

GF3002

脚本语言

刘志伟

刘志伟

读写table数据

实验1

余庆祥

2220631136

BIH

2022/11/13

|  |  |
| --- | --- |
| **1、实验名称** | **实验2 读写table数据** |
| **2、实验日期** | **2022/11/13** |
| **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)** | |
| **6、数据记录和计算** | |
| **（代码编写思路及关键代码）**   1. 第一题关键代码：   Cpp文件代码：  #include <iostream>  #include <lua.hpp>  using namespace std;  double getfield(lua\_State\* L, const char\* key);  void error(lua\_State\* L, const char\* fmt, ...)  {  va\_list argp;  va\_start(argp, fmt);  vfprintf(stderr, fmt, argp);  va\_end(argp);  lua\_close(L);  system("pause");  exit(EXIT\_FAILURE);  }  int main()  {  lua\_State\* pluaState = luaL\_newstate(); // 固定步骤  luaL\_openlibs(pluaState); // 固定步骤  if (luaL\_loadfile(pluaState, "color.lua") || lua\_pcall(pluaState, 0, 0, 0))  {  error(pluaState, "omg read invalid!");  }  lua\_getglobal(pluaState, "background");  if (!lua\_istable(pluaState, -1)) // 判断是否为 table 数据类型  {  error(pluaState, "attempt to call a not TABLE value!");  }  float red = getfield(pluaState, "r");  float green = getfield(pluaState, "g");  float blue = getfield(pluaState, "b");  cout << "red: " << red << endl;  cout << "green: " << green << endl;  cout << "blue: " << blue << endl;  lua\_close(pluaState);  return 0;  }  double getfield(lua\_State\* L, const char\* key)  {  lua\_getfield(L, 1, key);  double result = lua\_tonumber(L, -1);  lua\_pop(L, 1);  return result;  }  Lua文件代码：  background = {r = 0.30, g = 0.10, b = 0}   1. 第二题关键代码：   cpp文件代码：  #include <iostream>  #include <lua.hpp>  using namespace std;  double getfield(lua\_State\* L, const char\* key);  void error(lua\_State\* L, const char\* fmt, ...)  {  va\_list argp;  va\_start(argp, fmt);  vfprintf(stderr, fmt, argp);  va\_end(argp);  lua\_close(L);  system("pause");  exit(EXIT\_FAILURE);  }  int main()  {  lua\_State\* pluaState = luaL\_newstate(); // 固定步骤  luaL\_openlibs(pluaState); // 固定步骤  lua\_newtable(pluaState); // 栈中创建table  // 向table添加(r, 0)键值对  lua\_pushnumber(pluaState, 0); // 值入栈  lua\_setfield(pluaState, -2, "r"); // 向table中添加键值对(r, 0)，并出栈  // 向table添加(g, 0)键值对  lua\_pushnumber(pluaState, 0);  lua\_setfield(pluaState, -2, "g");  // 向table添加(b, 0)键值对  lua\_pushnumber(pluaState, 1);  lua\_setfield(pluaState, -2, "b");  // 出栈，并将其table命名为全局变量BLUE  lua\_setglobal(pluaState, "BLUE");  // 执行lua脚本  luaL\_dofile(pluaState, "color1.lua");  // 读取lua脚本  lua\_getglobal(pluaState, "background"); // 将lua中的background入栈  double red, green, blue;  if (!lua\_istable(pluaState, -1)) {  error(pluaState, "attempt to call a not TABLE value!");  }  // 读bakcground  red = getfield(pluaState, "r");  green = getfield(pluaState, "g");  blue = getfield(pluaState, "b");  lua\_close(pluaState);  cout << "red: " << red << endl;  cout << "green: " << green << endl;  cout << "blue: " << blue << endl;  return 0;  }  double getfield(lua\_State\* L, const char\* key)  {  lua\_getfield(L, 1, key);  double result = lua\_tonumber(L, -1);  lua\_pop(L, 1);  return result;  }  Lua文件代码  background = BLUE | |
| **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）**   1. 第一题运行结果截图：      1. 第二题运行结果截图： | |
| **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。）** | |