

C++语言基础

迂者 - 贺利坚

<http://blog.csdn.net/sxhelijian/>

<http://edu.csdn.net>





本节主题：

类模板

回顾：函数模板

```
int max(int a,int b,int c){  
    if(b>a) a=b;  
    if(c>a) a=c;  
    return a;  
}
```

```
long max(long a,long b,long c){  
    if(b>a) a=b;  
    if(c>a) a=c;  
    return a;  
}
```

```
double max(double a,double b,double c) {  
    if(b>a) a=b;  
    if(c>a) a=c;  
    return a;  
}
```

.....

使用虚拟类型T

```
template<typename T>  
T max(T a,T b,T c){  
    if(b>a) a=b;  
    if(c>a) a=c;  
    return a;  
}
```

数据类型参数化

类模板——参数化的类

```
template <class numtype> //虚拟类型名为numtype
class Compare{           //类模板名为Compare
public:
    Compare(numtype a,numtype b)
    {x=a;y=b;}
    numtype max( )
    {return (x>y)?x:y;}
    numtype min( )
    {return (x<y)?x:y;}
private:
    numtype x,y;
};
```

- 一类将可以用：虚拟类型参数名numtype可以替换成int、float、double、string甚至更多自定义类及类型；
- 类模板——参数化的类；

类模板的使用——参数化类

template<class numtype>

class Compare{

public:

.....

private:

numtype x,y;

};

int main(){

Compare<int> cmp1(3,7);

cout<<cmp1.max()<<" "<<cmp1.min()<<endl;

Compare<float> cmp2(45.78,93.6);

cout<<cmp2.max()<<" "<<cmp2.min()<<endl;

Compare<char> cmp3('a','A');

cout<<cmp3.max()<<" "<<cmp3.min()<<endl;

}

类模板声明：

template<class 虚拟类型参数>

class 类名 {.....}; //使用虚拟类型的类声明

用类模板定义对象的形式：

类模板名<实际类型名> 对象名;

类模板名<实际类型名> 对象名(实参表列);

在类模板外定义成员函数

template<class numtype>

class Compare{

public:

Compare(numtype a,numtype b);

numtype max();

private:

numtype x,y;

};

template<class numtype>

Compare<numtype>::Compare(numtype a,numtype b)

{x=a;y=b;}

template<class numtype>

numtype Compare<numtype>::max()

{return (x>y)?x:y;}

一般形式：

template<class 虚拟类型参数>

函数类型 类模板名<虚拟类型参数>:: 成员函数名(函数形参表列)

{...}

每个函数都用类模板

体会——抽象技术

| 抽 象 | 实 例 |
|---|--|
| 类是对象的抽象 | 对象是类的实例 |
| <pre>class Student{ private: int num; string name ... }</pre> | <pre>Student stud1, stud2,...;</pre> |
| 类模板是类的抽象 | 类是类模板的实例 |
| <pre>template<class T1,class T2> class someclass {...};</pre> | <pre>someclass<int,double> obj1; someclass<float, long> obj2; someclass<char, complex> obj3; ...</pre> |

类库中的模板

```
#include<vector>
#include <iostream>
using namespace std;
int main()
{
    int i = 0;
    vector<int> v;
    for( i = 0; i < 10; i++ )
    {
        v.push_back(i);//把元素一个一个存入到vector中
    }
    /* v.clear() 对存入的数据清空*/
    for( i = 0; i < v.size(); i++ )//v.size() 表示vector存入元素的个数
    {
        cout << v[i] << " "; //把每个元素显示出来
    }
    cout << endl;
    return 0;
}
```

Standard Template Library

标准模板库

——C++中的精华

任务：百度STL

在C++标准中，STL被组织为下面的13个头文件：

<algorithm>、<deque>、<functional>、
<iterator>、<array>、<vector>、<list>、
<forward_list>、<map>、<unordered_map>、
<memory>、<numeric>、<queue>、<set>、
<unordered_set>、<stack>和<utility>。

THANKS

本课程由 迂者-贺利坚 提供

CSDN网站：www.csdn.net
企业服务：<http://ems.csdn.net/>
人才服务：<http://job.csdn.net/>
CTO俱乐部：<http://cto.csdn.net/>
高校俱乐部：<http://student.csdn.net/>
程序员杂志：<http://programmer.csdn.net/>

CODE平台：<https://code.csdn.net/>
项目外包：<http://www.csto.com/>
CSDN博客：<http://blog.csdn.net/>
CSDN论坛：<http://bbs.csdn.net/>
CSDN下载：<http://download.csdn.net/>

