|  |  |
| --- | --- |
| **编号:** | **第1章第1次** |

****

信息科学与工程学院实验报告

《面向对象程序设计》

**Object-Oriented Programming**

|  |  |
| --- | --- |
| 姓名： | 朱会琛 |
| 学号： | 202311000202 |
| 班级： | 计工本2302 |
| 教师: | 张庆科 |
| 时间： | 2024年10月19日 |

**《面向对象程序设计》实验报告**

**报告要求：**实验报告包含实验目的、实验内容、实验过程（详细操作流程）、实验结果（程序运行结果高清截图）、实验分析总结五个部分。报告中若涉及代码程序，请在附录部分提供完整程序源码及源码托管地址(基于Highlight软件导入源码)。报告撰写完毕后请将PDF格式版本上传到坚果云作业提交系统。

宇文化及的Gitee链接：

https://gitee.com/yuwen-huaji/Code/tree/main/%E5%AE%9E%E9%AA%8C%E4%B8%89

1. **实验目的**

* 掌握指向常量指针的用法
* 掌握C++**重载函数**的基本用法
* 掌握C++**内联函数**的基本用法
* 掌握C++**带有默认形参值函数**的用法
* 掌握visual studio 程序调试基本方法

1. **实验内容**

#### 任务一：const与指针

建立VS项目, 设计带有**const** 常量指针的函数double triangleArea(const double \*parray)， 通过该函数计算输出三角形的面积。

要求主函数通过数组方式传递三角形边长，基于实验结果分析const在该函数中的作用。

输入输出样例：

请依次输入三条边长: 3 4 5

---------------------------

边长为3,4,5的三角形面积: 6

#### ****任务二****: 重载函数

建立VS项目, 重载函数int commonDivisor(int x1, int x2,...), 计算输出2个, 3个, 4个整数的最大公约数。

基于实验结果分析重载函数的特点和用法。

输入输出样例：

请输入3个整数: 6 15 18

---------------------------

整数6,15,18 的最大公约数为: 3

#### 任务三: 内联函数

建立VS项目, 设计实现两个功能相同的短函数（无递归和循环操作），将其中一个设置为内联函数。

要求在主函数内调用各函数10000次，计算比较两种函数各自的执行时间，基于实验结果分析内联函数和普通函数的区别。

输入输出样例：

请输入函数执行次数N: 1000000

--------------------------

普通函数执行N次时间为：xxxx

内联函数执行N次时间为：xxxx

#### 任务四：带有默认参数值的函数

建立VS项目，编写一个C++程序，定义一个函数calculateAre(...)，用于计算不同形状的面积。 函数接受两个参数：形状类型（std::string，默认值为"circle"）和相关的尺寸（double，默认值为1.0）。根据形状类型，计算并返回对应的面积。支持的形状：圆形(仅需要半径，默认长度为1.0); 矩形(需要长和宽，默认为1.0，1.0);三角形（需要三条边，默认为1.0，1.0，1.0） ，默认形状为矩形。请实现该函数，并在main函数中测试不同形状的面积计算，输出每种形状的面积。确保代码具有良好的结构和注释。

输入输出样例：

请输入要计算的图形类型(1-圆形，2-矩形，3-三角形): 3

请输入“三角形”的边长：3 4 5

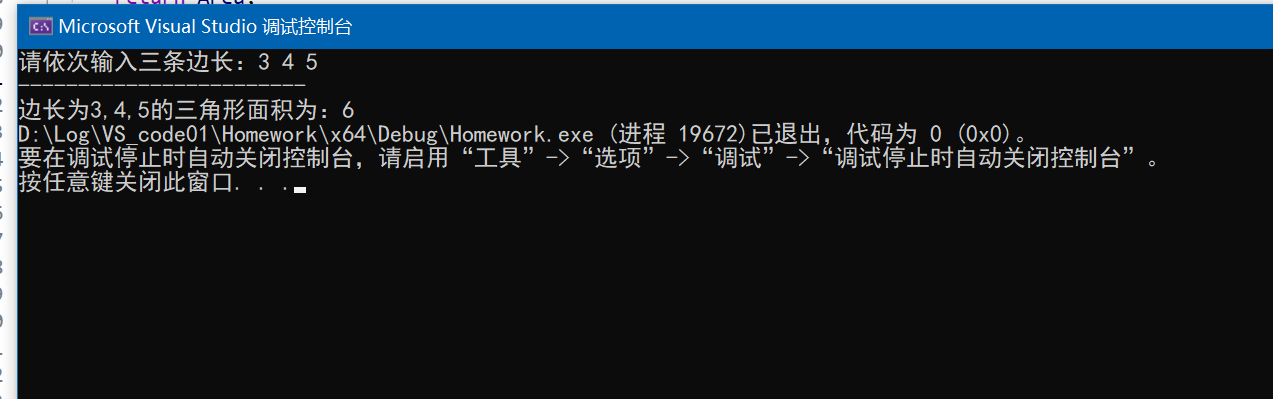
该图形为三角形，其面积为：6

1. **实验过程**

根据所学知识编写程序。

1. **实验结果**

**任务一：**

****

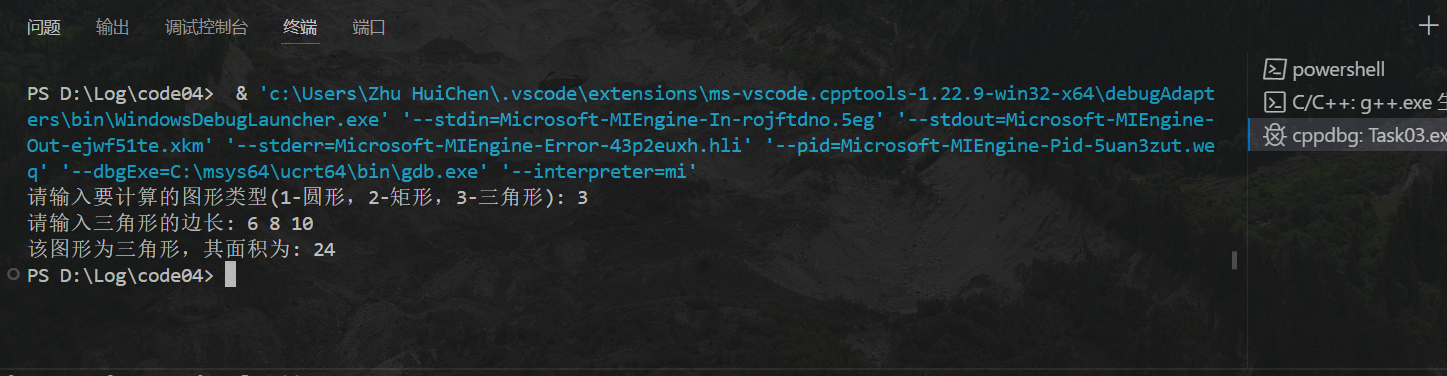
**任务二：**

****

**任务三：**

****

**任务四：**



1. **实验总结**

**答：**本次实验围绕多个 C++ 重要特性展开。通过实践，掌握了指向常量指针的用法，明确了其在限制数据修改方面的作用。同时，熟悉了重载函数、内联函数以及带有默认形参值函数的基本用法。重载函数增强了代码的灵活性和可扩展性；内联函数提高了程序执行效率；带有默认形参值函数则在函数调用时提供了便利。此外，还掌握了 visual studio 程序调试的基本方法，为后续的程序开发和问题排查奠定了基础。

* **附录：实验源代码（基于Highlight软件粘贴带有行号的源码）**

任务一：

#**include** **<**iostream**>**

**using** namespace std**;**

double triangleArea**(**const double \* parray) {

double Hc **= (**parray**[0] +** parray**[1] +** parray**[2]) /** **2;**

double Area **=** pow**(**Hc\*(Hc - parray[0]) \* (Hc - parray[1]) \* (Hc - parray[2]), 0.5);

return Area**;**

**}**

int main**() {**

double arr**[3] = {** **0** **};**

cout **<<** "请依次输入三条边长："**;**

**for** **(**int i **=** **0;** i **<** **3;** i**++) {**

cin **>>** arr**[**i**];**

**}**

cout **<<** "**------------------------**" **<<** endl **;**

cout**<<**"边长为"**<<** arr**[0]<<**',' **<<** arr**[1] <<**',' **<<** arr**[2] <<** "的三角形面积为：" **<<** triangleArea**(**arr**);**

return **0;**

**}**

任务二：

#**include** **<**iostream**>**

#**include** **<**vector**>**

#**include** **<**numeric**>**

**using** namespace std**;**

**//** 计算两个数的最大公约数

int commonDivisor**(**int x1**,** int x2**) {**

return gcd**(**x1**,** x2**);**

**}**

**//** 重载函数，计算三个数的最大公约数

int commonDivisor**(**int x1**,** int x2**,** int x3**) {**

return commonDivisor**(**commonDivisor**(**x1**,** x2**),** x3**);**

**}**

**//** 重载函数，计算四个数的最大公约数

int commonDivisor**(**int x1**,** int x2**,** int x3**,** int x4**) {**

return commonDivisor**(**commonDivisor**(**x1**,** x2**,** x3**),** x4**);**

**}**

int main**() {**

int n**;**

cout **<<** "请输入要计算的整数个数 **(2,** **3,** **4):** "**;**

cin **>>** n**;**

vector**<**int**>** numbers**(**n**);**

cout **<<** "请输入" **<<** n **<<** "个整数**:** "**;**

**for** **(**int i **=** **0;** i **<** n**; ++**i**) {**

cin **>>** numbers**[**i**];**

**}**

**//** 使用switch**,**选择整数个数

int gcd **=** **0;**

switch **(**n**) {**

**case** **2:**

gcd **=** commonDivisor**(**numbers**[0],** numbers**[1]);**

break**;**

**case** **3:**

gcd **=** commonDivisor**(**numbers**[0],** numbers**[1],** numbers**[2]);**

break**;**

**case** **4:**

gcd **=** commonDivisor**(**numbers**[0],** numbers**[1],** numbers**[2],** numbers**[3]);**

break**;**

**default:**

cout **<<** "只能计算**2,** **3,** **4**个整数的最大公约数。" **<<** endl**;**

return **1;**

**}**

cout **<<** "**---------------------------**" **<<** endl**;**

cout **<<** "最大公约数为**:** " **<<** gcd **<<** endl**;**

return **0;**

**}**

任务三：

**/**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* 功能 : 内联函数的使用

\*\* 作者 : tsingke

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#**include** **<**iostream**>**

#**include** **<**cstdlib**>**

#**include** **<**windows**.**h**>**

#**include** **<**ctime**>**

#**include** **<**cmath**>**

**using** namespace std**;**

const int **run** **=** **100000000;**

int multiply\_1**(**int x**,** int y**)**

**{**

return x\*y;

**}**

inline int multiply\_2**(**int x**,** int y**)**

**{**

return x\*y;

**}**

**/**\*----------------------------------\*

Main **Function**

\*-----------------------------------\*/

int main**()**

**{**

int t**=0;**

int a **=** **1;**

int b **=** **2;**

**/**\*----------------------------------\*

普通函数调用时间统计

\*-----------------------------------\*/

int ss\_1**,** ss\_2**;**

clock\_t start\_1**,** end\_1**;**

start\_1 **=** clock**();**

**while** **(**t**<run)**

**{**

ss\_1 **=** multiply\_1**(**a**++,** b**++);**

**//**cout **<<** a **<<** "^**2** **+** " **<<** b **<<** "^**2** **=** " **<<** ss **<<** endl**;**

t**++;**

**}**

end\_1 **=** clock**();**

cout **<<** "普通函数执行总时间**:** " **<<1.0**\*(end\_1 - start\_1) << endl;

**/**\*----------------------------------\*

内联函数调用时间统计

\*-----------------------------------\*/

clock\_t start\_2**,** end\_2**;**

start\_2 **=** clock**();**

t **=** **0;**

**while** **(**t **<** **run)**

**{**

ss\_2 **=** multiply\_2**(**a**++,** b**++);**

**//**cout **<<** a **<<** "^**2** **+** " **<<** b **<<** "^**2** **=** " **<<** ss **<<** endl**;**

t**++;**

**}**

end\_2 **=** clock**();**

cout **<<** "内联函数执行总时间**:** " **<<** **1.0**\*(end\_2 - start\_2) << endl;

system**(**"pause"**);**

return **0;**

**}**

任务四：

#**include** **<**iostream**>**

#**include** **<**string**>**

#**include** **<**cmath**> //** 用于 **sqrt**

**using** namespace std**;**

**//** 计算圆形的面积

double calculateArea**(**const string**&** shape **=** "rectangle"**,** double dimension1 **=** **2.0,** double dimension2 **=** **2.0,** double dimension3 **=** **3.0) {**

**if** **(**shape **==** "circle"**) {**

return M\_PI \* dimension1 \* dimension1; // 半径为 dimension1

**}** **else if** **(**shape **==** "rectangle"**) {**

return dimension1 \* dimension2; // 长和宽

**}** **else if** **(**shape **==** "triangle"**) {**

**//** 使用海伦公式计算三角形面积

double s **= (**dimension1 **+** dimension2 **+** dimension3**) /** **2; //** 半周长

return **sqrt(**s \* (s - dimension1) \* (s - dimension2) \* (s - dimension3)); // 面积

**}**

return **0.0; //** 默认返回**0**

**}**

int main**() {**

int choice**;**

cout **<<** "请输入要计算的图形类型**(1-**圆形，**2-**矩形，**3-**三角形**):** "**;**

cin **>>** choice**;**

**if** **(**choice **==** **1) {**

double radius**;**

cout **<<** "请输入圆形的半径**:** "**;**

cin **>>** radius**;**

double area **=** calculateArea**(**"circle"**,** radius**);**

cout **<<** "该图形为圆形，其面积为**:** " **<<** area **<<** endl**;**

**}** **else if** **(**choice **==** **2) {**

double length**,** width**;**

cout **<<** "请输入矩形的长和宽**:** "**;**

cin **>>** length **>>** width**;**

double area **=** calculateArea**(**"rectangle"**,** length**,** width**);**

cout **<<** "该图形为矩形，其面积为**:** " **<<** area **<<** endl**;**

**}** **else if** **(**choice **==** **3) {**

double a**,** b**,** c**;**

cout **<<** "请输入三角形的边长**:** "**;**

cin **>>** a **>>** b **>>** c**;**

double area **=** calculateArea**(**"triangle"**,** a**,** b**,** c**);**

cout **<<** "该图形为三角形，其面积为**:** " **<<** area **<<** endl**;**

**}** **else** **{**

cout **<<** "无效的选择。" **<<** endl**;**

**}**

return **0;**

**}**