

# 面向对象程序设计(OOP)

---

Evaluation only.

Created with Aspose Slides for Java 22.7

## C++ (第8章: 群体类和群体数据的组织)

福州大学·软件学院·软件工程系  
王灿辉 (wangcanhui@fzu.edu.cn)

# 第8章：群体类和群体数据的组织

## 群体的概念

---

- 群体是指由**多个数据元素组成的集合体**。群体可以分为两个大类：线性群体和非线性群体。线性群体中的元素按位置排列有序，可以区分为第一个元素、第二个元素等。非线性群体不用位置顺序来标识元素。
- 群体类：线性群体包括：字典，哈希表，数组，记录，文件，表，栈，队列，优先级队列等，非线性群体包括树，堆，集合等。

# 第8章：群体类和群体数据的组织概述

---

➤ 本章的绝大部分内容(包括：查找算法、排序算法、可变长带边界检验的数组、链表、栈、队列等)在“算法与数据结构”课程中有更为详细的介绍。本章仅介绍部分算法和基本的数据结构。

# 第8章：群体类和群体数据的组织

## 查找：顺序查找

---

/\*功能：顺序查找（函数模板）\*/

template <class T> //返回下标值或-1

int seq\_search(T \*data, int max\_size, T value) {

//data: 数组, max\_size: 数组大小

//value: 待查找的值

for (int i=0; i<max\_size; ++i)

if (data[i]==value) return i;

return -1; //完整程序

} //问题：查找第n次出现如何实现？

# 第8章：群体类和群体数据的组织

## 查找：折半查找(递归法)

```
template <class T>
int biquery(T *data, int low, int high, T val) {
//data: 有序数组, val: 待查找的值
//low/high: 最小下标/最大下标
    int mid = (low+high)/2; //中间位置(需类型转换吗?)
    if (data[mid]==val)    return mid;
    else {
        if (low<high)
            if(val>data[mid]) biquery(data, mid+1, high, val);
            else biquery(data, low, mid-1, val);
        else return -1;
    }
} /*想找所有值, 怎么改? */
```

完整程序



# 第8章：群体类和群体数据的组织

## 排序：直接选择法

---

```
template <class T>
void sort(T *data, int max_size) {
//data: 数组, max_size: 数组大小
for (int i=0; i<max_size-1; i++) {
    int maxIndex=i;
    for (int j=i+1; j<max_size; j++)
        if (data[maxIndex]<data[j]) //改为<=?
            maxIndex=j;
    if (i!=maxIndex) //元素交换
        swap(data[i], data[maxIndex]);
}
}
```

[完整程序](#)

# 第8章：群体类和群体数据的组织

## 排序方法

---

➤ 其他排序算法，冒泡排序和插入排

序：**自学**。  
Evaluation only.  
Created with Aspose.Slides for Java 22.7.  
Copyright 2004-2022 Aspose Pty Ltd.

➤ 更多其他的排序算法，如：快速排序、堆排序、归并排序等，待“算法与数据结构”课程介绍。

# 第8章：群体类和群体数据的组织

自定义数组(模板)类：向量Vector

---

//具有下标越界检查功能，大小可以动态增长。

//可以用来代替C++语言本身的数组。

```
template<class T> class Vector {
```

```
public:
```

```
    //构造函数等.....
```

```
private:
```

```
    T *alist;    //向量首指针
```

```
    int size;    //向量大小
```

```
};
```



# 第8章：群体类和群体数据的组织

## 自定义数组(模板)类：向量Vector

---

//定义向量类中使用的一些异常类

class InvalidSize{}; //非法向量大小异常

class OutOfMem{}; //内存分配错误异常

class OutOfRange{}; //下标越界异常

int index; //记录出现异常时的下标

public:

OutOfRange(int i):index(i) {} //构造函数

int getIndex() {return index;}

};

# 第8章：群体类和群体数据的组织

## 自定义数组(模板)类：向量Vector

---

```
template <class T> class Vector {  
public:  
    Vector(int size=10); //构造函数  
    Vector(const Vector<T> &v); //拷贝构造函数  
    ~Vector() {delete[] alist;} //析构函数  
    int getSize() const {return size;}  
    T &operator[](int i); //重载下标运算符[]  
    Vector<T>& operator=(const Vector<T> &v);  
    operator T *(void) const {return alist;}  
    void Resize(int size); //修改向量的大小  
};
```

# 第8章：群体类和群体数据的组织

## 自定义数组(模板)类：向量Vector

---

```
template <class T> //构造函数
Vector<T>::Vector(int size) {
    if (size<=0) //非法向量大小
        throw InvalidSize();
    this->size=size;
    alist=new T[size]; //动态申请内存
    if (alist==NULL) //内存申请失败
        throw OutOfMem();
}
```

# 第8章：群体类和群体数据的组织

自定义数组(模板)类：向量Vector

---

//拷贝构造函数(实现“深拷贝”)

```
template <class T>
```

```
Vector<T>::Vector(const Vector<T> &v) {
```

```
    size=v.size;
```

```
    alist=new T[size];
```

```
    if (alist==NULL) throw OutOfMem();
```

```
    for (int i=0;i<size;i++)
```

```
        alist[i]=v.alist[i];
```

```
}
```

# 第8章：群体类和群体数据的组织

自定义数组(模板)类：向量Vector

---

//重载下标运算符[], 带越界检查功能!

//必须返回引用, 才可用v[i]=4等形式赋值!

template <class T>

T &Vector<T>::operator[](int i) {

if (i>=0&&i<size) return alist[i];

throw OutOfRange(i); //抛出异常!

//把出错的下标存放在异常类OutOfRange的

//index成员中, 可以报告出错下标值!

}



# 第8章：群体类和群体数据的组织

## 自定义数组(模板)类：向量Vector

---

//重载赋值运算符=，必须返回引用(允许连续赋值)

```
template <class T>
Vector<T>& Vector<T>::operator=(const Vector<T> &v) {
    if (this!=&v) {
        delete[] alist;
        size=v.size;    alist=new T[size];
        if (alist==NULL) throw OutOfMem();
        for(int i=0;i<size;i++)alist[i]=v.alist[i];
    }
    return *this;
}
```

# 第8章：群体类和群体数据的组织

## 自定义数组(模板)类：向量Vector

---

```
template <class T> //修改(扩大)向量的大小
void Vector<T>::Resize(int size) {
    if (size < this->size) throw InvalidSize();
    if (size == this->size) return;
    T * newlist = new T[size];
    if (newlist == NULL) throw OutOfMem();
    for (int i = 0; i < this->size; i++)
        newlist[i] = alist[i];
    delete[] alist;
    alist = newlist;  this->size = size;
}
```

# 第8章：群体类和群体数据的组织

## 自定义数组(模板)类：向量Vector

---

➤ 可以将自己定义的类型(向量)转换为其他数据类型(T \*)以便进行混合运算(其他数据类型转换为自己定义的类型可以使用类型转换构造函数)

➤ 定义格式为：operator 类型() { //函数体 }  
与析构函数类似，无参也无返回值！同其他函数一样，类型转换函数也可以重载。

➤ 实例：

```
operator T *(void) const { return alist; }
```

# 第8章：群体类和群体数据的组织

## 自定义数组(模板)类：向量Vector

---

//显示向量内容(测试时使用)

```
template <class T>
void Vector<T>::show(void) const {
    cout<< "向量当前的大小为:"<<size<<endl;
    for (int i=0;i<size;++i)
        cout<<"v["<<i<<"]="<<*(alist+i)<<"\t";
    cout<<endl;
}
```

# 第8章：群体类和群体数据的组织

## 自定义数组(模板)类：向量Vector

---

```
void main() {  
    const int max_size=8;  
    Vector<int> v(max_size);  
    for (int i=0;i<v.getSize();++i)  
        v[i]=2*i+1; //下标引用(带越界检查)  
    v.show();    cout<<endl;  
    Vector<int> v1(v);  
    cout<<"执行语句: Vector<int> v1(v);后, \t";  
    v1.show();    cout<<endl;  
}
```



# 第8章：群体类和群体数据的组织

## 自定义数组(模板)类：向量Vector

---

```
void main() {  
    try { v[v.getSize()]=88; }  
    catch (OutOfRangeException) {  
        cout<<"下标越界，值为："<<r.getIndex()<<"\n";  
    }  
    Vector<int> v2(max_size/2);  
    v2=v;  
    v2.Resize(max_size*2);  
}
```

# 第8章：群体类和群体数据的组织

自定义数组(模板)类：向量Vector总结

---

- 用C++的方式处理异常！
- 使用模板类进一步提高类的通用性
- 拷贝构造函数和重载赋值=(返回引用)
- 类型转换构造函数(数组和指针)
- 重载下标运算符[], 越界检查, 返回引用
- 类的使用者捕获、处理异常
- 自定义数组(模板)类：Vector完整程序

# 第8章：群体类和群体数据的组织

## 自定义栈(Stack)类(模板)

---

- 用数组实现栈(Stack)：后进先出
- 可以从表类(List)派生而来，也可定义为基类！
- 栈底固定在数组的底部，即下标0存放最早入栈的元素。
- 自定义栈(Stack)类(模板)完整程序

# 第8章：群体类和群体数据的组织

## 自定义栈(Stack)类(模板)

---

- 链表：单向、双向、循环链表等
- 队列、循环链表等
- 用链表实现线性表(List)、栈(Stack)、队列(Queue)
- 上述类的使用实例
- 以上内容自学或待后续课程介绍！

# 本章内容讲授到此结束！

---

Evaluation only.  
Created with Aspose Slides for Java 22.7.  
Copyright 2004-2022 Aspose Pty Ltd.

福州大学·软件学院·软件工程系  
王灿辉 (wangcanhui@fzu.edu.cn)