Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

**ЛАБОРАТОРНА РОБОТА №2**

**Тема:** Конструктори і деструктори

з дисципліни: Об’єктно-орієнтоване програмування

|  |
| --- |
| Виконав: ст. гр. КН-23 |
| Царенко С. О. |
| Перевірив: асистент  Козірова Н. Л. |
|  |

Кропивницький

2024

**Тема:** Конструктори і деструктори

**Мета:** ознайомитись з основними поняттями конструктор і деструктор в ООП та навчитись їх програмно реалізовувати мовою С++.

**Варіант 5**

**Завдання 1**

1. Реалізуйте конструктор за замовчуванням, конструктор з параметрами та копіюючий конструктор для вашого класу з лабораторної роботи 1.

• Конструктор за замовчуванням має встановлювати значення полів за замовчуванням.

• Конструктор з параметрами має приймати значення для кожного поля.

• Копіюючий конструктор має копіювати значення полів з іншого об'єкта класу.

2. Реалізуйте деструктор для класу. Деструктор має виводити повідомлення про знищення об'єкта.

3. У функції main створіть об'єкт за допомогою конструктора за замовчуванням та виведіть значення його полів.

4. Створіть новий об'єкт за допомогою конструктора з параметрами та встановіть значення для полів. Виведіть значення полів цього об'єкта.

5. Створіть ще один об'єкт і скопіюйте значення полів з першого об'єкта за допомогою копіюючого конструктора. Виведіть значення полів цього об'єкта.

6. Завершіть функцію main, що призведе до виходу з області видимості створених об'єктів і виклику їх деструкторів. Переконайтесь, що повідомлення про знищення об'єктів виводяться.

**Завдання 2**

Розробіть клас «Рядок» – String довжини 𝑛. Клас повинен містити конструктори. Реалізуйте методи для виконання конкатенації рядків, вилучення символу із заданої позиції та порівняння рядків. Створіть масив об'єктів і передайте його у функцію, яка виконує сортування рядків.

**MyString.cpp:**

#include "MyString.h"

// Конструктор

MyString::MyString(const string& input) : str(input.empty() ? "Default" : input) {}

// Метод для конкатенації рядків

MyString MyString::concatenate(const MyString& other) const {

return MyString(str + other.str);

}

// Метод для вилучення символу з заданої позиції

void MyString::removeCharAt(int pos) {

str.erase(pos, 1);

}

// Метод для порівняння рядків

bool MyString::compare(const MyString& other) const {

return str == other.str;

}

// Метод для отримання рядка

const string& MyString::getString() const {

return str;

}

// Деструктор

MyString::~MyString() {}

// Статичний метод для сортування та виведення рядків

void MyString::sortArr(vector<MyString>& strings) {

sort(strings.begin(), strings.end());

cout << "Sorted strings:" << endl;

for (const auto& str : strings) {

cout << str.getString() << endl;

}

}

// Оператор для порівняння

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

return str < other.str;

}

**MyString.h:**

#pragma once

#include <iostream>

#include <string>

#include <vector>

#include <algorithm> // для sort

using namespace std;

class MyString {

private:

string str;

public:

// Конструктор з параметром

MyString(const string& input = "");

// Метод для конкатенації рядків

MyString concatenate(const MyString& other) const;

// Метод для вилучення символу з заданої позиції

void removeCharAt(int pos);

// Метод для порівняння рядків

bool compare(const MyString& other) const;

// Метод для отримання рядка

const string& getString() const;

// Оператор для сравнения

bool operator<(const MyString& other) const;

// Статичний метод для сортування та виведення

static void sortArr(vector<MyString>& strings);

// Деструктор

~MyString();

};

**Task2.cpp:**

#include <iostream>

#include <string>

#include <vector>

#include "MyString.h"

using namespace std;

int main() {

// Створення об'єктів String

MyString s1;

MyString s2("Alex");

MyString s3("Roberto");

// За замовч

cout << "Default constructor: " << s1.getString() << endl;

// Конкатенація рядків

MyString s4 = s2.concatenate(s3);

cout << "After con: " << s4.getString() << endl;

// Вилучення символу з рядка

s4.removeCharAt(1); // Видаляємо другий символ

cout << "After delete symb: " << s4.getString() << endl;

// Порівняння рядків

bool equal = s2.compare(s3);

cout << "Strings are same?: " << (equal ? "Yes" : "No") << endl;

// Вектор для зберігання об'єктів MyString

vector<MyString> strings = { s1, s2, s3 };

// Виклик статичного методу для сортування та виведення рядків

MyString::sortArr(strings);

return 0;

}

**Скрін виконання програми:**



**Висновок:**

У ході виконання лабораторної роботи №2 “ Конструктори і деструктори ”. Я ознайомився з основними поняттями конструктор і деструктор в ООП та навчитись їх програмно реалізовувати мовою С++.