西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

C++程序设计
Programming in C++

1011018

主讲：魏英，计算机学院

字符串的处理

- 3、字符串对象
- 4、字符串对象的操作

- C++为字符串提供了一种新的自定义类型：字符串类string。
- 采用类来实现字符串，具有如下特点：
- ①采用动态内存管理，不必担心存储空间是否足够，甚至都不用有字符数组的概念；
- ②能够检测和控制诸如越界之类的异常，提高使用的安全性；
- ③封装字符串多种处理操作，功能增强；
- ④可以按运算符形式操作字符串，使用简单。
- C++程序中使用string类型，比使用C风格字符串更方便、更安全。

- 使用string类需要将其头文件包含到程序中，预处理命令为：

```
#include <string> //不能写为string.h
```

- 1. 字符串对象的定义和初始化
- 定义和初始化字符串对象，与变量的方法类似。如果string对象没有初始化则一律是空字符串。需要注意的是C++字符串对象不需要NULL字符结尾。

```
char S1[20]; //C风格字符串
string str1; //定义string对象
string sx , sy , sz; //定义多个string对象
char S2[20]="Java"; //C风格字符串初始化
string str2="Java"; //string对象复制初始化
string str3("C++"); //string对象直接初始化
```

- 2．字符串对象的引用
- 与变量类似，直接使用string对象名就表示它的引用

```
str1 = "Pascal"; //使用string对象
```

- 3. 字符串对象的输入和输出
- 可以在输入输出语句中直接使用string对象来输入输出字符串

```
cin  >> str1; //输入字符串到str1对象中存放
cout << str2; //输出str2对象中的字符串
gets(S1); //输入C风格字符串到字符数组中存放
puts(S2); //输出C风格字符串
```

- 4. 字符串对象与C风格字符串的转换

```
str1="Java";
//C风格字符串可以直接赋给string

str1.c_str();
//string转换为C风格字符串，返回char指针

str1.copy(S1,n,pos);
//把str1中从pos开始的n个字符复制到S1字符数组
```

- string对象允许使用运算符进行操作，实现类似C风格字符串的处理。如复制（strcpy）、连接（strcat）、比较（strcmp）等。

- 字符串赋值
- string对象可以使用赋值运算，其功能是字符串复制。可以将字符串常量赋给string对象

```
str1 = "Pascal"; //字符串常量复制到string对象中
strcpy(S1,"Pascal"); //C风格字符串复制到字符数组中
str1.assign(S1,n); //将C风格字符串S1开始的n个字符赋值给str1
```

- 字符串连接运算
- string对象允许使用加号（+）和复合赋值（+=）运算符来实现两个字符串连接操作。

```
str1="12" , str2="AB" , str3="CD";
str1 = str2 + str3; //str1结果为ABCD
str1 = str1 + "PHP"; //str1结果为12PHP
str1 += str3; //str1结果为12CD
```

- 字符串关系运算
- string对象可以使用关系运算符来对字符串进行比较

```
str1="ABC" , str1="XYZ";
str1 > str2; //结果为假
str1 == str2; //结果为假
str1 == "ABC"; //结果为真
```

- 其他操作
- string对象可以调用其成员函数来实现字符串处理，这里列举一些重要的操作，更详细的内容可以查阅C++标准库手册。

```
str1="ABCDEFGHIJK";
//获取字符串的长度
n = str1.size(); //n为11
n = str1.length(); //n为11
//检查字符串是否为空字符串
b = str1.empty(); //b为假
```

```
//得到子字符串
str2 = str1.substr(2,4); //从下标2开始的4个字符，str2为CDEF
//查找子字符串
n = str1.find("DEF",pos); //从pos开始查找字符串"DEF"在str1中的位置，n为3
//删除字符
str1.erase(3,5); //从下标3开始往后删5个字符，str1变为ABCIJK
//增加字符
str1.append("12345",1,3); //在str1末尾增加"12345"下标从1开始的3个字符，即"234"
//字符串替换和插入操作
str1.replace(p0,n0,S1,n); //删除从p0开始的n0个字符，然后在p0处插入字符串S1前n个字符
str1.replace(p0,n0,str2,pos,n); //删除从p0开始的n0个字符，然后在p0处插入字符串str2中pos开始的前n个字符
```

```
str1.insert(p0,S1,n); //在p0位置插入字符串S1前n个字符
str1.insert(p0,str2,pos,n); //在p0位置插入字符串str2中pos开始的前n个字符
```

- 字符串对象数组
- 可以定义字符串对象数组，即数组元素是字符串对象，定义形式与数组类似，例如：

```
string SY[5]={"Jiang","Cao","Zou","Liu","Wei"};
//定义字符串对象数组且初始化
```

SY[0]	J	i	a	n	g
SY[1]	C	a	o		
SY[2]	Z	h	o	u	
SY[3]	L	i	u		
SY[4]	W	e	i		

```
string SY[5]={"123","12","1234","1","12345"};
//长度 3,2,4,1,5

char SA[5][20]={"123","12","1234","1","12345"};
//长度均是20
```

C程序设计