UNIVERSIDADE FEDERAL DE ALAGOAS

CARLLOS EDUARDO FERREIRA LOPES
RAFAEL EMÍLIO LIMA ALVES

COMPILADORES - Linguagem PYragua

Maceió
Dezembro de 2021

Sumário

1 Estrutura Geral de Um Programa

1.1 Ponto Inicial do Programa

O ponto inicial de um programa na linguagem PYragua é a função obrigatória **Central** ela é declarada a partir da palavra reservada Central e seu escopo é delimitado pelas palavras **Initiate** e **Halt**, as quais são também utilizadas para delimitar o escopo das demais funções e blocos.
Exemplo:

```
Funcao Int Central () Initiate

...

Halt
```

1.2 Definição de Instruções

As instruções só podem ser declaradas dentro do escopo das funções, sendo uma instrução por linha e ao final de cada linha há um ; delimitando cada instrução.

1.3 Variáveis

A declaração de uma variável nessa linguagem e sua atribuição devem ser feitas em instruções distintas. Dessa forma, ao inicializarmos uma variável declaramos seu tipo seguido do identificador da mesma, e o valor atribuído é um valor default. Esse valor será substituído quando uma instrução de atribução for feita para a variável declarada, não podendo ser feita sem a variável estar declarada. A estrura dos nomes de variáveis e funções iniciam-se pelas letras [A-Z] seguidas das letras [a-zA-z] ou os dígitos [0-9] ou o underline, a linguagem PYragua é case sensitive.
Ademais, cada instrução não suporta múltiplas declarações ou atribuiçãos de variáveis.

```
Int A;

A = 10
```

2 Conjunto de Tipos de dados e Nomes

2.1 Palavras reservadas

Initiate, Halt, Central, Funcao, Retorna, Int, Str, Float, Char, Bool, Array, Vazio, Verdade, Falso, Se, SeNao, Loop, Enquanto, Nulo, Pare, Ler, Escrever, Escreverpl, E e Ou.

2.2 Identificador

Um identificador pode começar com letras maíusculas ou minúsculas, podendo ser seguido de letras maíusculas ou minúsculas, dígitos e/ou underline. Não se pode utilizar palavras reservadas, ou espaços em branco. Antes de ser utilizado em um bloco de instruções o identificador deve ser declarado. A linguagem Pyragua é uma linguagem de tipagem estática, não admitindo coerção.

2.3 Comentário

Nessa linguagem os comentários são apenas por linha, cada linha de comentário deve ser iniciada com o caracter: "**§**".

2.4 Inteiro

A palavra reservada Int define as váriaveis/funções do tipo Inteiro, que possuirão um número inteiro de 32 bits.
Exemplo: **Int** x = 10;

2.5 Ponto Flutuante

A palavra reservada Float define as variáveis/funções do tipo Ponto Flutuante, as quais possuem um número de bits limitados a 32 bits.
Exemplo: **Float** media = 9.25;

2.6 Char

A palavra reservada Char define as variáveis do tipo Char, as quais possuem tamanho de bits 8.
Exemplo: **Char** a = 'a';

2.7 Booleano

A palavra reservada Bool define as variáveis ou funções do tipo booleano que pode assumir apenas dois valores: Verdadeiro ou Falso, cada valor atribuído possue um tamanho de 8 bits.
Exemplo: **Bool** isRunning = Falso;

2.8 String

A palavra reservada Str define as variáveis do tipo String com tamanho de 8 bits. Sua atribuição é delimitada por aspas simples(") que indicam o inicio e fim da cadeia de

caracteres.
Exemplo: **Str** Nome = 'Joao';

2.9 Arranjo Unidimensional

Um Arranjo Unidimensional (vetor) na linguagem PYragua é um conjunto de dados do mesmo tipo de tamanho fixo. O tipo, identificador e tamanho são definidos respectivamente obedecendo a estrutura: **Tipo** Identificador [**Tamanho**], sendo um tipo entre os definidos previamente, um identificador de acordo com as regras de nomeação e o tamanho um valor de uma constante inteira, que define quantos dados do tipo determinado serão armazenados no Arranjo.
Exemplo: **Float** Notas [30];

3 Conjunto de operadores

3.1 Operadores Aritméticos

Operação	Símbolo	Exemplo
Adição	+	a + b
Subtração	-	a - b
Multiplicação	*	a * b
Divisão	/	a / b
Resto de Divisão	%	a % b
Unário Negativo	¬	¬ a

3.2 Operadores Relacionais

Operação	Símbolo	Exemplo
Igualdade	==	a == b
Diferente	!=	a != b
Menor que	<	a <b
Maior que	>	a >b
Menor igual	<=	a <= b
Maior igual	>=	a >= 2

3.3 Operadores Lógicos

Operação	Símbolo	Exemplo
Negação	!	!a
Conjunção	E	a E b
Disjunção	Ou	a Ou b

3.4 Operadores de Concatenação

Operação	Símbolo	Exemplo
Concatenação	@	a@b

3.5 Operações de cada tipo de dado

Operador	Tipos que realizam a operação
+, - , *, /	Int, Float
^, %	Int
<, >, <=, >=	Int, Float
==, !=	Int, Float, Bool, Char, Str
E, Ou, !	Bool
@	Int, Float, Char, Str

3.6 Valores Padrão

Os valores padrões da linguagem são atribuídos às variáveis na declaração das mesmas, antes que o usuário faça uma instrução atribuindo novos valores de acordo com a necessidade de sua aplicação. Sendo assim, quando as variáveis são atribuídas, os seguintes valores padrões são associados de acordo com o tipo:

Tipo	Valor Default
Int	0
Float	0.0
Char	Nulo
Str	Nulo
Bool	Falso

3.7 Precedência e Associatividade

Operadores	Operação	Associatividade
-	Unário Negativo	Direita ->Esquerda
* /	Multiplicação, divisão	Esquerda ->Direita
%	Resto	Esquerda ->Direita
- +	Subtração e Adição	Esquerda ->Direita
!	Negação	Direita ->Esquerda
<><= >=	Comparação	Não Associativos
== !=	Igualdade e Desigualdade	Não Associativo
E Ou	Lógicos	Esquerda ->Direita

4 Instruções

4.1 Atribuição

A atribuição na linguagem é definida pelo símbolo '=', onde à esquerda temos o identificador de uma variável já definida e à direita do símbolo temos o valor correspondente ao tipo da variável.

Exemplo:

```
Int A;

A = 10
```

4.2 Estruturas Condicionais de uma ou duas vias

A primeira estrutura condicional deve ser usando a palavra reservada **Se**, seguida de uma expressão lógica entre parênteses e dos limitadores do bloco de código onde instruções específicas ligadas a condicional estão. Após o Se, podemos ter um **SeNao**, onde caso a expressão lógica do Se não assuma um valor verdadeiro, então será executado o bloco de código contido no SeNao, o qual também é delimitado pelas palavras reservadas Initiate e Halt.

Exemplo:

```
Se (EXPRESSÃO LÓGICA) Initiate

...

Halt

SeNao  Initiate


Halt
```

4.3 Estrutura condicional de controle lógico

A palavra reservada atribuída para essa instrução é o **Enquanto** o qual deve ser seguido de uma expressão lógica entre parênteses e os delimitadores de escopo, os quais irão conter instruções que serão executadas em um laço de repetição enquanto a expressão lógica não seja verdadeira.

Exemplo:

```
Enquanto (EXPRESSÃO LÓGICA) Initiate

...

Halt
```

4.4 Estrutura iterativa por contador

Na linguagem PYragua, essa instrução é definida pela palavra reservada **Loop**, a qual é seguida de parênteses com valores para a execução da instrução, cada um deles separados por vírgula.
O primeiro valor é uma variável inteira que define o contador que será incrementado a cada iteração do laço, o segundo valor define o valor inicial desse contador, o valor seguinte é o limitante do contador, o qual define o limite de iterações, e por último o valor incrementado ao contador ao fim de cada repetição.
Após esses valores, temos os delimitadores de escopo (Initiate e Halt), onde estão contidas as instruções que serão realizadas em cada iteração do ciclo.

Exemplo:

```
Loop(Int cont, a, b, c) Initiate

...

Halt
```

4.5 Entrada e Saída

4.5.1 Entrada

Para entrada de dados é utilizada a palavra reservada **Ler**, segunda de parênteses que contém como parâmetro apenas o identificador da variável a qual será atribuída o valor fornecido pelo usuário. Dessa forma, permitindo que o usuário informe o valor da variável.

4.5.2 Saída

Para a saída de dados, a linguagem utiliza o comando **Escrever**, o qual é seguido de parênteses, onde entre aspas é especifiada a mensagem e/ou valores de variáveis que serão exibidas para o usuário.

4.6 Funções

As funções podem ser definidas em qualquer parte do documento, exceto dentro do escopo de outras funções. A declaração de uma função é iniciada pela palavra reservada Funcao seguida do tipo de dado do retorno da função (Int, Bool, Str, Float, Char) seguido do identificador da função que deve ser escrito em letras **minúsculas**, seguido de parênteses que podem ou não conter parâmetros separados por vírgula. Por fim, seu escopo é delimitado pelas palavras reservadas Initiate e Halt.

```
Funcao Bool NomeDaFuncao (Int a, Int B) Initiate
  Int a = 0;
  Int B = 1;


  Retorna Verdadeiro;

Halt
```

5 Algoritmos

5.1 Alô Mundo

```
Funcao Int Central() Initiate

    Escrever('Alô mundo!')

Halt

```

5.2 Fibonacci

```
Funcao Int fibonacci(Int n) Initiate
    Se(n <= 1) Initiate
        Retorna n;
    Halt

    Retorna fibonacci(n - 1) + fibonacci(n - 2);

Halt


Funcao Int Central() Initiate

    Int n;
    Escrever('Informe a quantidade desejada de termos da sequência');
    Ler(n);
    Escrever(fibonacci(n);

Halt
```

5.3 Shell Sort

```
Funcao Int Central() Initiate
        Int n, atual;
        Escrever('Tamanho do array: ');
        Ler(n);

        Int array[n];

        Escrever('Elementos do array: ');

        Loop (Int i = 0, 1, n) Initiate
            Ler(array[i]);
        Halt

        Escrever('Array: ');

        Loop (Int i = 0, 1, n) Initiate
            atual = array[i];
            Escreverpl(atual);
        Halt

        shellsort(array[n], n);

        Escrever('Valores ordenados: ');

        Loop (Int i = 0, 1, n) Initiate
            atual = array[i];
            Escreverpl(atual);
        Halt

        Retorna;
Halt
```

```
Funcao Vazio shellsort(Int array[ ], Int n) Initiate
        Int c, j;
        Int m = 1;

        Enquanto (m < n) Initiate
            m = m * 3 + 1;
        Halt

        m = m / 3;

        Enquanto(m > 0) Initiate

            Loop (Int i = m, 1, n) Initiate

                    c = array[i];
                    j = i;

                    Enquanto (j >= m E array[j - m] > c) Initiate

                            array[j] = array[j - m];
                            j = j - m;

                    Halt

                    array[j] = c;
            Halt

            m = m / 2;

        Halt

        Retorna;

    Halt
```