**实验11 读写table数据**

**实验目的：**

（1）理解lua实现配置文件编写；

（2）掌握table数据的读写。

**实验内容：**

* **第一部分：读table数据。**在C++程序中读取lua脚本内table的每个索引值。

（1）打开记事本，写语句background = {r=0.30, g=0.10, b=0}，保存为color.lua。

（2）打开vs2019，创建一个控制台工程,将color.lua复制到项目目录下。

（3）自定义函数getfield()获取table中指定索引的值，函数原型如下：

//虚拟栈栈顶元素为table类型时，返回其索引key的值（double）

double getfield(lua\_State\* L, const char\* key);

（4）main函数中，先通过lua\_getglobal()函数将lua文件中全局变量压入栈，并检测栈中元素是否是table类型。若是，则调用自定义函数getfield()读取table各字段的值，否则，弹出错误提示。

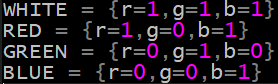
（5）运行结果如下：



* **第二部分：写table数据。**在C++程序中向lua脚本添加table。注意，先完成**写操作**，再运行脚本。

1. 编写lua配置文件。在脚本文件“color1.lua”中编写代码：background = BLUE。

接下来，用 c++ 语言在 lua 环境中创建全局变量，然后用颜色 table 来初始化这些变量，以此方法来预定义脚本中的BLUE。最终结果等价于用户在其脚本中编写如下内容：

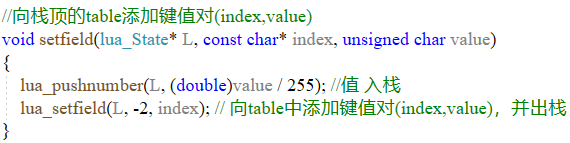


通常，C++语言中RGB使用[0,255]范围内的整数表示，而Lua 中所有数字都是实数，所以在lua中使用 [0,1]范围内的数字来表示，程序中要注意转换。

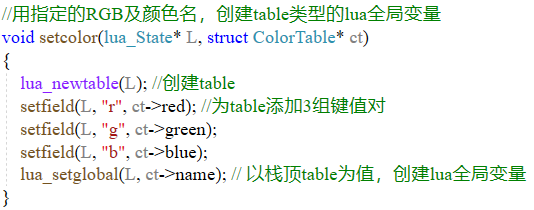
1. 定义颜色表。C++程序中，创建用于描述颜色的类型ColorTable，并预定义具有4种颜色的数组colors。具体代码如下：



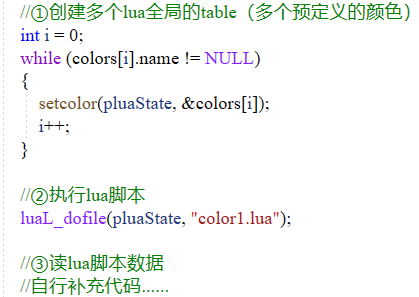
1. 自定义setfield 函数，设置 table 字段值。



1. 自定义 setcolor 函数，用指定的RGB及颜色名称，创建table类型的lua全局变量。



1. 修改 main 函数。在创建 lua 环境之后，在其中写入多个预定义的颜色值，然后执行lua脚本，并读取其中的背景色。



1. 运行程序，查看结果。
2. 【思考题】实现另一种预定义颜色的方法：不在 lua 环境中定义这些预定义的颜色，而是让用户用字符串来表示颜色名称。例如： background = "BLUE"。

**实验要求**：

（1）独立完成；

（2）win7以上兼容系统，lua解释器，VS2010以上版本；

（3）记录并分析实验结果。

**实验报告要求**：

（1）按照学院的实验报告格式编写实验报告(注意有封面的)，不能直接使用本指导书修改；

（2）指导书截图不用复制到实验报告，实验报告要附上自己的关键代码。最主要是要有合理的实验分析和总结；

（3）以“学号+姓名+实验\*”命名word文档，提交到智慧树对应位置。