Universidade Federal de Alagoas - UFAL
Instituto de computação - IC
Disciplina: Compiladores
Prof: Drº Alcino Dall Igna Junior

Especificação da Linguagem ENL

Alunos: Thiago Emmanuel G. Rodrigues
Erivaldo Lourenço Mariano

Maceió, 26 de Março de 2019

Introdução

A linguagem ENL foi baseada na linguagem C, a linguagem é estática, case sensitive, desenvolvida com a ideia de ser prática, fácil e rápida.

Estrutura geral do programa

Na linguagem ENL a função *begin()* será o ponto de entrada do programa. Se a existir um programa na linguagem ENL ela tem que ter a função void *begin().* Ex de um programa em ENL:

```
void begin()
{
}
```

Todo comando em ENL tem por obrigação terminar com ‘;’.

As variáveis podem ser declaradas dentro da função begin, de outras funções nesse caso serão variáveis locais para declarar uma variável global basta declarar ela antes de **void begin().**

```
int b;
void begin()
{
    int a = 1;
     b = 2;
```

```
}
```

Variáveis do mesmo tipo podem ser declaradas na mesma linha. Toda declaração de variável termina com ‘;’.

Comentários

Pode se fazer comentários usando o operador **#** no início e final do comentário.

Funções

As funções são declaradas com a palavra reservada *`function`* *, e a palavra reservada* *`return`* *indica o retorno da função.*

Exemplo de declaração de função:

```
function tipo_de_retorno identificador( tipo parametro1, tipo parametro2, … , tipo parametroN ){

    return valor_ou_alguma_variável;
}
```

Tipos de dados

A Linguagem ENL é estaticamente tipada e é necessário a declaração explícita do tipo das variáveis em sua declaração. As palavras que definem seu tipo são reservadas.

Inteiro

Palavra **int** define um tipo inteiro. A declaração de uma variável do tipo inteiro é feita da seguinte forma:

int variavel;

Variáveis do tipo inteiro podem fazer as seguintes operações: soma, subtração, multiplicação, divisão e resto.

Ponto flutuante

Palavra **float** define um tipo ponto flutuante. A declaração de uma variável de ponto flutuante é da seguinte forma:

float variavel;

Variáveis do tipo ponto flutuante podem fazer as seguintes operações: soma, subtração, multiplicação e divisão.

Caractere

A palavra **char** define um tipo caractere que está representado no padrão ASCII puro. A declaração é da seguinte forma:

char caractere;

Cadeia de caractere

A palavra **cchar** define uma cadeia de caracteres. Uma cadeira de caractere é uma sequência de caracteres. A declaração é da seguinte forma:

cchar cadeia[tamanho];

Todos os elementos do cchar sao do tipo char. As operações que podem ser feita com variáveis do tipo cchar sao: concatenação.

Booleano

A palavra **boolean** define uma variável que tem como valores *true* ou *false.*
Sendo a declaração da seguinte forma:

bool variavel;

Arranjo unidimensional

vector - Define um arranjo de uma dimensão. Esse arranjo é uma coleção de objetos que contém obrigatoriamente, os tipos primitivos, e são declarados da seguinte forma:

vector tipo variavel[tamanho];

vector int exemplo[10];

Inicialização

As variáveis podem ser inicializadas no momento da criação por um valor ou expressão. Caso não sejam inicializadas, seus valores default são os que seguem:

Tipo	Valor default
int	0
float	0.0
bool	false
array	inicializado com o valor padrão de seu tipo definido
char, string	null

Operadores

Dividimos os operadores em vários tipos, verificados na tabela abaixo:

Operadores aritméticos	**Símbolo**
Soma Subtração	+ -

Multiplicação Divisão Módulo unário negativo	* / % ~

A associatividade para operadores aritméticos será da esquerda para direita. A ordem de precedência segue abaixo:

- Unário negativo, Multiplicação, Divisão, Módulo
- Adição e Subtração

Operadores de comparação	Símbolo
Igual Diferente Menor igual Maior igual Menor que Maior que	== != <= >= < >

A associatividade para operadores de comparação será da esquerda para direita. A ordem de precedência segue abaixo:

- < , > , <= , >=
- == !=

Operadores lógico	Simbolo
AND OR NOT	&& || !

A associatividade para operadores lógicos será da esquerda para direita. A ordem de precedência segue abaixo:

- !
- && ||

Coerção

A coerção pode ser feita colocando o tipo da variável que você quer na frente da variável que deseja mudar.
A linguagem suporta coerção entre int e float, e também entre int, float, e bool para char ou cchar.

Concatenação

A linguagem ENL suporta concatenação de caracteres através do operador **++.**

Atribuição

A atribuição é feita usando o operador '=', como mostra na tabela de operadores. Funciona de modo tal que o valor da esquerda do operador = , recebe o valor da direita.

EX:

```
int a = b+1;
```

Neste caso a variável recebe o resultado da expressão b+1. A atribuição é associativa a direita, de modo que no caso **a=b=c**, feito primeiro a operação **b=c** e o resultado é atribuído a **a**.

Operações suportadas

Tipo	**Operadores aritméticos**	**Operadores de comparação**	**Operadores lógicos**	**Concatenação**
int	Soma Subtração Multiplicação Divisão Módulo unário negativo	Igual Diferente Menor igual Maior igual Menor que Maior que		Se concatenados a um caractere ou cadeia de caracteres
float	Soma Subtração Multiplicação Divisão Módulo unário negativo	Igual Diferente Menor igual Maior igual Menor que Maior que		Se concatenados a um caractere ou cadeia de caracteres
char		Igual Diferente Menor igual Maior igual Menor que Maior que		Podem ser concatenados
cchar		Igual Diferente Menor igual Maior igual Menor que Maior que		Podem ser concatenados
bool		Igual Diferente	AND OR NOT	Se concatenados a um caractere ou cadeia de caracteres
vect	Vai suportar as operações que o tipo do vetor suporta	Vai suportar as operações que o tipo do vetor suporta	Vai suportar as operações que o tipo do vetor suporta	Vai suportar as operações que o tipo do vetor suporta

Instruções

Estrutura condicional de uma e duas vias:

```
if  ( expressão lógica ){
        instruções
}
else{
}
```

Estrutura iterativa com controle lógico:

```
while ( expressão lógica ){
        instruções
}
```

Estrutura iterativa controlada por contador:

```
for ( inicio, fim, ritmo) {
        instruções
}
```

A palavra reservada ‘break’ pode ser usada para sair da estrutura iterativa.

Programas:

Alo mundo:

```
int begin()
{
     put(‘Ola mundo’);

}
```

Fibonacci

```
function void fibonacci(int n){
```

```
     int num, sum, cont, a, b;
     num = n;
     sum = 0;
     a = 1;
     b = 1;
     put(sum);
     for(i = 1; i < num; i = i+1)
     {
          b = a;
          a = sum;
          sum = a + b;
          put(sum);
     }
}

int begin()
{
     fibonacci(5);
}
```

Shell Sort

```
function void shellSort(vect vet, int tam){
     int i, j, valor;
     int gap = 1;

     while(gap<tam){
          gap = 3*gap+1;
     }
     while (gap>1){
          gap = gap/3;
          for(i=gap; i<tam; i=i+1){
               valor = vet[i];
               j = i - gap;
               while(j >= 0 && valor < vet[j]){
                    vet[j+gap] = vet[j];
                    j = j - gap;
               }
          vet[j + gap] = valor;
          }
     }
```

```
}


int begin()
{

     int vetor[50], tam = 50;
     shellsort(vetor, tam);
}
```