Linguagem de Programação: Java

Departamento de Sistemas de Informação
Universidade Federal de Sergipe (UFS) – Itabaiana, SE – Brasil

Clovijan Bispo Rocha
Gilmário Dos Santos Silva
Magda Tainy Nunes Amaral

1. Introdução

Lua é uma linguagem de programação de extensão projetada para dar suporte à programação procedimental em geral com facilidades para a descrição de dados. Ela também oferece um bom suporte para programação orientada a objetos, programação funcional, e programação orientada a dados.

Lua é planejada para ser usada como uma linguagem de script poderosa, leve, e embarcável por qualquer programa que necessite de uma. Lua é implementada como uma biblioteca, escrita em C puro, o subconjunto comum de C Padrão e C++. Por ser uma linguagem de extensão, Lua não possui a noção de um programa "principal": ela somente funciona embarcada em um cliente hospedeiro, chamado de programa embarcante ou simplesmente de hospedeiro.

O programa hospedeiro pode invocar funções para executar um pedaço de código Lua, pode escrever e ler variáveis Lua, e pode registrar funções C para serem chamadas por código Lua. Através do uso de funções C, Lua pode ser aumentada para lidar com uma variedade ampla de domínios diferentes, criando assim linguagens de programação personalizadas que compartilham um arcabouço sintático.

A distribuição Lua inclui um exemplo de um programa hospedeiro chamado Lua, o qual usa a biblioteca Lua para oferecer um interpretador Lua de linha de comando completo, para uso interativo ou em lote. Lua é software livre, e é fornecido como de praxe sem garantias, como dito em sua licença. A implementação descrita neste manual está disponível no site oficial de Lua, www.lua.org.

2. Lexemas do Lua

Em Lua, Nomes, também chamados de identificadores, podem ser qualquer cadeia de letras, dígitos, e sublinhados que não começam com um dígito. Esta definição está de acordo com a definição de nomes na maioria das linguagens. Além disso, a definição de letras irá depender de qual é o idioma (locale), em outras palavras, quaisquer caracteres considerados como alfabético pelo idioma corrente pode ser usado como um identificador. Esses identificadores são usados para nomear variáveis e campos de tabelas.

É bom notar que Lua é uma linguagem que diferencia minúsculas de maiúsculas, por exemplo, and é uma palavra reservada, mas And e AND são dois nomes válidos diferentes. Como convenção, nomes que começam com um sublinhado seguido por letras maiúsculas (tais como _VERSION) são reservados para variáveis globais internas usadas por Lua.

2.1. Comentários

Um comentário em Lua começará com um hífen duplo (--) em qualquer lugar, desde que fora de uma cadeia de caracteres. Se por acaso o texto imediatamente depois do hífen duplo (--) não é uma abertura de colchete longo ([[), o comentário é um comentário curto, o qual se estende até o fim da linha. Caso contrário, ele será um comentário longo, que se estende até o fechamento do colchete longo (]]) correspondente. Comentários longos são frequentemente usados para desabilitar código temporariamente.
Além disso, vale ressaltar que ao usar os colchetes longos haverá a necessidade de usar o sinal igual (=) para definir o nível desse colchete longo, sendo que pode haver n níveis .

Exemplo:

```
-- Comentários de linha
-- [[ comentário de blocos completos de código --]]
[=[ comentários de nível 1 ]=]
```

2.2. Palavras Reservadas

As palavras reservadas são termos especiais que têm um significado pré-definido na linguagem de programação e não podem ser usados como identificadores (nomes de variáveis, funções, etc.). Elas são reservadas para serem usadas como parte da sintaxe da linguagem e geralmente têm uma função específica, como controlar o fluxo de execução do programa ou definir o escopo de variáveis.

Em lua temos as seguintes palavras reservadas:

and	**break**	**do**	**else**	**elseif**
end	**false**	**for**	**function**	**if**
in	**local**	**nil**	**not**	**or**
repeat	**return**	**then**	**true**	**until**
while				

2.3 Literais reservados

Em Lua, existem três literais reservados: **nil**, **true** e **false**.

O literal *nil* representa o valor nulo ou ausente. Ele é usado para indicar que uma variável não tem nenhum valor atribuído. O literal *true* representa o valor lógico verdadeiro.O literal *false* representa o valor lógico falso.

exp ::= nil | false | true

O literal Boolean é o tipo dos valores false e true. Entretanto, tanto nil como false tornam uma condição falsa enquanto que qualquer outro valor torna a condição verdadeira.

2.4 Operadores e Delimitadores

Operadores e Delimitadores são elementos da sintaxe de uma linguagem de programação que são usados para controlar o fluxo de execução do programa. Em Lua temos os seguintes Operadores e delimitadores.

2.4.1 Aritméticos

Operadores aritméticos são operadores matemáticos que são usados para realizar operações matemáticas em números ou variáveis. Em Lua os Operadores são:

Operador Binário	**Operação**
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo
^	Exponenciação

Operador Unário	**Operação**
+	Positivo
-	Negação

2.4.2 Relacionais

Operadores relacionais são operadores que são usados para comparar valores e determinar se uma relação matemática entre eles é verdadeira ou falsa. Em Lua, os operadores relacionais incluem:

Operador	**Operação**
==	Verifica Igualdade
~=	Verifica Diferença
<	Verifica se é menor
>	Verifica se é maior
<=	Verifica se é menor ou igual

>=	Verifica se é maior ou igual

2.4.3 Lógicos

Operadores lógicos são usados para combinar expressões lógicas e determinar se uma condição é verdadeira ou falsa. Em Lua, os operadores lógicos incluem:

Operador	Operação
and	and
or	or
not	not

2.4.4 Bit a Bit

Em Lua, você pode usar operadores bit a bit para realizar operações lógicas em cada bit individual de um número inteiro. Aqui estão os operadores bit a bit disponíveis em Lua:

Operador	Operação
&	bitwise AND
|	bitwise OR
~	bitwise exclusive OR
>>	right shift
<<	left shift
~	unary bitwise NOT

2.5 Literais String

O literal string representa cadeias de caracteres. Em Lua, cadeias de caracteres podem conter qualquer caractere de 8 bits, incluindo zeros ('\0') dentro dela.

exp ::= Cadeia

2.6 Literais Numéricas

exp ::= Numero

Em lua não haverá distinção entre os números inteiros e reais, ou seja, todos os valores numéricos são tratados da mesma maneira. Sendo assim, os códigos

```
a = 9
b = 9.0
c = 0.9e1
d = 90e-1
```

irão armazenar o valor numérico 9 nas variáveis a,b,c e d.

2.7 Literal Booleano

O literal Boolean é o tipo dos valores false e true. Entretanto, tanto nil como false tornam uma condição falsa enquanto que qualquer outro valor torna a condição verdadeira.

exp ::= nil | false | true

2.8 Identificador

Em Lua os nomes (também chamados de identificadores) podem ser qualquer conjunto de letras e underscore, desde que não comece com número, coincidindo com a definição do que é um nome em muitas linguagens. Esses nomes e identificadores são usados em variáveis e campos de tables.

Um exemplo de identificador válido é:

nil exemplo = 123

Onde nil é o tipo e exemplo representa o identificador (nome) da variável e 123 é o valor armazenado nela.

Outro exemplo aceito é:

string _nome = ‘Joao’

Onde string é o tipo da variável e _nome é o identificador (nome) da variável, sendo Joao o valor armazenado nela.

Não são aceitos identificadores que possuam caracteres especiais nem espaços, sendo assim os exemplos abaixo demonstram identificadores não aceitos:

string &nome
nil 78opcao
number numero daconta

2.9 Variáveis

[escopo][tipo de variável][nome da variável] = [Valor padrão]

Variáveis em Lua são espaços de memória que armazenam valores. Em Lua, você pode atribuir valores a variáveis usando o sinal de igualdade (=). Além disso, as variáveis em Lua não têm tipo fixo, o que significa que você pode armazenar qualquer tipo de valor em uma variável, incluindo números, strings, tabelas e funções, mas estão associados aos valores armazenados nas variáveis, por exemplo:

a = “Exemplo” a armazena string

b = 1.23 b armazena um número
b = nil b armazena nil
a = 5 a armazena número

Aqui estão algumas regras para nomes de variáveis em Lua:

1. Deve começar com uma letra ou sublinhado.
2. Pode conter letras, números e sublinhados.
3. Não pode conter espaços.
4. Não pode ser uma palavra-chave.

As variáveis globais em Lua não precisam ser declaradas (visto no exemplo acima), por quando escrevemos a = 5, a variável a é, por default, uma variável global. Entretanto se desejarmos que uma variável tenha um escopo local, devemos declará-la previamente usando a palavra reservada local (Exemplo abaixo).

```
local a
a = 5
```

3. Referências

Manual de referências de Lua 5.2. Disponível em <Manual de Referência de Lua 5.2>. Acessado em 07 de Fevereiro de 2023.