

GF3002

脚本语言

刘志伟

刘志伟

C与lua交互之函数调用

实验1

余庆祥

2220631136

BIH

2022/11/20

|  |  |
| --- | --- |
| **1、实验名称** | **实验3 C与lua交互之函数调用** |
| **2、实验日期** | **(2022年11月20日)** |
| **3、实验目的** | |
| 1. 理解迭代器的概念。 2. 掌握迭代器的编写。 3. 掌握泛型for的原理及应用。 | |
| **4、实验环境** | |
| win7以上兼容系统，lua解释器，VS2019 | |
| **5、实验的步骤和方法** | |
| **(将你设计程序的步骤详细描述，例如关键代码编写顺序说明，设计出来的类图，相关类成员的说明等等)**  先创建一个虚拟栈，加载相关的库文件，执行指定lua文件，获取lua文件的全局变量，c++执行结果。 | |
| **6、数据记录和计算** | |
| **（代码编写思路及关键代码）**  第一个功能代码：  Cpp文件代码：  #include <lua.hpp>  #include <iostream>  using namespace std;  void func(lua\_State\* L, int& a, int& b, int& c, int& d)  {  lua\_getglobal(L, "luafunc");  if (lua\_pcall(L, 0, 4, 0) != 0)  {  luaL\_error(L, "error running function 'func': %s", lua\_tostring(L, -1));  }  d = (int)lua\_tointeger(L, -1);  c = (int)lua\_tointeger(L, -2);  b = (int)lua\_tointeger(L, -3);  a = (int)lua\_tointeger(L, -4);  lua\_pop(L, 4);  }  double twoNumberSquareSum(lua\_State\* L, double x, double y)  {  lua\_getglobal(L, "twoNumberSquareSum");  lua\_pushnumber(L, x);  lua\_pushnumber(L, y);  if (lua\_pcall(L, 2, 1, 0) != 0)  {  luaL\_error(L, "error running function 'twoNumberSquareSum': %s", lua\_tostring(L, -1));  }  if (!lua\_isnumber(L, -1))  {  luaL\_error(L, "function 'twoNumberSquareSum' must return a number.");  }  double z = lua\_tonumber(L, -1);  lua\_pop(L, 1);  return z;  }  double maximumValue(lua\_State\* L, int argnum, ...)  {  lua\_getglobal(L, "maximumValue");  va\_list argptr;  va\_start(argptr, argnum);  for (int i = 0; i < argnum; i++)  {  double a = va\_arg(argptr, double);  lua\_pushnumber(L, a);  }  va\_end(argptr);  if (lua\_pcall(L, argnum, 1, 0) != 0)  {  luaL\_error(L, "error running function 'maximumValue': %s", lua\_tostring(L, -1));  }  if (!lua\_isnumber(L, -1))  {  luaL\_error(L, "function 'findmax' must return a number", lua\_tostring(L, -1));  }  double z = lua\_tonumber(L, -1);  lua\_pop(L, 1);  return z;  }  int main()  {  lua\_State\* pluaState = luaL\_newstate();  luaL\_openlibs(pluaState);  if (luaL\_dofile(pluaState, "func.lua"))  luaL\_error(pluaState, "\'%s\'", lua\_tostring(pluaState, -1));  int a, b, c, d;  func(pluaState, a, b, c, d);  cout << a << ", " << b << ", " << c << ", " << d << endl;  cout << "两个数的平方和：" << twoNumberSquareSum(pluaState, 2, 3) << endl;  cout << "任意个数的最大值：" << maximumValue(pluaState, 4, 0.5, 0.3, 0.09, 0.91) << endl;  lua\_close(pluaState);  return 0;  }  Lua文件代码：  function luafunc()  return 1, 2, 3, 4;  end  function twoNumberSquareSum(x, y)  return x^2 + y^2;  end  function maximumValue(...)  local t = {...}  local max = -math.huge  for i, v in ipairs(t) do  max = v  end  return max  End  第四题代码：  Cpp文件代码：  #include <lua.hpp>  #include <iostream>  using namespace std;  void SecondFun(lua\_State\* L, double\* result, int argnum, ...)  {  lua\_getglobal(L, "SecondFun");  va\_list argptr;  va\_start(argptr, argnum);  for (int i = 0; i < argnum; i++)  {  double a = va\_arg(argptr, double);  lua\_pushnumber(L, a);  }  va\_end(argptr);  if (lua\_pcall(L, argnum, 2, 0) != 0)  {  luaL\_error(L, "error ", lua\_tostring(L, -1));  }  if (!lua\_isnumber(L, -1))  {  luaL\_error(L, "error ", lua\_tostring(L, -1));  }  for (int i = 0; i < 2; i++)  {  result[i] = lua\_tonumber(L, -1);  lua\_pop(L, 1);  }  }  int main()  {  lua\_State\* pluaState = luaL\_newstate();  luaL\_openlibs(pluaState);  if (luaL\_dofile(pluaState, "func.lua"))  luaL\_error(pluaState, "\'%s\'", lua\_tostring(pluaState, -1));  double result[2];  SecondFun(pluaState, result, 3, 4.0, 5.0, 6.0);  cout << "和：" << result[1] << endl;  cout << "积：" << result[0] << endl;  lua\_close(pluaState);  return 0;  }  Lua文件代码：  function SecondFun(...)  local t = {...}  local sum = 0  local product = 1  for i, v in ipairs(t) do  sum = sum + v  product = product \* v  end  return sum, product;  end  第五题代码：  Cpp文件代码：  #include <lua.hpp>  #include <iostream>  using namespace std;  int C\_findmax(lua\_State\* L)  {  double x = lua\_tonumber(L, -2);  double y = lua\_tonumber(L, -1);  double z = x > y ? x : y;  lua\_pushnumber(L, z);  return 1;  }  int main()  {  lua\_State\* pluaState = luaL\_newstate();  luaL\_openlibs(pluaState);  lua\_register(pluaState, "findmax", C\_findmax);  if (luaL\_dofile(pluaState, "func.lua"))  {  luaL\_error(pluaState, "\'%s\'", lua\_tostring(pluaState, -1));  }  lua\_close(pluaState);  return 0;  }  Lua文件代码：  print(findmax(0.5, 0.23))  第六题代码：  Cpp文件代码：  #include <lua.hpp>  #include <iostream>  using namespace std;  int C\_add(lua\_State\* L)  {  double x = lua\_tonumber(L, -2);  double y = lua\_tonumber(L, -1);  double z = x + y;  lua\_pushnumber(L, z);  return 1;  }  int C\_sub(lua\_State\* L)  {  double x = lua\_tonumber(L, -2);  double y = lua\_tonumber(L, -1);  double z = x - y;  lua\_pushnumber(L, z);  return 1;  }  int main()  {  lua\_State\* pluaState = luaL\_newstate();  luaL\_openlibs(pluaState);  lua\_register(pluaState, "c\_add", C\_add);  lua\_register(pluaState, "c\_sub", C\_sub);  if (luaL\_dofile(pluaState, "func.lua"))  {  luaL\_error(pluaState, "\'%s\'", lua\_tostring(pluaState, -1));  }  lua\_close(pluaState);  return 0;  }  Lua文件代码：  print("两个数的和为：", c\_add(0.5, 0.23))  print("两个数的差为：", c\_sub(0.5, 0.23))  思考题代码：  Cpp文件代码：  #include <lua.hpp>  #include <iostream>  using namespace std;  int C\_add(lua\_State\* L)  {  double z = 0;  int i = 1;  while (true)  {  if (!lua\_isnumber(L, -i))  break;  z += lua\_tonumber(L, -i);  i++;  }  lua\_pushnumber(L, z);  return 1;  }  int main()  {  lua\_State\* pluaState = luaL\_newstate();  luaL\_openlibs(pluaState);  lua\_register(pluaState, "c\_add", C\_add);  if (luaL\_dofile(pluaState, "func.lua"))  {  luaL\_error(pluaState, "\'%s\'", lua\_tostring(pluaState, -1));  }  lua\_close(pluaState);  return 0;  }  Lua文件代码：  print("多个数的和为：", c\_add(1.0, 2.0, 3.0, 4.0)) | |
| **7、实验结果或结论（**实验结果怎么样？你从这个实验你学会了什么？得出了什么结论？） | |
| **（实验结果截图+文字说明）**  第一个功能效果图：    第四题效果截图：    第五题效果截图：    第六题效果截图：    思考题效果截图： | |
| **8、备注或说明**  **）** | |
| **（可写上实验成功或失败的原因，实验后的心得体会、建议等。）**  **第四题一开始失败的原因：**  **①在输入可变参数时，给的是int型常量，而要求输入的是double型，导致一直计算有误；**  **②**在函数里返回指针，这是一个错误的行为，函数里指针指向的是存放局部变量的堆栈内存空间，函数结束后将内容销毁，导致返回的指针一直指向不到内容。 | |