

浙江工业大学

C++程序设计实验报告



实验题目 类、对象

学 号 302023315399

姓 名 郑华展

班 级 软件工程 06

提交日期 2024. 03. 27

一、 实验目的：

1. 掌握函数重载、函数模板的定义和使用
2. 掌握文件的输入输出
3. 掌握单链表的创建、遍历和节点的删除

二、 实验内容：

1. 需要求 3 个长方形的体积，请编写一个基于对象的程序。数据成员包括 length(长)、width(宽)、height(高)。要求用成员函数实现以下功能：

- (1) 由键盘分别输入 3 个长方形的长、宽、高；
- (2) 计算长方柱的体积；
- (3) 输出 3 个长方柱的体积。

2. 找出一个整型数组中的元素的最大值

3. 定义类 Calculator 实现计算器的常用功能。

例如：加、减、乘、除、平方根、常用对数、自然对数、幂、正弦、余弦等。

设计主函数使程序能正常运行。

注意：可以用 C++提供的库函数来求解常用对数、自然对数等

4. 设计一个类 CDateInfo，要求其满足如下要求。

- (1) 要求有一个无参数的构造函数，其初始的年月日分别为 2024,3,21，并输出“无参构造函数被调用”。
- (2) 要求有一个带参数的构造函数，其参数分别对应年月日，并输出“有参构造函数被调用”。
- (3) 要求用一个成员函数实现日期的设置。
- (4) 要求用一个成员函数实现日期的获取，并输出获取的日期。
- (5) 要求有一个析构函数输出“析构函数被调用”。
- (6) 编写一个主函数，使程序能正常运行

三、实验过程

1. 需要求 3 个长方形的体积，请编写一个基于对象的程序。数据成员包括 length(长)、width(宽)、height(高)。要求用成员函数实现以下功能：

- (1) 由键盘分别输入 3 个长方形的长、宽、高;
- (2) 计算长方柱的体积;
- (3) 输出 3 个长方柱的体积。

【程序代码】

```
h CubicVolume.cpp

#include <iostream>
using namespace std;
class node {
private:
    double length, width, height;

public:
    node() { length = 0, width = 0, height = 0; }
    void set_data(double l, double w, double h) {
        length = l, width = w, height = h;
    }
    double v() { return length * width * height; }
};

int main() {
    node v[4];
    double l, w, h;
    for (int i = 1; i <= 3; i++) {
        cin >> l >> w >> h;
        v[i].set_data(l, w, h);
    }
    for (int i = 1; i <= 3; i++) {
        cout << v[i].v() << endl;
    }
    return 0;
}
```

codetoimg.com

【运行结果】

```
// INPUT
```

```
12.0 15.3 6.9
```

```
1 3 6
```

```
888 5656 9594
```

```
//OUTPUT
```

```
1266.84
```

```
18
```

```
4.81861e+10
```

【程序分析】

首先，定义了一个名为 `node` 的类，其中包含私有成员 `length`、`width` 和 `height`，分别表示长度、宽度和高度。公有部分包含了一个默认构造函数 `node()`，以及一个用于设置长度、宽度和高度的成员函数 `set_data(double l, double w, double h)`，以及一个计算体积的成员函数 `v()`。

2. 找出一个整型数组中的元素的最大值

【程序代码】

```
h MaxnElement.cpp

#include <iostream>
#include <vector>
using namespace std;
class node {
private:
    int *arr, num;
    int maxn = -2e9;

public:
    node(int* a, int s) {
        arr = a;
        num = s;
    }
    int max(int n, int m) { return n > m ? n : m; }
    int findMax() {
        for (int i = 0; i < num; i++) {
            maxn = max(arr[i], maxn);
        }
        return maxn;
    }
};

int main() {
    // int mp[10010];
    vector<int> mp;
    int tmp, ind = 0;
    while (cin >> tmp) {
        // mp[ind++] = tmp;
        mp.push_back(tmp);
    }
    node Arr(&mp[0], mp.end() - mp.begin());
    cout << Arr.findMax() << endl;
    return 0;
}
```

codetoimg.com

【运行结果】

// INPUT

1 1254 3154 645 8 845 9 6 13 0 456 1

// OUTPUT

3154

【程序分析】

首先定义了一个名为 `node` 的类，其中包含了私有成员 `arr`、`num` 和 `maxn`。
`arr` 是一个指向整数的指针，`num` 表示数组的大小，`maxn` 用于存储最大值。

`node` 类的构造函数接受一个整数数组和其大小作为参数，并将其存储在 `arr` 和 `num` 中。`max(int n, int m)` 函数用于比较两个整数并返回较大的那个数。

`findMax()` 函数用于在数组中寻找最大值，它通过遍历数组并使用 `max()` 函数更新 `maxn` 的值，最后返回找到的最大值。

3. 定义类 `Calculator` 实现计算器的常用功能。

例如：加、减、乘、除、平方根、常用对数、自然对数、幂、正弦、余弦等。

设计主函数使程序能正常运行。

注意：可以用 C++ 提供的库函数来求解常用对数、自然对数等

【程序代码】

```
Calculator.cpp

#include <cmath>
#include <iostream>
#include <map>
using namespace std;

class Calculator {
private:
    double x, y;
    string cal;

public:
    Calculator(double x, string c, double y) {
        this->x = x;
        cal = c;
        this->y = y;
    }
    double add(double a, double b) { return a + b; }
    double subtract(double a, double b) { return a - b; }
    double mul(double a, double b) { return a * b; }
    double divide(double a, double b) {
        if (b == 0) {
            cout << "Error: Division by zero!" << endl;
            return -1;
        }
        return a / b;
    }
    double squareRoot(double x) {
        if (x < 0) {
            cout << "Error: Square root of negative number!" << endl;
            return -1; // 返回不是一个数字
        }
        return sqrt(x);
    }

    // 常用对数
    double commonLog(double x) { return log10(x); }

    // 自然对数
    double naturalLog(double x) { return log(x); }

    // 幂
    double power(double base, double exponent) { return pow(base, exponent); }

    // 正弦
    double sine(double angle) { return sin(angle); }

    // 余弦
    double cosine(double angle) { return cos(angle); }
};
```

```

int main() {
    map<string, double (Calculator::*)(double, double)>
        doubleFunctionMap_couple;
    map<string, double (Calculator::*)(double)> doubleFunctionMap_single;

    doubleFunctionMap_couple["+"] = &Calculator::add;
    doubleFunctionMap_couple["-"] = &Calculator::subtract;
    doubleFunctionMap_couple["*"] = &Calculator::mul;
    doubleFunctionMap_couple["/"] = &Calculator::divide;
    doubleFunctionMap_couple["^"] = &Calculator::power;

    doubleFunctionMap_single["sqrt"] = &Calculator::squareRoot;
    doubleFunctionMap_single["Log10"] = &Calculator::commonLog;
    doubleFunctionMap_single["Ln"] = &Calculator::naturalLog;

    doubleFunctionMap_single["sin"] = &Calculator::sine;
    doubleFunctionMap_single["cos"] = &Calculator::cosine;

    double a, b;
    string cal;
    cin >> a >> cal >> b;

    Calculator calc(a, cal, b);

    if (doubleFunctionMap_single.find(cal) != doubleFunctionMap_single.end()) {
        cout << (calc.*(doubleFunctionMap_single[cal]))(a) << endl;
    } else if (doubleFunctionMap_couple.find(cal) !=
        doubleFunctionMap_couple.end()) {
        cout << (calc.*(doubleFunctionMap_couple[cal]))(a, b) << endl;
    } else {
        cout << "Unknown operation!" << endl;
    }

    return 0;
}

```

codetoimg.com

【运行结果】

//INPUT

12 + 2

12 * 12

12 / 2

12 - 2

12 ^ 2

// OUTPUT

14

144

6

10

144

【程序分析】

- 定义了一个名为 `Calculator` 的类，包含私有成员 `x`、`y` 和 `cal`，分别表示操作数 1、操作符和操作数 2。
- 类中包含了各种数学运算和函数，如加减乘除、平方根、对数、幂、正弦和余弦等。
- `main()` 函数中创建了两个 `map` 容器，分别用于存储单目运算和双目运算的函数指针。将各种运算函数指针存储到 `map` 容器中，键值对应于操作符或函数名和相应的函数指针。
- 从标准输入中读取两个操作数和一个操作符。
- 创建一个 `node` 类型的对象 `calc`，并传入操作数 1、操作符和操作数 2。
- 使用 `if` 和 `else if` 语句根据操作符在 `map` 容器中查找相应的函数指针，并调用相应的函数进行计算。
- 如果操作符未知，则输出错误信息。

4. 设计一个类 `CDateInfo`，要求其满足如下要求。

- (1) 要求有一个无参数的构造函数，其初始的年月日分别为 2024,3,21，并输出“无参构造函数被调用”。
- (2) 要求有一个带参数的构造函数，其参数分别对应年月日，并输出“有参构造函数被调用”。
- (3) 要求用一个成员函数实现日期的设置。
- (4) 要求用一个成员函数实现日期的获取，并输出获取的日期。
- (5) 要求有一个析构函数输出“析构函数被调用”。
- (6) 编写一个主函数，使程序能正常运行

【程序代码】


```
h Class.cpp

#include <iostream>
using namespace std;

class CDateInfo {
private:
    int year;
    int month;
    int day;

public:
    // 无参数构造函数
    CDateInfo() {
        year = 2024;
        month = 3;
        day = 21;
        cout << "无参数构造函数被调用" << endl;
    }

    // 带参数构造函数
    CDateInfo(int y, int m, int d) {
        year = y;
        month = m;
        day = d;
        cout << "有参数构造函数被调用" << endl;
    }

    // 设置日期
    void setDate(int y, int m, int d) {
        year = y;
        month = m;
        day = d;
    }

    // 获取日期并输出
    void getDate() {
        cout << "当前日期为: " << year << "-" << month << "-" << day << endl;
    }

    // 析构函数
    ~CDateInfo() { cout << "析构函数被调用" << endl; }
};

int main() {
    // 使用无参数构造函数创建对象
    CDateInfo date1;
    date1.getDate(); // 输出日期

    // 使用带参数构造函数创建对象
    CDateInfo date2(2023, 4, 15);
    date2.getDate(); // 输出日期

    // 使用成员函数设置日期
    date1.setDate(2025, 6, 8);
    date1.getDate(); // 输出日期

    return 0;
}
```

【运行结果】

// OUTPUT

无参构造函数被调用

当前日期为: 2024-3-21

有参构造函数被调用

当前日期为: 2023-4-15

当前日期为: 2025-6-8

析构函数被调用

析构函数被调用

【结果分析】

- 创建 `date1` 对象时,调用了无参构造函数,并输出了 "无参构造函数被调用"。然后使用 `getDate()` 函数输出了日期 "2024-3-21"。
- 创建 `date2` 对象时,调用了带参数构造函数,并输出了 "有参构造函数被调用"。然后使用 `getDate()` 函数输出了日期 "2023-4-15"。
- 使用 `date1.setDate(2025, 6, 8)`; 修改了 `date1` 对象的日期为 "2025-6-8", 并且使用 `getDate()` 函数输出了修改后的日期。
- 当 `main()` 函数结束时,局部对象 `date1` 和 `date2` 的生命周期结束,因此调用了它们的析构函数,并输出了 "析构函数被调用"

三、 小结与收获

1.无参数构造函数（默认构造函数）:

无参数构造函数是一个不带参数的特殊成员函数,用于创建对象时进行初始化。它没有返回值,且函数名与类名相同。

当没有提供任何参数时,将调用无参数构造函数进行对象的初始化。

在给定的问题中,要求无参数构造函数初始化年、月、日为 2024、3、21,并在构造函数内输出 "无参构造函数被调用"。

2.带参数的构造函数:

带参数的构造函数是一个有参数的特殊成员函数,用于在创建对象时根据提供的参数进行初始化。

它可以接受多个参数，并根据这些参数对对象进行初始化。

在给定的问题中，要求带参数的构造函数接受年、月、日作为参数，并在构造函数内输出 "有参构造函数被调用"。

3.成员函数：

成员函数是定义在类中的函数，用于操作类的数据成员或执行其他任务。

可以有参数，也可以没有参数。

在给定的问题中，要求定义成员函数来实现日期的设置和获取，并输出获取的日期。

4.析构函数：

析构函数是用于在对象生命周期结束时进行清理的特殊成员函数。

它没有参数，没有返回值，函数名前加上波浪号 ~，与类名相同。

当对象被销毁时（例如，超出其作用域或显式删除），将调用析构函数进行资源释放和清理。

在给定的问题中，要求析构函数输出 "析构函数被调用"。