**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ВСП – Технічний коледж**

**НУ «Львівська політехніка»**

Відділення

Інформаційних технологій

Та комп`ютерної техніки

**ЗВІТ**

Про виконання лабораторної роботи №2

З предмету «Навчальна практика з основ об'єктно-орієнтованого програмування»

Виконав

Студент групи 31 ПЗ

Тисляк А.О.

Перевірив викладач:

Чистяк В.В.

Львів 2021

**Лабораторна робота № 2 «Перевантаження операторів С++»**

**Мета роботи:** ознайомитись з поняттям та призначенням дружніх функцій та конструктора копіювання. Навчитися реалізовувати перевантаження операторів.

*Варіант 10*

**Завдання:** описати клас, що реалізовує вказаний нижче тип даних. Клас повинен містити множину конструкторів для створення об'єктів певного типу (конструктор по замочуванню та з параметрами, конструктор копії) та подані у таблиці операції над об’єктами класу (плюс обов’язково операції присвоювання та порівняння) з використанням механізму перевантаження операцій.

**Завдання мого варіанту:** створити клас «рядок», який буде підтримувати аткі операції: об’єднання, копіювання.

**Код програми:**

***Файл StudentProgress.h***

#pragma once

#include <iostream>

class MyString {

private:

int len;

char\* str;

public:

MyString();

MyString(const char\* str);

MyString(const MyString& other);

MyString(MyString&& other) noexcept;

~MyString();

MyString& operator=(const MyString& other);

bool operator>(const MyString& other) const;

bool operator<(const MyString& other) const;

bool operator>=(const MyString& other) const;

bool operator<=(const MyString& other) const;

bool operator==(const MyString& other) const;

bool operator!=(const MyString& other) const;

friend MyString operator+(const MyString& a, const MyString& b);

friend std::ostream& operator<<(std::ostream& os, const MyString& obj);

friend std::istream& operator>>(std::istream& is, MyString& obj);

int length() const;

};

***Файл MyString.cpp***

#include <iostream>

#include <string>

#include "MyString.h"

MyString::MyString() : str(nullptr), len(0) {}

MyString::MyString(const char\* str) {

this->len = strlen(str);

this->str = new char[this->len];

for (int i = 0; str[i] != '\0'; i++)

this->str[i] = str[i];

}

MyString::MyString(const MyString& other) {

this->len = other.len;

this->str = new char[this->len];

for (int i = 0; i < this->len; i++)

this->str[i] = other.str[i];

}

MyString::MyString(MyString&& other) noexcept {

//std::cout << "CP CONS\n";

this->len = other.len;

this->str = other.str;

other.str = nullptr;

}

MyString::~MyString() { delete[] this->str; }

int MyString::length() const {

return this->len;

}

MyString& MyString::operator=(const MyString& other) {

if (this == &other)

return \*this;

delete[] this->str;

this->str = new char[other.len];

this->len = other.len;

for (int i = 0; i < this->len; i++)

this->str[i] = other.str[i];

return \*this;

}

MyString operator+(const MyString& a, const MyString& b) {

MyString res;

res.len = a.len + b.len;

res.str = new char[res.len];

for (int i = 0; i < a.len; i++)

res.str[i] = a.str[i];

for (int i = 0; i < b.len; i++)

res.str[a.len + i] = b.str[i];

return res;

}

std::ostream& operator<<(std::ostream& os, const MyString& object) {

for (int i = 0; i < object.len; i++)

os << object.str[i];

return os;

}

std::istream& operator>>(std::istream& is, MyString& object) {

char\* buff = new char[2500];

is.getline(buff, 2500);

object = MyString{ buff };

delete[] buff;

return is;

}

bool MyString::operator>(const MyString& other) const {

if (this->len > other.len)

return true;

else if (this->len < other.len)

return false;

for (int i = 0; i < this->len; i++) {

if (this->str[i] > other.str[i])

return true;

else if (this->str[i] < other.str[i])

return false;

}

return false;

}

bool MyString::operator<(const MyString& other) const {

if (this->len < other.len)

return true;

else if (this->len > other.len)

return false;

for (int i = 0; i < this->len; i++) {

if (this->str[i] < other.str[i])

return true;

else if (this->str[i] > other.str[i])

return false;

}

return false;

}

bool MyString::operator>=(const MyString& other) const {

if (this->len > other.len)

return true;

else if (this->len < other.len)

return false;

for (int i = 0; i < this->len; i++) {

if (this->str[i] > other.str[i])

return true;

else if (this->str[i] < other.str[i])

return false;

}

return true;

}

bool MyString::operator<=(const MyString& other) const {

if (this->len < other.len)

return true;

else if (this->len > other.len)

return false;

for (int i = 0; i < this->len; i++) {

if (this->str[i] < other.str[i])

return true;

else if (this->str[i] > other.str[i])

return false;

}

return true;

}

bool MyString::operator==(const MyString& other) const {

if (this->len < other.len)

return false;

else if (this->len > other.len)

return false;

for (int i = 0; i < this->len; i++) {

if (this->str[i] < other.str[i])

return false;

else if (this->str[i] > other.str[i])

return false;

}

return true;

}

bool MyString::operator!=(const MyString& other) const {

return !(\*this == other);

}

***Файл Main2.cpp***

#include <iostream>

#include <cstring>

#include <string>

#include "MyString.h"

int main() {

using namespace std;

MyString a, b;

cout << "Enter the first MyString obj: ";

cin >> a;

cout << "Enter the second MyString obj: ";

cin >> b;

cout << "\na == " << a << "\nb == " << b << "\n";

cout << "len a == " << a.length() << "\nlen b == " << b.length() << "\n";

cout << "\na + b == " << (a + b) << "\n";

cout << "a > b == " << (a > b) << "\n";

cout << "a < b == " << (a < b) << "\n";

cout << "a >= b == " << (a >= b) << "\n";

cout << "a <= b == " << (a <= b) << "\n";

cout << "a >= b == " << (a == b) << "\n";

cout << "a != b == " << (a != b) << "\n";

cout << "\nAfter a = b:\n";

a = b;

cout << "a == " << a << "\nb == " << b << "\n";

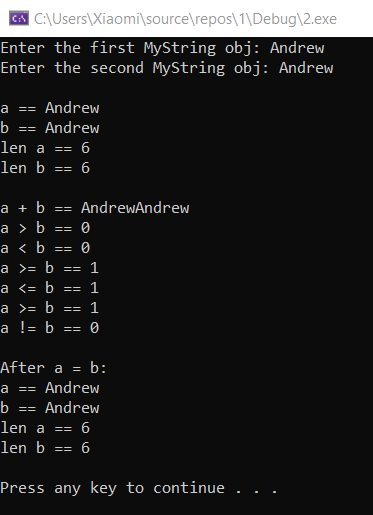
cout << "len a == " << a.length() << "\nlen b == " << b.length() << "\n\n";

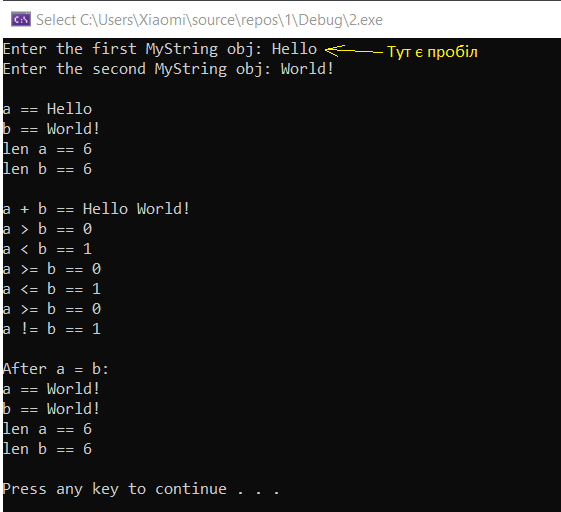
system("pause");

return 0;

}

***Скрін-шоти роботи програми:***

******

******

**Висновок**

Отже, під час виконання лабораторної роботи №2, я ознайомився з поняттям та призначенням дружніх функцій та конструктора копіювання. Навчився реалізовувати перевантаження операторів.