Universidade Federal de Alagoas
Instituto de Computação
Ciência da computação

Compiladores - Especificação

Harlan

José Leandro da Silva Filho

Maceió, 15 de Março de 2018

Conteúdo

1. Introdução

Harlan é uma linguagem de programação estaticamente tipada, imperativa e fortemente influenciada por C e Python.

2. Estrutura

Funções e variáveis podem ser declaradas em qualquer lugar do programa e são acessíveis a partir do momento em que forem declaradas. Variáveis quando declaradas só são acessíveis pelo escopo local. A execução de todo programa é iniciada a partir da função main() que deve ser obrigatoriamente definida sem parâmetros.

3. Tipos

Harlan possui o seguinte conjunto de tipos: inteiro (int), ponto flutuante (float), cadeia de caractere (string), booleano (bool) e arrays unidimensionais (definidos usando o operador “[]”). Não existe coerção de tipo em Harlan. Constantes numéricas de ponto flutuante são definidas por digitos com decimais separados por “.”.

4. Operadores

A tabela abaixo descreve a precedência de operadores em Harlan, da menor precedência para a maior.

or	“Ou” booleano
and	“E” booleano
not	Negação
<, <=, >, >=, !=, ==, in	Comparação
+, -	Adição/Concatenação e Subtração
*, /, %	Multiplicação, Divisão e Resto
-x, +x	Unário negativo e positivo

A avaliação de operadores que possuírem a mesma precedência segue a regra de associatividade da esquerda para a direita.

5. Identificadores

Identificadores são obrigatoriamente iniciados por letras. Subsequentemente ao primeiro caractere, podem conter letras e digitos. Identificadores não podem conter espaços em branco.

6. Comentários

Comentários são feitos usando o operador “//”.

7. Instruções

7.1 if - else

A instrução “if” é usada da seguinte forma:

```
if(condicao) {
        corpo_do_if
}
```

A cláusula “else” pode ser opcionalmente adicionada a instrução para definir um corpo de instruções alternativo para ser executado caso a “condicao” não seja satisfeita.

```
if(condicao) {
        corpo_do_if
} else {
        corpo_do_else
}
```

No caso de instruções if aninhadas, a cláusula else pertencerá a última instrução if fechada.

7.2 while

A instrução while executa seu bloco até que a condição (expressão booleana) estabelecida não seja mais verdadeira.

```
while(condicao) {
        corpo_do_while
}
```

7.3 for

A instrução for é usada para criar um loop que será executado uma quantidade predeterminada de vezes.

```
for i in range(0, 10) {
        corpo_do_for
}
```

A função embutida range() retorna um iterador que será percorrido pela instrução durante sua execução.

7.4 input()

Entrada de dados em Harlan é obtida através do retorno da função embutida input().

7.5 print()

A função print(format, ...) exibirá o conteúdo dos parâmetros subsequentes ao primeiro na saída padrão em um formato definido pelo primeiro parâmetro (format).
Exemplo:

```
print("%s", "Hello, world!");
```

8. Atribuição

Em Harlan, atribuição é feita através do operador de atribuição “=”.
Exemplo:

```
int x = 15;
int y = 10 + x;
```

9. Funções

Funções em Harlan são definidas da seguinte forma:

```
int foo(int x) {
        return x*x;
}
```

O tipo dos parâmetros da função são especificados antes dos identificadores de cada parametro. O retorno da função é definido antes do identificador da função.

10. Escopo

Escopos são delimitador por chaves (“{}”).

11. Exemplos de código

Olá mundo:

```
main() {
	print("Olá mundo!");
}
```

Série fibonacci:

```
fib(int n) {
	int a = 0;
	int b = 1;
	int next;

	while(a < n) {
		print(a);
		next = a + b;
		a = b;
		b = next;
	}
}
```

12. Especificação dos tokens

Expressões regulares

ID = “[A - Z][a - z] [A - Z][a - z][0 - 9]*”;
KWIO = “input” | “print”;
KWMAIN = “main”;
KWIF = “if”;
KWELSE = “else”;
KWFOR = “for”;
KWWHILE = “while”;
PTYPE = “string“ | “int“ | “float” | “bool“;
CTEINT = “[0 - 9]+”;
CTEFLOAT = “[+|-]?([0 - 9]*\\.[0 - 9]+)”;
LTBOOL = “true” | “false”;
OPAT = “=”;
OPCMP = “>” | “>=” | “<” | “<=” | “==” | “!=”;
OPMBR = “in”;
OPAD = “+” | “-”;
OPML = “*” | “/” | “%”;
OPCONJ = “and”;
OPDISJ = “or”;
OPNEG = “not”;
ST = “;”
CLN = “,”;
PARST = “(“;
SQBRST = “[”;
CLBRST = “{“;
PAREND = “)”;
SQBREND = “]”;
CLBREND = “}”;
COM = “//[A - B][a - z][0 - 9]*”;

Categorias

Token - Categoria
ID - 1
KWMAIN - 2
KWIF - 3
KWELSE - 4
KWFOR - 5
KWWHILE - 6
PTYPE - 7
CTEINT - 8
CTEFLOAT - 9
LTBOOL - 10
OPAT - 11
OPCMP - 12
OPMBR - 13
OPAD - 14
OPML - 15
OPCONJ - 16
OPDISJ - 17
OPNEG - 18
ST - 19
CLN - 20
PARST - 21
SQBRST - 22
CLBRST - 23
PAREND - 24
SQBREND - 25
CLBREND - 26
COM - 27