

2-13 特殊运算符重载及注意点_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 2-13 特殊运算符重载及注意点涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

13. 特殊运算符重载及注意点

一、赋值运算符 = 重载

将一个对象赋值给另一个对象的时候 **，会调用赋值运算符重载函数 **

```
String str1;

String str2 = str1;//String str2(str1)调用拷贝构造函数

str2 = str1;//调用赋值运算符重载函数
```

注意:

- 如果没有手动添加它，编译器会提供一个默认的赋值运算符重载函数
- C++ 强制规定，赋值运算符重载函数只能定义为类的成员函数

```
String& String::operator=(const String &other)
{
    if(this == &other){
        return *this;
    }

    if(m_data != NULL){
        delete[] m_data;
    }

    if(other.m_data != NULL){
        m_data = new char[strlen(other.m_data) + 1];
        strcpy(m_data,other.m_data);
    }else{
        m_data = other.m_data;//NULL
    }

    return *this;
}
```

二、左移运算符 << 重载

```
/*
String Str;

str << "ab" << "cd" << "123";

str.show();//"abcd123"
*/

String & String::operator<<(const char *str)
{
    int len = 0;

    char *oldStr = NULL;

    if(!str){
        return *this;
    }

    if(this->str){
        len = strlen(this->str) + 1;
        oldStr = this->str;
    }

    len += strlen(str);

    this->str = new char[len];

    if(oldStr){
        strcpy(this->str,oldStr);

        delete[] oldStr;
    }else{
        this->str[0] = '\0';
    }

    strcat(this->str,str);

    return *this;
}
```

三、不能被重载的运算符

大部分运算符都可以重载，以下运算符不能重载

	A	B
1	写法	解释
2	.	成员访问运算符
3	.* 和 ->*	成员指针运算符".*"即成员是指针类型
4	::	域运算符
5	sizeof	长度运算符
6	?	条件运算符(三目运算符)
7	#	预处理符号

四、注意点

- C++ 中大部分运算符都可以重载，只有一小部分不可以重载，其中 "="和"&" 不重载，编译器默认支持
- 重载运算符限制在 C++ 语言中已有的运算符范围内进行重载，不能创建新的运算符
- 重载之后的运算符不能改变运算符的优先级和结合性，也不能改变运算符操作数的个数及语法结构
- 重载运算符函数不能有默认的参数，否则就改变了运算符的参数个数
- 重载的运算符只能是用户自定义的类型
- 运算符重载函数可以是成员函数的形式，友元函数形式，也可以是非成员非友元的普通函数形式

五、任务

实现如下功能代码

```
String str("hello")

//通过cout输出str对象的内容:"hello"

cout << str << endl;
```

- 划线
- 写笔记

学习要认真，笔记应当先



公开笔记 0/1000 提交



Sunny_SunshineX

删除 编辑

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta，点击查看详细说明

