**武汉纺织大学**

**《面向对象程序设计》实验报告**

**实验项目：类的声明和定义**

**成 绩：**

**学 号： 2004210523**

**姓 名： 文东柳**

**班 级： 计算机12005班**

**指导教师： 熊明福**

**报告日期： 2022年10月12日**

1. **实验内容**

1、设计一个Circle类，有数据成员radius（半径）、成员函数area()，计算圆的面积。构造一个Circle对象进行测试。

2、模仿Stack栈类，设计线性表类或循环队列类。（二选一）

1. **实验目的**

掌握类和对象的定义方法；学会正确使用public、private对类的成员进行访问控制；掌握函数成员的两种定义方法；对类的设计有初步概念（类是数据和对这些数据处理的函数成员的集合）。

1. **设计思路**

**1、**

根据题意设计类Circle，其中包含两个数据成员和一个成员函数，数据成员radius存放用户输入的圆的半径，Area用于存放半径对应的圆面积，用于向调用处进行返回，成员函数area用于计算圆的面积。在main()函数中实例化对象后，调用对象的成员函数area()传入圆的半径进行计算圆的面积，并且接收返回的值进行打印。

**2、**

设计Stack类，其中包含三个数据成员和两个成员函数，数据成员c[100]为字符串型数组，用于存放用户输入的信息；a用于存放从字符串数组中取出的元素；count用于记录目前存放在数组中的元素个数；inputString成员函数用于接收并保存用户输入的字符串至数组中；outString成员函数用于返回数组中的最后一个元素。在mian()函数中实例化类后，可通过调用对象的相关方法，实现栈的效果。

1. **核心实现代码和实现效果截图**

**1、**

// 1、设计一个Circle类，有数据成员radius（半径）、成员函数area()，计算圆的面积。构造一个Circle对象进行测试。

#include <iostream>

using namespace std;

const double PI = 3.14;

class Circle

{

private: double radius;//数据成员，用于存放圆的半径

private: double Area;//数据成员，用于存放圆的面积

public: double area(double r)//成员函数，用于获取计算圆的面积并返回给调用处。

{

radius = r;

Area = PI \* pow(radius,2);

return Area;

}

};

int main()

{

Circle c;//实例化类

double radius=0;//定义变量，接收用户输入的半径数据。

cout << "Enter radius>";

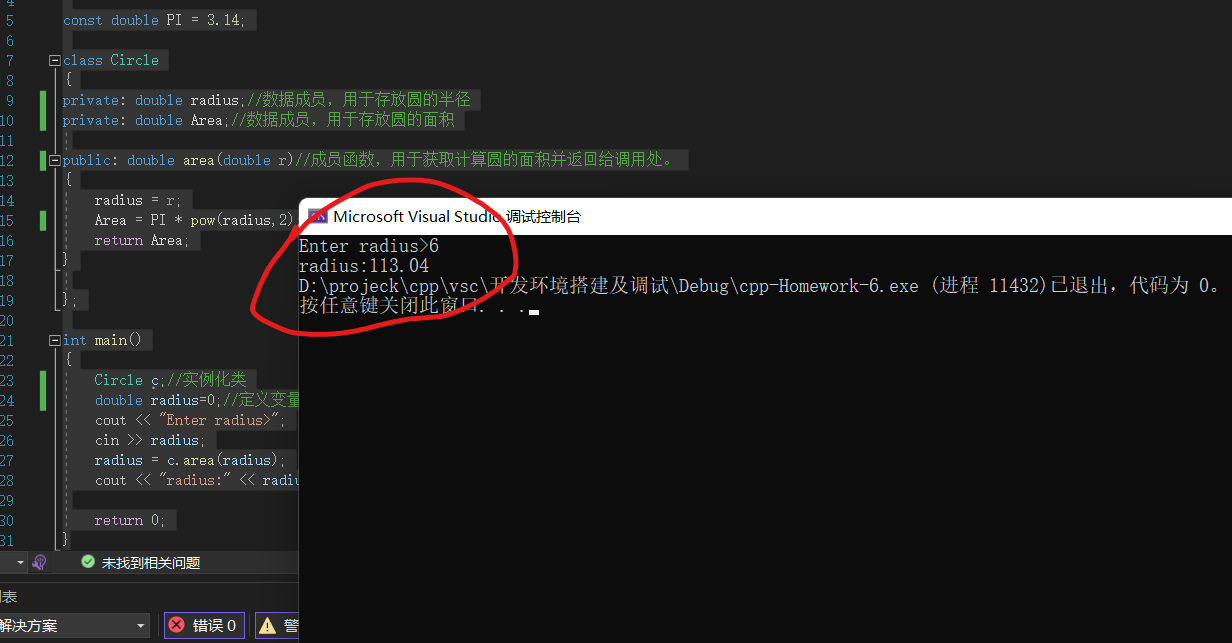
cin >> radius;

radius = c.area(radius);

cout << "radius:" << radius;

return 0;

}



**2、**

// cpp-Homework-7.cpp : 此文件包含 "main" 函数。程序执行将在此处开始并结束。

//

/\*

\* 模仿Stack栈类，设计线性表类或循环队列类。（二选一）

\*/

#include <iostream>

#include <string>

using namespace std;

class Stack

{

private:

string c[100];

string a;

int count = 0;

public:

void inputString(string s)

{

c[count] = s;

++count;

}

string outputString()

{

a = c[count-1];

//c[count].erase(count - 1, count - 1);

count=count-1;

return a;

}

};

int main()

{

Stack stack;

int i=0,count=0;

string c;

cout << "请输入你要传入的字符串数量,输入>";

cin >> i;

count = i;

for (int a = 0; a < i; a++)

{

cin >>c;

stack.inputString(c);

}

while (1)

{

int i=0;

cout << "\n输入1进行读取字符,-1结束运行>";

cin >> i;

if (i == 1 && count!=0)

{

c=stack.outputString();

cout << c;

--count;

}

else if (i == -1)

{

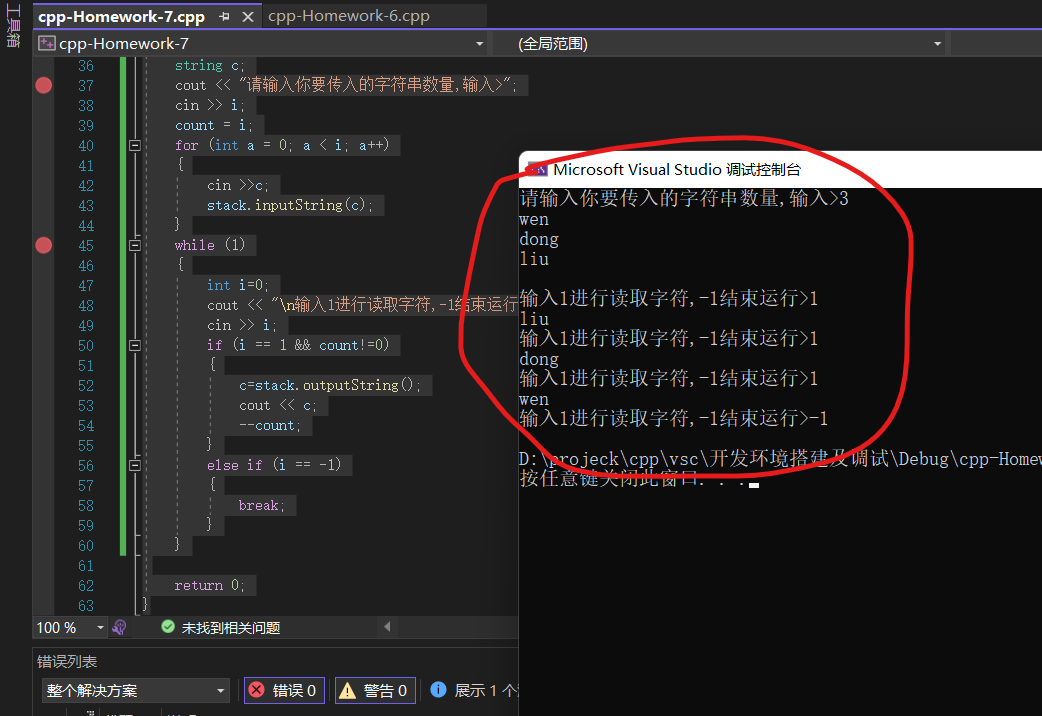
break;

}

}

return 0;

}



1. **总结**

需要对数据结构有一定掌握，了解并熟悉c++ 类与对象的基本概念，掌握基本的开发方式。代码已上传至github: <https://github.com/sky-wdl/CPP_learn>。