até ‘Z’ ou ‘a’ até ‘z’;
  - dígitos: 0 até 9;
  - um sublinhado: \_
4. O primeiro símbolo deve ser uma letra ou o sublinhado.

Assim, não são aceitos espaços ou caracteres especiais em qualquer parte de um identificador. Até mesmo o cedilha (‘ç’) é considerado um caractere especial.

São exemplos de identificadores válidos:

- a, a1, b568, codigo, raizquadrada, preco especial;
- \_veiculo, elemento do conjunto, raiz quadrada, b54;
- CorVeiculo, Ano, MES, SaLdO, Parcela Quitada;
- Este nome eh muito longo e raramente deve ser usado mas eh valido.

Exemplos de identificadores inválidos:

- 12porcento, 4por4: iniciam por número;
- cor veiculo: contém espaço em branco;
- preço: possui um caractere inválido (cedilha);
- pássaro: apresenta acentuação;
- %b54 : não inicia por letra ou por sublinhado.

Observações:

- Os identificadores na linguagem programação PASCAL não são case-sensitive;
- Não se pode declarar uma variável com o mesmo nome que foi atribuído ao programa ou a qualquer outro identificador previamente declarado no mesmo escopo;
- Como o compilador Pascal ignora o estado de maiúsculas e minúsculas, os identificadores podem ser grafados de maneiras diferentes e ter o mesmo significado, como por exemplo: saldo, SALDO, Saldo ou SaLdO;
- Considerar todos os identificadores como letras minúsculas.

### 2.1.1 Declaração de variáveis (Var)

São representados por identificadores, formados por: uma letra, ou uma letra seguida de letras ou dígitos.

Exemplo: A, Nota, matricula, x, a, a1, a2, bd, A32B.

Maiúsculas são iguais a minúsculas: AB = aB.

Exemplo declaração de variáveis(Var):

```
var
  nome : string;
  idade : integer;
  altura: real;
```

ou

```
var
  a, b, c : integer;
```

### 2.1.2 Declaração de Constantes (Const)

Valor fixo. Não pode ser alterado durante a execução de um programa. Podem ser numéricas, lógicas ou alfanuméricas (literal).

Numéricas: 25, +34, -315, -0.53, 7.8E-3, .13

Lógicas: true, false

Alfanuméricas: A-Z, a-z, 0-9, !, ?, /, \*, &, \$, %,...

Constantes diretas:

Exemplo: 25, 'CADEIA DE SIMBOLOS'.

Exemplo declaração de constantes (const):

Constantes denominadas:

Exemplo: PI cujo valor é 3.14159.

```
const
```

```
    PI = 3.14;
```

```
    nota_max = 10;
```

É importante ressaltar que o operador utilizado para declarar uma constante é o '=' diferentemente do operador utilizado para declarar uma variável que é o ':='.

### 2.1.3 Variáveis de atribuição

No comando de atribuição, a variável e seu valor devem ter o mesmo tipo, exceto nos seguintes casos:

- Variáveis reais podem receber valores integer;
- Variáveis string podem receber valores char.

Exemplos:

```
time := 'Flamengo';
```

```
media := (nota1+nota2) / 2;
```

## 2.2 Comentários

Um comentário de linha única é uma sequência de caracteres que começa com // e se estende até o final da linha.

Um comentário de várias linhas pode ser feito de duas formas diferentes, são elas:

- A primeira consiste em uma sequência de caracteres entre { e };
- A segunda forma consiste em uma sequência de caracteres entre (\* e \*);

Os comentários de várias linhas podem ser aninhados, mas precisam ser aninhados corretamente. Portanto, um comentário como /\*\*\*/ será rejeitado como tendo um comentário não terminado.

## 2.3 Palavras Reservadas

and	end	procedure	while
array	for	program	with
begin	function	until	to
case	mod	repeat	of
const	if	var	end
div	then	nil	
do	or	shr	
downto	not	shl	
else	xor	string	

## 2.4 Operadores e Delimitadores

### 2.4.1 Aritméticos

Operador	Operação
+	Soma
-	Subtração
*	Multiplicação
/	Divisão

### 2.4.2 Relacionais

Operador	Operação
=	Igual a
<>	Diferente
<=	Menor ou igual
>=	Maior ou igual
>	Menor que
<	Maior que

### 2.4.3 Lógicos

Operador	Operação
and	Conjunção (E)
or	Disjunção (OU)
not	Negação (NÃO)
xor	Disjunção exclusiva (OU EXCLUSIVO)

## 2.5 Literais Numéricos

### 2.5.1 Números inteiros

Integer: equivalente ao tipo inteiro em algoritmos.

Nome	Valor mínimo	Valor máximo	Bytes ocupados
integer	-32.768	-32.767	2

Exemplo de declaração: var numero: integer;

Tipos variantes:

Nome	Valor mínimo	Valor máximo	Bytes ocupados
shortint	-128	127	1
byte	0	255	1
word	0	65.535	2
longint	-2.147.483.648	2.147.483.647	4

### 2.5.2 Números reais

Os tipos reais compreendem aqueles representados internamente pelo computador e interpretados como sendo de ponto flutuante.

Nome	Valor mínimo	Valor máximo	Bytes ocupados
Real	-2.9 e-39	1.7 e38	6

Exemplo de declaração: `var numero: real;`

Uma variável do tipo real pode receber a soma de um real com um inteiro. Neste caso, antes da operação, o computador faz a conversão do inteiro para um formato em ponto flutuante.

Tipos variantes:

Nome	Valor mínimo	Valor máximo	Bytes ocupados
single	1.5 e-45	3.4 e38	4
double	5.0 e-324	1.7e308	8
extended	3.4 e-4.932	1.1 e4.932	10
comp	-9.2 e18	9.2 e18	8

## 2.6 Literal Booleano

O boolean (booleano) é um tipo lógico que assume apenas um entre dois possíveis valores: false ou true, que equivalem respectivamente a falso ou verdadeiro (0 ou 1).

Sempre que um valor diferente de 0 for associado a uma variável booleana, esta será verdadeira.

Valores	
true	false

Exemplo de declaração: `var valor: boolean;`

## 2.7 Literal String

O tipo string é utilizado para armazenar palavras, isto é, sequências de símbolos ASCII.

Em Pascal, as palavras, assim como o tipo char, devem estar entre apóstrofes (aspas simples). Pode-se concatenar strings e caracteres usando o operador de adição (+).

Exemplo de declaração: `var time: String;`

Exemplo de atribuição: `time:= 'Flamengo';`

Podemos limitar o tamanho de uma string, utilizando [ + tamanho do texto + ] ao lado da especificação de tipo.

Exemplo de declaração:

```
var time: String [10];  
var sexo: String [1];  
var nome_completo: String [100];
```

## 2.8 Literal Char

O tipo char (ou caractere) armazena apenas um caractere da tabela ASCII<sup>10</sup> e ocupa exatamente um byte de memória. Na realidade, não é o caractere que é armazenado na memória, mas o código ASCII correspondente.

O código ASCII do caractere 'A', por exemplo, é 65. Então, se for atribuído 'A' a uma variável do tipo caractere, na verdade será atribuído o valor 65 para esta variável. Pode-se atribuir diretamente o valor ASCII a uma variável caractere inserindo-se o sustenido na frente do número correspondente.

Exemplo de declaração: var sexo: char;

Exemplo de atribuição: a := #65;  
Equivalente ao caractere 'A' da tabela ASCII.

Exemplo de atribuição: sexo:= 'M';  
Equivalente ao caractere que ocupa a posição 77 da tabela ASCII.

## 2.9 Tipo array (matriz)

Em sua forma elementar, o tipo array é utilizado para armazenar, sob um mesmo nome de variável, uma quantidade fixa de elementos do mesmo tipo. Por isto, o tipo array também é conhecido como um arranjo homogêneo.

A declaração de uma variável do tipo array é feita da seguinte forma:

array [<lista de tipos enumeráveis>] of <tipo>
--

Onde <tipo> é qualquer tipo previamente definido pelo programador ou pré-definido pela linguagem, e <lista de tipos enumeráveis> é uma lista de faixas de números separadas por vírgula.

Por restrição dos tipos enumeráveis, os valores devem ser do tipo ordinal e conhecidos em tempo de compilação. Os casos mais comuns de matrizes são de uma e de duas dimensões, embora possam ser declaradas com dimensões maiores. As de uma única dimensão são costumeiramente denominadas vetores, as de duas dimensões são chamadas de matrizes propriamente ditas.

## 2.10 Bloco de um programa

Um texto contendo um programa na linguagem Pascal deve obedecer a regras de sintaxe (forma) bem específicas, para que o compilador possa reconhecer as suas diversas partes, e convertê-lo (traduzir) para uma sequência equivalente de comandos em linguagem de máquina.

Nas definições abaixo, as palavras e símbolos que não estão entre os sinais de < e > são considerados como elementos da linguagem Pascal e devem ser escritos literalmente. As frases entre os sinais < e > são explicações em português do que deve entrar no lugar delas. Dizemos que essas frases estão escritas na meta-linguagem (a linguagem que usamos para definir uma outra linguagem).

```
program <identificador>;  
<declarações de constantes>  
<declarações de variáveis>  
<declarações de funções e procedimentos>  
<bloco>
```