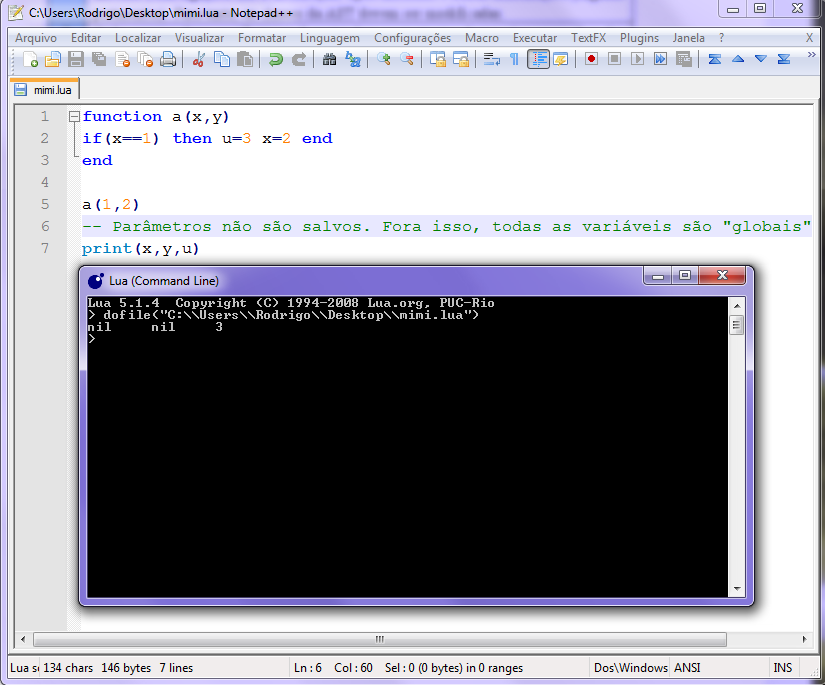
|  |  |
| --- | --- |
| Restrições contextuais – “LUA” | |
| 1 | Todos os identificadores precisam ter sido “declarados” (algum valor/definição atribuído) antes de serem utilizados. |
| 2 | Break só pode ser utilizado dentro do escopo de um while. |
| 3 | Todos os operadores devem ser aplicados a operandos do mesmo tipo. |
| 4 | Em A = B, o tipo de B precisa ser igual ao tipo de A e A é uma variável. |
| 5 | Return só pode ser utilizado dentro do escopo de uma função. |
| 6 | A expressão na condição de um if e while precisa ser do tipo int. |
| 7 | Operadores +, -, \* e / devem ser aplicados a operandos int ou float. |
| 8 | Operadores +, -, \* e / retornam o tipo dos seus operandos. |
| 9 | Operadores ==, !=, >, <, >=, <= devem ser aplicados a operandos int, float. |
| 10 | Operadores ==, !=, >, <, >=, <= retornam valor de tipo int. |
| 11 | A linguagem é *case sensitive*. |
| 12 | Não pode haver mais de um identificador (nome de função) com o mesmo *spelling*. |

**Comentários:**



- Qualquer variável da gente é global, tirando os parâmetros passados nas funções.

- Em lua não tem definição de tipo. No nosso, a gente vai assumir que se for definido a=3, a vai ser inteiro pra sempre! Então não vamos poder fazer depois disso um a=3.2!

- Na definição de funções não é dito se tem ou não retorno e nem qual o tipo dele. Então não vai existir nenhuma restrição dizendo algo a respeito de retorno de funções. **(Exceto a restrição 5)**

- Não precisamos definir uma função main para o código executar. Código em qualquer canto pode ser executado.

- A **restrição 3** está correta se considerarmos boolean. **Caso não consideremos,** vamos ter:

Operadores +, -, \* e / devem ser aplicados a operandos int ou double.

Operadores +, -, \* e / retornam o tipo dos seus operandos.

Operadores ==, !=, >, <, >=, <= devem ser aplicados a operandos int, double.

Operadores ==, !=, >, <, >=, <= retornam inteiros 1 ou 0 que representam a mesma ideia de verdadeiro e falso.

- **Vai que é tua! ...**