

# 11장. string 클래스 디자인

부산대학교

[ddosun@pusan.ac.kr](mailto:ddosun@pusan.ac.kr)

김미경

# 11-1 표준 string 클래스

```
#include <iostream>
#include <string.h>
using std::endl;
using std::cout;
using std::cin;

using std::ostream;
using std::istream;
```

1. 생성자, 소멸자, 복사생성자, 대입 연산자 정의
2. 문자열 전달 생성자 (디폴트 매개변수 포함)
3. + 연산자 오버로딩
4. <<연산자 오버로딩
5. += 연산자 오버로딩
6. ==연산자 오버로딩
7. >>연산자 오버로딩

```
int main()
{
    String str1="Good ";
    String str2="morning";
    Sstring str3=str1+str2;
    cout<<str1<<endl;
    cout<<str2<<endl;
    cout<<str3<<endl;

    str1+=str2;
    if(str1==str3)                //str1과 str3의 내용 비교
        cout<<"equal!"<<endl;

    String str4;
    cout<<"문자열 입력: ";
    cin>>str4;
    cout<<"입력한 문자열: "<<str4<<endl;
    return 0;
}
```

## 11-2 사용자 정의 string 클래스

```
using std::ostream;
using std::istream;

class String{
    int len; // 'W0' 포함한 문자 길이
    char* str;
public:
    String(const char* s=NULL);
    String(const string& s);
    ~String();
    String& operator=(const String& s);
    String& operator+=(const String& s);
    bool operator==(const String& s);
    String operator+(const String& s);

    friend ostream& operator<<(ostream& os, const String& s);
    friend istream& operator>>(istream& is, String& s);
};
```

## 11-2 사용자 정의 string 클래스

```
string::string(const char* s){  
    len=(s!=NULL ? strlen(s)+1 : 1);  
    str=new char[len];  
  
    if(s!=NULL)  
        strcpy(str, s);  
}
```

```
string::string(const string& s){  
    len=s.len;  
    str=new char[len];  
    strcpy(str, s.str);  
}  
string::~~string(){  
    delete []str;  
}
```

## 11-2 사용자 정의 string 클래스

```
string& string::operator=(const string& s){  
    delete []str;  
    // 메모리의 유출을 막기 위해  
    len=s.len;  
    str=new char[len];  
    strcpy(str, s.str);  
    return *this;  
    // 연속적인 대입 연산을 허용하기 위해  
}
```

```
string& string::operator=(const string& s){  
    char* tStr = new char[s.len];  
    delete[]str;  
    strcpy(tStr, s.str);  
    str = tStr;  
    len = s.len;  
    return *this;}
```

```
string string::operator+(const string& s){  
    char* tStr=new char[len+s.len-1];  
    strcpy(tStr, str);           // 문자열 복사  
    strcat(tStr, s.str);        // 문자열 추가  
  
    string temp(tStr);  
    delete []tStr;  
    return temp;  
}
```

## 11-2 사용자 정의 string 클래스

```
ostream& operator<<(ostream& os, const string& s){  
    os<<s.str;  
    return os;  
}
```

```
string& string::operator+=(const string& s){  
    len=len+s.len-1;  
    char* tStr=new char[len];  
    strcpy(tStr, str);           // 문자열 복사  
    delete []str;  
  
    strcat(tStr, s.str);         //문자열 추가  
    str=tStr;  
    return *this;  
}
```

## 11-2 사용자 정의 string 클래스

```
bool string::operator==(const string& s){  
    return strcmp(str, s.str)? false:true;  
}
```

```
istream& operator>>(istream& is, string& s){  
    char str[100];  
    is>>str;  
  
    s=string(str);  
    return is;  
}
```

## 11-3 OOP 프로젝트 4 단계

- **대입 연산자, 복사 생성자**
  - Account.h, Account.cpp
- **string 클래스의 적용**
  - 복사 생성자와 대입 연산자의 적용이 불필요