面向对象程序设计

string 字符串类

2020 年 春

耿楠

计算机科学系
信息工程学院

西北农林科技大学
NORTHWEST A&F UNIVERSITY
中国 · 杨凌

- `#include <string>`
- string 类构造函数原型
  - `string()`
  - `string(const string& rhs)`
  - `string(const string& rhs, unsigned pos, unsigned n)`
  - `string(const char *)`
  - `string(const char *s, unsigned n)`
  - `string(unsigned n, char c)`

- string 类构造函数

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
   char * S1 = "The quick brown fox jumps over the lazy dog";
   string S2 = S1;
   string S3("lazy dog");          //用字符串初始化新串
   string S4(S3);                  //利用已存在的串 S3 初始化新串
   string S5(S2, 4, 15);           //利用已存在的串 S3 初始化新串
   string S6(S1, 19);              //利用已存在的字符数组初始化新串
   string S7(50, '-');
   cout << S7 << endl;
   cout << "S2=" << S2 << endl;
   cout << "S3=" << S3 << endl;
   cout << "S4=" << S4 << endl;
   cout << "S5=" << S5 << endl;
   cout << "S6=" << S6 << endl;
   cout << S7 << endl;
}
```

- string 类成员函数
  - **length,size**：返回字符串长度
  - **append, push_back**：添加新串到本字符串末尾
  - **assign**：字符串选择赋值
  - **insert**：字符串插入函数
  - **substr**：返回子字符串
  - **find, rfind**：字符串查找，未找到返回 **string::npos**
  - **replace**：字符串替换
  - **swap**：交换两个字符串

- 使用 string 类成员函数查找字符串并替换

```cpp
#include <iostream>
#include <string>
using namespace std;

int main()
{
   string text("I like C++, I like to code in C++.");
   string newstr, sstr;
   int pos;
   cout << "Input string and new string:";
   cin >> sstr >> newstr;
   if((pos = text.find(sstr)) == string::npos)
      cout << sstr << " not found in \"" << text << "\"" << endl;
   else
   {
      cout << "old string: " << text << endl;
      text.replace(pos, sstr.length(), newstr);
      cout << "new string: " << text << endl;
   }
   return 0;
}
```

- 使用 string 类成员函数查找字符串并替换

```cpp
#include <string>
#include <iostream>
using namespace std;

int main()
{
   string originText("I like C++, I like to code in C++.");          //文本
   string text = originText;
   string newstr, sstr;
   int pos;                                                          //存放查找到串的位置
   cout << "Input string and new string:";
   cin >> sstr >> newstr;
   if((pos = text.find(sstr)) == string::npos)                       //未查找到
      cout << sstr << " not found in \"" << text << "\"" << endl;
   else
      text.replace(pos, sstr.length(), newstr);

   while((pos = text.find(sstr, pos + 1)) != string::npos)
      text.replace(pos, sstr.length(), newstr);

   cout << "01234567890123456789012345678901234567890" << endl;
   cout << originText << endl;
   cout << text << endl;
   return 0;
}
```

- string 类操作符
  - +
  - =
  - +=
  - ==, !=, < , <=, >, >=
  - []
  - <<
  - >>

- string 类操作符

```cpp
#include <iostream>
#include <string>
using namespace std;

int main()
{
   string str1("I like C++!");
   string str2("I like Java!");
   string str3;
   int pos;

   if(str1 < str2)
      str3 = str1 + str2;
   else
      str3 = str2 + str1;
   pos = str3.rfind("+");
   str3.replace(pos + 1, 7, " and");

   cout << str3 << endl;
   return 0;
}
```

- string 类迭代器
  - `string::iterator`
  - `begin()`, `end()`, `rbegin()`, `rend()`
- string 类泛型算法
  - `copy`, `reverse`, `sort` 等

string 类迭代器与泛型算法

```cpp
#include <iostream>
#include <string>
#include <algorithm>

using namespace std;

void print(char c)
{
   cout << c;
}

int main()
{
   string s1 = "abcdefghijklmnopqrstuvwxyz";

   cout << s1 << endl;
   reverse(s1.begin(), s1.end());
   cout << s1 << endl;
   sort(s1.begin(), s1.end());
   for_each(s1.begin(), s1.end(), print);

   return 0;
}
```

- 操作
  - `unsigned copy(char *s, unsigned pos=0) const;`
  - `const char *c_str() const;`
  - `const char *c_str() const;`

结尾符

string 类串不以`'\0'`结尾，使用 `c_str()` 函数时自动添加结尾符。

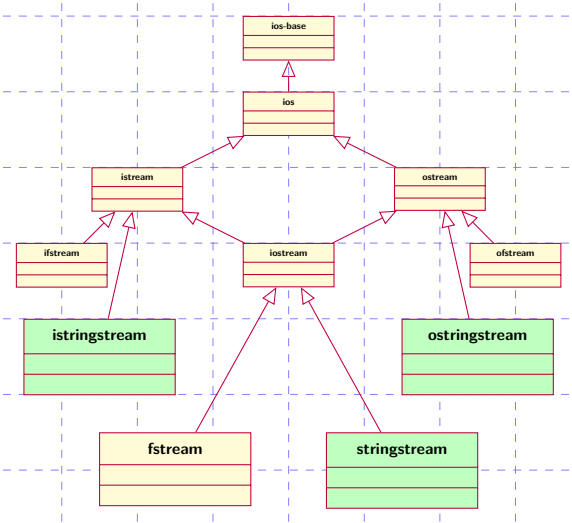
提取单词

```
char* strtok(char* str, const char* delimiters);
```

```
#include <iostream>
#include <string>
#include <cstring>
using namespace std;

int main()
{
   string s1 = "The quick brown fox jumps over the lazy dog!";
   char *c_str = NULL;
   char *p;

   c_str = (char *)s1.c_str();
   p = strtok(c_str, " !");
   while(p != NULL)
   {
      cout << strlen(p) << '\t' << p << endl;
      p = strtok(NULL, " !");
   }
   return 0;
};
```

- #incude <sstream>
  - istringstream
  - ostringstream
  - istream& getline (istream& is, string& str, char delim);

- 输入输出流类层次结构图



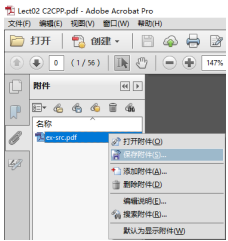
- 字符串到数字的转换 (=atoi)

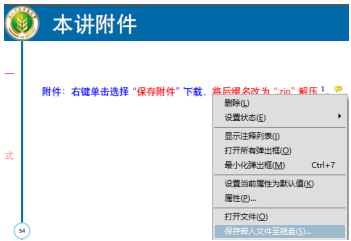
```cpp
#include <iostream>
#include <cstdlib>
#include <string>
#include <sstream>
using namespace std;

int main()
{
   int year, month, day;
   string strDate = "2010 12 31";
   istringstream streamInfo(strDate);
   streamInfo >> year >> month >> day;
   cout << year << month << day;
   return 0;
};
```

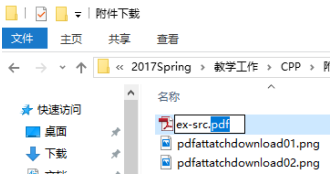
- 数字到字符串的转换 (=itoa)

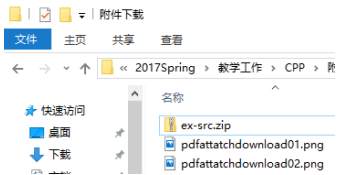
```
#include <iostream>
#include <cstdlib>
#include <string>
#include <sstream>
using namespace std;

int main()
{
   int year = rand() % 2011;
   int month = rand() % 13;
   int day = rand() % 29;

   string strDate;
   ostringstream streamInfo;
   streamInfo << year << '/' << month << '/' << day;
   strDate = streamInfo.str();
   cout << strDate;
   return 0;
};
```

附件：右键单击该链接，选择“保存附件”下载，将后缀名改为“.zip”解压[1] [2]。









[1] 请退出全屏模式后点击该链接。

[2] 以 Adobe Acrobat Reader 为例。

本讲结束，谢谢!
欢迎多提宝贵意见和建议

西北农林科技大学
NORTHWEST A&F UNIVERSITY
中国 · 杨凌