

面向对象程序设计

实验报告

项目名称: _ 模板_

专业班级: _ 联软件 2001_

学 号: __202041030102__

学生姓名: _____张圃源

实验成绩:

批阅老师:

2021年11月20日

目录

5	实验小结	4
	4.2 类模板的创建方法	3
	4.1 函数模板的创建方法	2
4	实验过程及结果	2
3	实验说明	1
2	实验内容	1
1	目的要求	1

项目 3 模板 (2 学时)

1 目的要求

(1) 掌握函数模板的创建方法,掌握利用函数模板创建实例的方法; (2) 掌握类模板的创建方法,掌握利用类模板创建实例(对象)的方法。

2 实验内容

(1)程序编写:根据问题描述和程序的输出结果,对给出的程序代码进行修改,最终给出自己的解决方案,本次实验内容是:重载函数模板 printArray,实现 Array 的正确打印,并使用 printArray 函数模板创建实例。(2)程序调试:根据给出的存在问题的 Arithmetic 类模板的程序代码,利用类模板的相关技术,a. 修改程序中的编译错误使之能够正确地编译执行;b. 对照程序的正确输出结果,修改程序中的逻辑错误使其输出结果和给定的正确输出结果一致。修改程序中的逻辑错误使其输出结果和给定的正确输出结果一致。

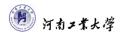
3 实验说明

人们需要编写多个形式和功能都相似的函数,因此有了函数模板来减少重复劳动;人们也需要编写多个形式和功能都相似的类,于是 C++ 引人了类模板的概念,编译器从类模板可以自动生成多个类,避免了程序员的重复劳动。C++ 中类模板的写法如下: template < 类型参数表 > class 类模板名 成员函数和成员变量;

类型参数表的写法如下: class 类塑参数 1, class 类型参数 2, ...

类模板中的成员函数放到类模板定义外面写时的语法如下: template < 类型参数表 > 返回值类型类模板名 < 类型参数名列表 >:: 成员函数名 (参数表)

用类模板定义对象的写法如下: 类模板名 < 真实类型参数表 > 对象名 (构造函数实际参数表);



如果类模板有无参构造函数,那么也可以使用如下写法:类模板名 < 真实 类型参数表 > 对象名;

4 实验过程及结果

4.1 函数模板的创建方法

```
#include <iostream>
using namespace std;
template<typename T> void Swap(T *a, T *b){
    T \text{ temp} = *a;
     *a = *b;
     *b = temp;
int main(){
     //交换 int 变量的值
     int n1 = 100, n2 = 200;
     Swap(\&n1, \&n2);
     cout\!<\!\!<\!\!n1\!<\!\!<",\llcorner"<\!\!<\!\!n2<\!\!<\!\!endl;
     //交换 float 变量的值
     float f1 = 12.5, f2 = 56.93;
    Swap(\&f1, \&f2);
     cout << f1 << ", " << f2 << endl;
     //交换 char 变量的值
     char c1 = 'A', c2 = 'B';
    Swap(\&c1, \&c2);
     cout << c1 << ", " << c2 << endl;
     //交换 bool 变量的值
     bool b1 = false, b2 = true;
    Swap(\&b1, \&b2);
     cout << b1 << ", _ " << b2 << endl;
     return 0;
}
```

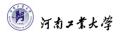
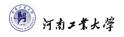


图 1: 运行截图

4.2 类模板的创建方法

```
#include <iostream>
#include <string>
using namespace std;
template <class T1, class T2>
class Pair
public:
    T1 key; //关键字
    T2 value; //值
    Pair (T1 k, T2 v): key(k), value(v) { };
    bool operator < (const Pair<T1,T2> & p) const;
};
template < class T1, class T2>
bool Pair<T1,T2>::operator < (const Pair<T1,T2> & p)
   const
//Pair的成员函数 operator <
{ //"小"的意思就是关键字小
    return key < p.key;
int main()
    Pair < string, int > student("Tom", 19); //实例化出一个类
       Pair < string, int >
    cout << student.key << "" << student.value;
    return 0;
}
```



Tom 19 ------Process exited after 0.03557 seconds with return value 0 请按任意键继续. . .

图 2: 运行截图

5 实验小结

使用类模板的目的是为了减轻程序员的工作量和增加代码的复用性。如编写一个交换 int 类型的模板,那它就不能实现交换 double 类型的功能,但类模板就能做到交换 int 类型也能交换 double 类型。