Lua实现类创建对象：

local Account={balance = 0}

function Account:new(o)

o=o or {}

setmetatable(o,self)

self.\_\_index=self

return o

end

实现私密性

function newAccount(initialBalance)

local self={

balance=initialBalance,

haha =10

}--私有变量

--私有方法

local withdraw=function(v)

self.balance=self.balance+v

end

local deposit=function(v)

self.balance=self.balance+v

end

return {

deposit=deposit--放入接口变成共有方法

}

End

弱引用表，如果一个数据只被弱引用了 则会被回收

a={}

--回收key

b={\_\_mode="k"}

--字符串，数字，布尔 不会从弱引用的table中删除

setmetatable(a,b)

a[1]=10

--数字不会被回收

collectgarbage()

print(a[1])--10

local key = {}--key被引用着 所以不会被回收

a[key]=10

collectgarbage()

print(a[key])--10

a[2]={}

collectgarbage()

print(a[2])--table...

--回收value

b.\_\_mode="v"

collectgarbage()

print(a[2])--nil

b.\_\_mode="kv"

string库

local str = "A122df"

print(str:upper())--A122DF

print(string.upper(str))--A122DF

print(str:lower())--a122df

print(string.lower(str))--a122df

print(string.rep(str,3))--a122dfa122dfa122df

print(string.char(97))--a

print(string.char(97,98,99))--abc

print(string.byte("abc"))--97

print(string.byte("abc",2))--98

print(string.byte("abc",-2))--99

print(string.byte("abc",2,-1))--98 99

print(string.format("abc%.4f",3.1415926))--3.1416

print(string.sub(str,5,6))--df

print(string.sub(str,5,-1))--df

print(string.sub(str,5))--df

print(string.find("abchellode","hello"))--4 8

print(string.match("abchellode","hello"))--"hello"

print(string.gsub("abchellodhelloe","hello","w"))--"abcwdwe" 2

--把句子中的单词收集到word中

local word = {}

local xx = "im a grate man"

for w in string.gmatch(xx,"%a+") do

word[#word+1]=w

end