Python 解释器手册

2019 程序设计 OOP 大作业

实现内容：一个 python 语言子集的解释器

具体要求：

1. 程序结构：

   从上到下逐行执行

2. 文法规则：

   2.1 源文件编码

   ACSII 编码，区分大小写。中文字符是未定义的.

   2.2 关键字

   None True False def return break continue if elif else while or and not

   关键字不可作为变量名或函数名

   2.3 空白字符的处理

   空格、制表符在源文件中可以区分词素（Token），同时在一行的开头可以表示缩进 。换行符表示新的语句的开始。

   2.4 注释

   从# 开始到本行结束的内容都会被作为注释。

   2.5 标识符

   标识符的第一个字符必须是英文字母，第二个 字符开始可以是英文字母、数字或者下划线。标识符区分大小写。长度超过 64 个字符的标识符是未定义的.

   2.6 常量

   2.6.1 逻辑常量

   True 为真，False 为假

   2.6.2 整数常量

   整数常量以十进制表示。整数常量不设负数，负数可以由正数取负号得到。整数的范围是没有限制的，这意味着你必须实现高精度整数。首位为 0 的整数常量是未定义的。

   2.6.3 字符串常量

   字符串常量是由双引号或单引号括起来的字符串。

   可以由两个字符串拼接而形成新的字符串常量。如“123”“456”相当于“123456”

   字符串中的所有字符必须是可示字符（printable character），空格中的一种。

   2.6.4 空值常量

   None 用来表示变量没有指向任何值。

3. 运算符

   3.1 算术运算符

   +,-,*,/

   在字符串意义下+表示拼接，* a 表示拼接 a 次

   3.2 关系运算符

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

   连续比较：若出现 1<2>3 这样连续的关系运算符，将它拆成相邻的比较并用 and 连接，如(1<2) and (2<3)

   3.3 逻辑运算符

   and or not

   与标准 python 语法不同，这里只返回布尔类型

and 的定义如下：先计算运算符左边的值，转为 bool 类型，若为 false，返回 false，否则计算运算符右边的值，转为 bool 类型，并返回它。

or ， not 的定义可类比

3.4 赋值运算符

=

给一个变量赋值的意义是将这个变量指向右值，右值不管类型。

对于一个之前未定义的变量，赋值运算会先定义变量。

与标准 python 不同：

全局变量的生效范围是全部范围（不用 global 关键字即可访问），局部变量的生效范围是在当前语句块（被缩进和取消缩进包起来的部分），具体局部变量和全局变量划分规则和 c++一样。

如 a=1,a="123",a=1.1 这 3 条语句依次执行，再输出 a，结果是 1.1

可以连等，如 a=b="123",意思是 b="123",a=b 这两条语句依次执行

可以给多变量赋值，如 a,b=1,2 意思是 a=1,b=2 依次执行

3.5 增量赋值

+=，-=，*=，/=

对字符串而言+=就是往后加字符，*=就是把字符串复制多遍加起来，剩下两个符号对字符串无定义

3.6 圆括号

()

圆括号除了用在表达式里，可以用来调用函数，定义函数。

3.7 优先级

= < or < and < not < comparison < +,- < *,/ < ()

4. 数据类型

bool 只有 True False 两个值

int 高精度整数

float c++中的 double

str 字符串，immutable

5. 语句

5.1 变量定义/赋值语句

变量名=(变量名=…变量名=)值

语法规则参见 3.4

5.2 表达式语句

变量名+=值

运算时如果两个运算数类型不一样，按照 c++规则执行自动类型转换

5.3 条件语句

```
if expression1:
    语句块
elif expression2:
    语句块
else expression3:
    语句块
```

elif(相当于 else if) , else 可以没有

5.4 循环语句

```
while expression :
    语句块
```

5.5 跳转语句

break ,return ,continue

6. 函数

6.1 函数定义

def func_name (parameters):

语句块

参数列表如 a,b,c，变量名之间用逗号分隔,可以为空。有些变量可以有默认值，但都必须出现在无默认值的变量后面，如(a,b=1)合法，但(a=1,b)不合法

6.2 函数调用

函数名（传入参数列表） 如 foo(1,2)

函数调用必须出现在该函数定义之后。

参数有两种形式：keyword 和 positional

Keyword argument 比如 foo(a=1,b=2)表示传入参数 a 的值为 1，b 的值为 2。

Positional argument 是指 foo(1,2)这样按照出现顺序指代参数。

注意!! 若一个参数列表中同时有两种参数形式，positional argument 必须出现在 keyword argument 之前，如 foo(1,b=2)

6.3 内建函数

print 输出 可以有任意个参数，逐个输出，中间用空格分隔。输出后换行。输出 float 时保留 6 位小数。输出字符串不要输出前后的引号。如 print("123",1.0) 请输出 123, 1.000000

int 将 float 或 bool 转成 int

float 将 int 或 bool 转成 float

str 将 int 或 float 或 bool 转成 int

bool 将 int 或 float 或 str 转成 bool，对于 str，如果是""，为 false，否则为 true

都只有一个参数

6.4 函数递归

限制递归层数 2000 层

7. 评分相关

本次作业要求用 OOP 完成，若不按照要求最多得一半分。

助教组会提供一些数据，只要通过全部，即可得到满分。若有一些数据无法通过，按通过比例给分。部分数据已经给在了 testData 文件中，其他数据被隐去了。这意味着你必须自己写测试数据给自己测。如果你对自己的数据是否满足要求有疑问，请及时向助教提问。千万不要尝试通过 hack 的方法获取数据，一经发现零分处理。

ddl 截止后会安排 code review，根据代码是否符合要求酌情扣分。如有实现实用功能或在手册之外的语法（比如容器，自定义类等），可酌情加分。

8. Hint

建议使用 antlr4 框架完成，助教组已经提供了现成的语法文件。

若你想挑战自己，可以自己写 parser 或者自己重写语法文件。

9. 其他

如果有不懂的文法规则参考 python3.g4，以上面的规则为准

其他如果有问题的可以及时联系助教明确语法。