第十六章之（一）string类

string类是由头文件string支持的（但是头文件string.h和cstring支持对C-风格字符串进行操纵的C库字符串函数，但不支持string类）。

不明白，实际情况如下：

string类对象可以正常使用算术运算符，但是假如使用string.h或者cstring头文件的话，无法使用cout来输出string类对象，只能使用printf("%s", a.c\_str())这样的方法进行输出。

string类具有的方法有：

①若干构造函数；

②将字符串赋给string类变量（string a="abc";）；

③比较字符串（依次判断2个字符串字符的ASCII值）；

④合并字符串；

⑤访问各个元素的重载运算符（string a="abc"; cout<<a[1]<<endl;）；

⑥用于在字符串中查找字符和字符串的工具等（不知道）。

**构造函数：**

首先，string是模板具体化basic\_string<char>的一个typedef，

因此，使用basic\_string<char> a = "aaa";和使用string a = "aaa";是一样的。

另外，也省略了内存管理的相关参数（不明白这句话指的是哪里省略了，是说不用用户自己编写相关的代码么？）。

其次，size\_type（见构造函数）是一个依赖于实现的整型，是在头文件string中定义的。——不明白

string类将string::npos定义为字符串的最大长度，通常为**unsigned int**的最大值。

另外，表格中使用了缩写NBTS（null-terminated string）来表示以空字符为结尾的字符串——即传统的C字符串。

|  |  |
| --- | --- |
| 构造函数 | 描述 |
| string(const char\*) | 将string对象初始化为s指向的NBTS |
| string(size\_type n, char c) | 创建一个包含n个元素的string对象，其中每个元素都被初始化为字符c，如string a(5,'a'); a是"aaaaa" |
| string(const string &str); | 将一个string对象初始化为string对象str（调用复制构造函数） |
| string(); | 调用默认构造函数调用一个string对象，长度为0（第一个字符是换行符） |
| string(const char\*s ,size\_type n); | 将string对象初始化为s指向的NBTS的前n个字符，即使n比s指向的字符串还长也允许。例如string a("abcd", 2);输出的2 |
| template<class Iter>  string(Iter begin,Iter end); | 将string对象初始化为区间[begin,end]内的行为就像指针，用于指定位置，范围包括begin，但不包括end |
| string(const string&str, string  size\_type pos=0,size\_type n=npos) | 将一个string对象初始化为对象str中从位置pos开始到结尾的字符，或者是从pos开始的n个字符） |
|  |  |
|  |  |