

# 计算机科学基础II

---

## 实验十五 模板

曹鹏

Email: [caopeng@seu.edu.cn](mailto:caopeng@seu.edu.cn)

Tel: 13851945861

# 实验目的

---

## 实验十五

1. 掌握函数模版的编写方法。
2. 掌握类模版的定义和使用的方法。

# 实验十五 1

## 建立求数组元素中最小值的函数模板

```
#include <iostream>
#include <string>
using namespace std;
//定义求最小值函数模板min

int ia[5]={10,7,14,3,25};
double da[6]={10.2,7.1,14.5,3.2,25.6,16.8};
string sa[5]={"上海","北京","沈阳","广州","武汉"};
int main() {
    int i=min(ia,5);
    cout << "整数最小值为: " << i << endl;
    double d=min(da,6);
    cout << "实数最小值为: " << d << endl;
    string s=min(sa,5);
    cout << "字典排序最小为: " << s << endl;
    return 0;
}
```

整数最小值为: 3  
实数最小值为: 3.2  
字典排序最小为: 北京

# 实验十五 1

建立求数组元素中最小值的函数模板

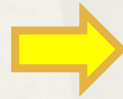
```
template <typename Groap>
Groat max(const Groap *r_array, int size)
//Groat: 类型参数
//const标识指标所指数据不可修改, 防止误改实参
{
    Groap min_val=r_array[0];
    int i;
    for (i=1;i<size; ++i)
        if(r_array[i]<min_val) min_val=r_array[i];
    return min_val;
}
```

## 实验十五 2

将实验十一的改造为模板类，使用时类型参数实例化为`char`

```
//set是类
int main(){
    int i;
    char ch[30]=.....;
    set s,s1,s2,s3,s4;

    .....
    set s5(s);
    .....
}
```



```
//set是类模板
int main(){
    int i;
    char ch[30]=.....;
    set<char> s,s1,s2,s3,s4;

    .....
    set<char> s5(s);
    .....
}
```

# 实验十五 2

//set是类

```
class set{
    char elements[maxnum];
    int num;
public:
    set(){num=0;}
    set(set& a){
        int i;
        num=a.num;
        for(i=0;i<num;i++)
            elements[i]=a.elements[i];
    }
    bool Member(char&);
    ErrCode AddElem(char&);
    void RmvElem(char&);
    void Copy(set&);
    bool Equal(set&);
    void print();
    void Intersect(set&,set&);
    ErrCode Union(set&,set&);
    bool Contain(set&);
};
```



//set是类模板

```
template<typename T> class set{
    T elements[maxnum];
    int num;
public:
    set(){num=0;}
    set(set& a){
        int i;
        num=a.num;
        for(i=0;i<num;i++)
            elements[i]=a.elements[i];
    }
    bool Member(T &);
    ErrCode AddElem(T &);
    void RmvElem(T &);
    void Copy(set&);
    bool Equal(set&);
    void print();
    void Intersect(set&,set&);
    ErrCode Union(set&,set&);
    bool Contain(set&);
};
```

# 实验十五 2

//set是类

```
bool set::Member(char& elem){
    int i;
    for(i=0;i<num;i++)
        if(elements[i]==elem)
            return true;
    return false;
}

ErrCode set::AddElem(char& elem)
{.....}

void set::RmvElem(char& elem)
{.....}

void set::Copy(set& s) {.....}

bool set::Equal(set& s) {.....}

void set::print() {.....}
```



//set是类模板

```
template<typename T>
bool set<T>::Member(T & elem){
    int i;
    for(i=0;i<num;i++)
        if(elements[i]==elem)
            return true;
    return false;
}

template<typename T>
ErrCode set<T>::AddElem(T & elem)
{.....}

template<typename T>
void set<T>::RmvElem(T & elem)
{.....}

template<typename T>
void set<T>::Copy(set& s) {.....}

template<typename T>
bool set<T>::Equal(set& s) {.....}

template<typename T>
void set<T>::print() {.....}
```



# 实验十五 2

//set是类

```
void set::Intersect(set& s1,set& s2)
{.....}
```

```
ErrCode set::Union(set& s1,set& s2)
{.....}
```

```
bool set::Contain(set& s){.....}
```

//set是类模板

```
template<typename T>
void set<T>::Intersect(set& s1,set& s2)
{.....}
```

```
template<typename T>
ErrCode set<T>::Union(set& s1,set& s2)
{.....}
```



```
template<typename T>
bool set<T>::Contain(set& s){.....}
```



---



# End