과목: 자료구조

교수: 최재현 교수님

자료구조 <과제 4>

- String 클래스의 작성 -

홍지훈

이름: 홍지훈

학과: 소프트웨어학부

분반: 나

학번 : 20201777

0. 과제

- 강의내용과 교재를 참고하여 String 클래스를 작성하고 예제코드를 작성하여 제출

1. 소스코드

1-1. String.h

#pragma once

#include <iostream>

using namespace std;

class String

{

private:

    char \*buffer;

    int length;

    int size;

    String(int m);

public:

    String();

    String(String& s); //복사생성자

    String(char\* init, int m); //길이m string init 초기화

    ~String();

    String Concat(String t);

    String& operator=(const String&); // 대입

    bool operator==(String t); // 동일한지 비교

    bool operator!(); // 공백이면 true 아니면 false

    int Length(); // 문자수 반환

    String Substr(int i, int j); // i~j사이의 string 반환

    int Find(String pat); //string에서 pat스트링을 찾아서 위치를 반환 없으면 -1

    void print();

};

1-2. String.cpp

#include "String.h"

String::String() : String(10) {}

String::String(int m)

{

    size = m;

    buffer = new char[m];

    length = 0;

}

String::~String()

{

    delete[] buffer;

}

String::String(String& s) : String(s.length + 1)

{

    for (int i = 0; i < s.length; i++)

        buffer[i] = s.buffer[i];

    buffer[s.length] = '\0';

    length = s.length;

}

String::String(char\* init, int m) : String(m + 1)

{

    for (int i = 0; i < m; i++)

        buffer[i] = init[i];

    buffer[m] = '\0';

    length = m;

}

String String::Concat(String t)

{

    String result(length + t.length + 1);

    for (int i = 0; i < length; i++)

        result.buffer[i] = buffer[i];

    for (int i = 0; i < t.length; i++)

        result.buffer[length + i] = t.buffer[i];

    result.buffer[length + t.length] = '\0';

    result.length = length + t.length;

    return result;

}

String& String::operator=(const String& s)

{

    delete[] buffer;

    buffer = new char[s.length + 1];

    for (int i = 0; i < s.length; i++)

        buffer[i] = s.buffer[i];

    buffer[s.length] = '\0';

    length = s.length;

    return \*this;

}

bool String::operator==(String t)

{

    if (length != t.length) return false;

    for (int i = 0; i < length; i++)

        if (buffer[i] != t.buffer[i])

            return false;

    return true;

}

bool String::operator!()

{

    if (length == 0)

        return true;

    return false;

}

int String::Length()

{

    return length;

}

String String::Substr(int i, int j)

{

    String result(j - i + 2);

    result.length = j - i + 2;

    for (int k = 0; k < j-i+1; k++)

        result.buffer[k] = buffer[k+i];

    result.buffer[j-i+1] = '\0';

    return result;

}

int String::Find(String pat)

{

    for (int i = 0; i + pat.length+1 < length; i++) {

        bool suc = true;

        for (int j = 0; j < pat.length; j++) {

            if (i + j >= length || buffer[i + j] != pat.buffer[j]) {

                suc = false;

                break;

            }

        }

        if (suc) return i;

    }

    return -1;

}

void String::print()

{

    for (int i = 0; i < length; i++)

        cout << buffer[i];

    cout << endl;

}

1-3. main.cpp

#include "String.h"

using namespace std;

int main(void) {

    String str1((char\*)"string", 6);

    str1.print();

    String str2((char\*)"apple", 5);

    str2.print();

    cout << "str1 and str2 is same? => " << (str1 == str2) << endl;

    str1 = str2;

    str1.print();

    cout << "str1 length = " << str1.Length() << endl;

    cout << "str1 and str2 is same? => " << (str1 == str2) << endl;

    String str3;

    cout << "str3 length = " << str1.Length() << endl;

    cout << "str3 isEmpty? => " << (!str3) << endl;

    cout << "str1 isEmpty? => " << (!str1) << endl;

    String str4((char\*)"ssusoft", 7);

    str3 = str4.Concat(str2);

    str3.print();

    cout << "str3 length = " << str3.Length() << endl;

    String str5;

    str5 = str3.Substr(5, 7);

    str5.print();

    cout << "str5 length = " << str5.Length() << endl;

    String str6((char\*)"soft", 4);

    cout << "find \"soft\" from str3 : " << str3.Find(str6) << endl;

    cout << "find \"apple\" from str3 : " << str3.Find(str1) << endl;

    return 0;

}

2. 실행 화면

