

20220330-C++&计算机网络

1.过程描述

2.结果输出

1.过程描述

```
1  #pragma once
2  #ifndef STRING_H
3  #define STRING_H
4
5  #include <iostream>
6  using std::ostream;
7  using std::istream;
8  class String
9  {
10 private:
11     char* str;
12     int len;
13     static int num_strings;
14     static const int CINLM = 80;
15 public:
16     String(const char* s);
17     String();
18     String(const String&);
19     ~String();
20     int length() const { return len; } //这里的const表示不能对函数体内的成员数
    据做任何改动
21
22     String& operator=(const String&);
23     String& operator=(const char*);
24     char& operator[](int i);
25     const char& operator[](int i) const;
26
27     friend bool operator<(const String& st1, const String& st2);
28     friend bool operator>(const String& st1, const String& st2);
29     friend bool operator==(const String& st1, const String& st2);
30     friend ostream& operator<<(ostream& os, const String& st);
31     friend istream& operator>>(istream& is, String& st);
32
33     static int HowMany();//使得可以访问private成员
34 };
35
36 #endif
```

```
1  #include "String1.h"
2  #include <cstring>
3  using std::cin;
4  using std::cout;
5  int String::num_strings = 0;
6
7  String::String(const char* s)
8  {
9      len = std::strlen(s);
10     str = new char[len + 1];
11     std::strcpy(str, s);
12     num_strings++;
13 }
14
15 String::String()
16 {
17     len = 4;
18     str = new char[1];
19     str[0] = '\0';
20     num_strings++;
21 }
22
23
24 String::String(const String& st)
25 {
26     num_strings++;
27     len = st.len;
28     str = new char[len + 1];
29     std::strcpy(str, st.str);
30 }
31
32 String::~String()
33 {
34     --num_strings;
35     delete[] str;
36 }
37
38 String& String::operator=(const String& st)
39 {
40     if (this == &st)
41     {
42         return *this;
43     }
44     else
45     {
```

```

46         delete[] str;
47         len = st.len;
48         str = new char[len + 1];
49         std::strcpy(str, st.str);
50         return *this;
51     }
52 }
53
54 String& String::operator=(const char* s)
55 {
56     delete[] str;
57     len = std::strlen(s);
58     str = new char[len + 1];
59     std::strcpy(str, s);
60     return *this;
61 }
62
63 char& String::operator[](int i)
64 {
65     return str[i];
66 }
67
68 const char& String::operator[](int i) const
69 {
70     return str[i];
71 }
72
73 int String::HowMany()
74 {
75     return num_strings;
76 }
77
78 bool operator<(const String& st1, const String& st2)
79 {
80     return (std::strcmp(st1.str, st2.str) < 0);
81 }
82
83 bool operator>(const String& st1, const String& st2)
84 {
85     return st2 < st1;
86 }
87
88 bool operator==(const String& st1, const String& st2)
89 {
90     return (std::strcmp(st1.str, st2.str) == 0);
91 }
92
93 ostream& operator<<(ostream& os, const String& st)

```

```

94 ▾ {
95     os << st.str;
96     return os;
97 }
98
99 istream& operator>>(istream& is, String& st) //重载了>>运算符
100 ▾ {
101     char temp[String::CINLM];
102     is.get(temp, String::CINLM);
103     if (is)
104         st = temp; //这里用到了上面=的重载
105     while (is && is.get() != '\n')
106         continue:

```

```
1  #include <iostream>
2  #include "String1.h"
3  const int ArSize = 10;
4  const int MaxLen = 81;
5
6  int main()
7  {
8      using std::cout;
9      using std::cin;
10     using std::endl;
11
12     String name;
13     cout << "Hi,What's your name?\n";
14     cin >> name;//通过重载>>, 使得可以接受String对象作为右操作符
15
16     cout << name << ", please enter up to " << ArSize << " short
sayings<empty line to quit>:\n";//重载了<<
17
18     String sayings[ArSize];//创建了一个String对象数组
19     char temp[MaxLen];
20     int i;
21     for (i = 0; i < ArSize; i++)
22     {
23         cout << i + 1 << ": ";
24         cin.get(temp, MaxLen);
25         while(cin && cin.get() != '\n')
26             continue;
27         if (!cin || temp[0] == '\0')
28             break;
29         else
30             sayings[i] = temp;//用到了运算符=的重载
31     }
32     int total = i;//这里total=10
33     if (total > 0)
34     {
35         cout << "Here are your sayings:\n";
36         for (i = 0; i < total; i++)
37             cout << sayings[i][0] << ": " << sayings[i] << endl;//用到了[]
的重载, 本来不能直接通过对象[]来访问字符串中的单个字符
38         int shortest = 0;
39         int first = 0;
40         for (i = 1; i < total; i++)
41         {
42             if (sayings[i].length() < sayings[shortest].length())
43                 shortest = i;
```

```

44         if (sayings[i] < sayings[first])//用到了<的重载。这里如果返回
           true, 说明右边排在左边的后面, 也就是从字母顺序上来说右边比左边大
45             first = i;
46     }
47     cout << "Shortest saying:\n" << sayings[shortest] << endl;
48     cout << "First alphabetically:\n" << sayings[first] << endl;
49     cout << "This program used " << String::HowMany() << " String
objects.Bye.\n";
50     }
51     else
52     {
53         cout << "No input!Bye.\n";
54     }
55     return 0;
56 }

```

2.结果输出

今天效率不太行, 原本打算把计算机网络全书看完的, 结果只看了应用层跟无线通信(蜂窝这块还跳了很多), 还差网络安全跟音视频两大章节, 要看完估计还得再画上一天时间。明天打算先开始数据结构跟算法, 这余下的两块后面抽时间再完成, 等数据结构完成之后用一天时间把计算机网络的习题做一做, 为后面TCP跟IP的C++实现预热一下。