

2-12 运算符重载_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 2-12 运算符重载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。

12. 运算符重载

C/C++ 中提供的运算符只能针对基本类型 (int,short,char,float,double) 完成计算，不能用于我们自己构建的类类型对象之间运算。

运算符重载，就是对已有的运算符重新进行定义，赋予其另一种功能，以适应不同的数据类型。

返回类型 operator 运算符 (参数)

一般会将它声明为类的友元函数，这样它可以访问类的私有成员

```
class String{

public:

    String(const char *str = NULL,int max = 100);

    String(const String &other);

    ~String(void);

    void show(void);

    friend String operator+(const String &op1,const String &op2);

private:

    char *str;

};

String operator+(const String &op1,const String &op2)

{

    String newStr;

    if(!op1.str || !op2.str){

        cout << "Error,op1 or op2 is empty" << endl;

        return newStr;

    }

    int size = strlen(op1.str) + strlen(op2.str) + 1;

    newStr.str = new char[size];

    strcpy(newStr.str,op1.str);

    strcat(newStr.str,op2.str);

    return newStr;

}
```

- 双目运算符重载为类的成员函数时，函数只显式说明一个参数，该形参是运算符的右操作数

- ** 前置单目运算符 (++/-)** 重载为类的成员函数时，不需要显式说明参数，即函数没有形参
- 后置单目运算符 (++/-) 重载为类的成员函数时，函数要带有一个整型 (int) 形参

```

class String{

public:

    String(const char *str = NULL,int max = 100);

    String(const String &other);

    ~String(void);

    void show(void);

    String operator+(const String &op2);

private:

    char *str;

};

String String::operator+(const String &op2)

{

    String newStr;

    if(!this->str || !op2.str){

        cout << "Error,op1 or op2 is empty" << endl;

        return newStr;

    }

    int size = strlen(this->str) + strlen(op2.str) + 1;

    newStr.str = new char[size];

    strcpy(newStr.str,this->str);

    strcat(newStr.str,op2.str);

    return newStr;

}

class String{

public:

    String(const char *str = NULL,int max = 100);

    String(const String &other);

    ~String(void);

    void show(void);

    String &operator++();

    String operator++(int);

private:

    char *str;

};

```

```
String &String::operator++(void)
{
    if(!this->str){
        return *this;
    }

    for(char *p = this->str;p && *p;p++){
        *p = *p + 1;
    }

    return *this;
}

String String::operator++(int)
{
    if(!this->str){
        return *this;
    }

    QString oldStr(*this);

    for(char *p = this->str;p && *p;p++){
        *p = *p + 1;
    }

    return oldStr;
}
```

在 String 类中完成对 “[]” 运算符重载，实现下标访问。

```
String str("hello");

str[0] = 'w';
str[1] = 'x';

for(int i = 0;i < str.size();i++){
    cout << str[i] << endl;
}
```

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

