

# 面向对象程序设计

## 实验2 CString类

55240425

屈熙宸

C语言没有字符串类型，C++提供了String类型，模拟String的封装，定义CString类

程序源码+运行结果

## Cstring.h

```
class CString {
private:
    char* str;
    int length;

public:
    // 构造函数
    CString(const char* s = "");

    // 拷贝构造函数
    CString(const CString& other);

    // 析构函数
    ~CString();

    // 重载赋值运算符
    CString& operator=(const CString& other);

    // 重载加法运算符：字符串拼接
    CString operator+(const CString& other) const;

    // 重载下标运算符：访问字符
    char& operator[](int index);

    // 获取字符串长度
    int len() const;

    // 截取子字符串，从 start 到 end (不包含 end)
    CString operator()(int start, int end) const;

    // 输出字符串
    friend ostream& operator<<(ostream& os, const CString& s);
};
```

```
#include "CString.h"
//构造函数
CString::CString(const char* s) {
    length = strlen(s);
    str = new char[length + 1];
    strcpy(str, s);
}
//拷贝构造
CString::CString(const CString& other) {
    length = other.length;
    str = new char[length + 1];
    strcpy(str, other.str);
}
//析构
CString::~CString() {
    delete[] str;
}
//重载=
CString& CString::operator=(const CString& other) {
    if (this != &other) {
        delete[] str;
        length = other.length;
        str = new char[length + 1];
        strcpy(str, other.str);
    }
    return *this;
}
//重载<<
ostream& operator<<(ostream& os, const CString& s) {
    os << s.str;
    return os;
}
```

```
//重载<<
ostream& operator<<(ostream& os, const CString& s) {
    os << s.str;
    return os;
}
//重载+
CString CString::operator+(const CString& other) const {
    CString result;
    result.length = length + other.length;
    result.str = new char[result.length + 1];
    strcpy(result.str, str);
    strcat(result.str, other.str);
    return result;
}
//重载[]
char& CString::operator[](int index) {
    return str[index];
}
//获取长度
int CString::len() const {
    return length;
}
```

```
//重载()  
CString CString::operator()(int start, int end) const {  
    int sLength = end - start;  
    char* sStr = new char[sLength + 1];  
  
    for (int i = 0; i < sLength; i++) {  
        sStr[i] = str[start + i];  
    }  
    sStr[sLength] = '\\0';  
  
    CString result(sStr);  
    delete[] sStr;  
    return result;  
}
```

You, 13分钟前 | 1 author (You)

```
1  #include <iostream>
2  #include "CString.h"
3
4  int main() {
5      CString m("software");
6      CString n(m), o;
7      o = m;
8      cout << n[4] << endl;           // 输出第4位字符
9      cout << n.len() << endl;        // 输出字符串的长度
10     o = m + n;                       // 合并字符串
11     cout << o << endl;               // 输出合并后的字符串
12     o = m(2,8);                      // 截取从2位到第8位的子字符串
13     cout << o << endl;               // 输出合并后的字符串
14     return 0;
15 }
```

You, 13分钟前 • Uncommitted changes

● PS D:\code\jlu\JLU\_OOP\_Assignments\tes

W

8

softwaresoftware

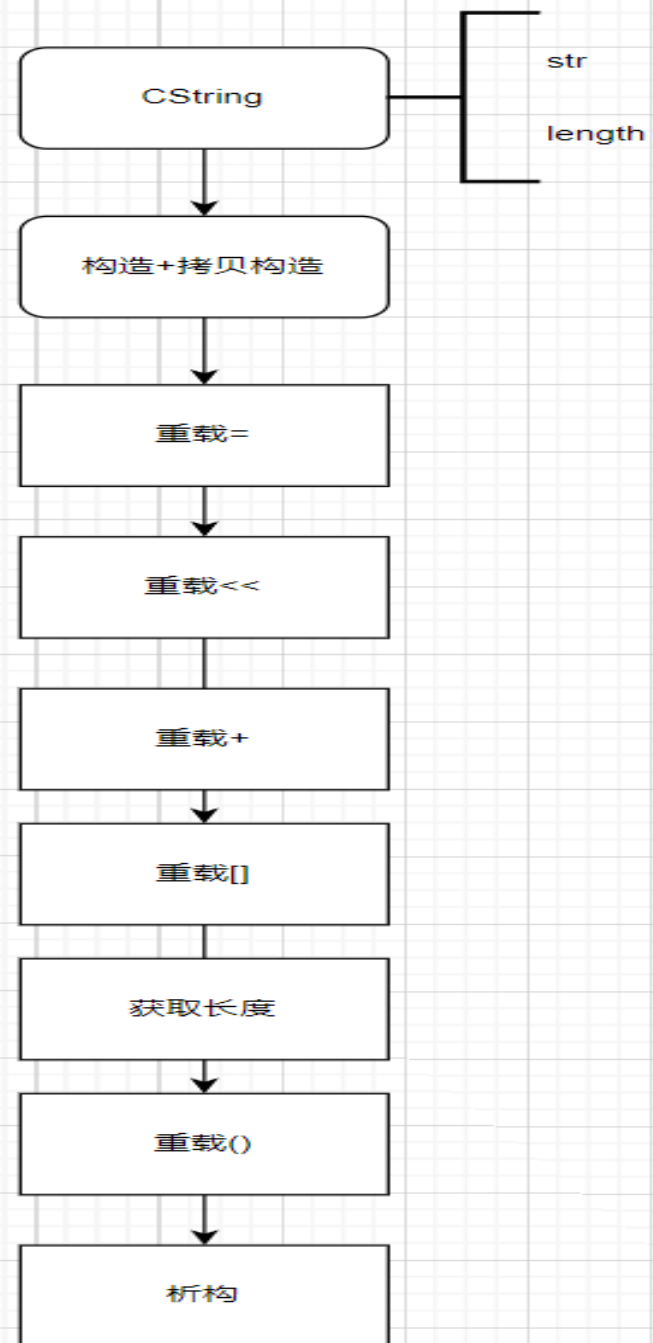
✧ ✧ ftware



# 1.分析题目要求

- 要求用cstring功能自定义一个类，并且模拟string类
- 要求自行实现构造与析构函数，重载=、<<、+、()

## 2.流程图



### 3.分析难点

- 难点在于各种运算符的重载
- 解决：
- 加深了运算符重载的相关知识
- 理解了链式结构

## 4.分析

- 1.优点:
  - 类接口清晰, 要求都得到实现
- 2.缺点:
  - 未能完全重现string的功能

## 5.收获

- 更深入的理解了类的封装
- 更深刻的认识了运算符重载
- 学会了链式编程