Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ №2

з дисципліни «Об’єктно-орієнтоване програмування»

**Тема:** Конструктори і деструктори

Виконав: студент гр. КН-23

**Чубенко.І.К**

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++.

Завдання

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

Варіант 6 Розробіть клас «Множина (цілих чисел)» – Set потужності 𝑛. Клас повинен містити конструктори. Реалізуйте методи для визначення належності заданого елемента множині, перетину, об'єднання та різниці двох множин. Створіть масив об'єктів і передавайте пари об'єктів у функцію, яка будує множину, що складається з елементів, які входять тільки в одну з заданих множин і повертає її в головну програму.

Завдання1

Animals.cpp

#include "Animal.h"

#include <iostream>

using namespace std;

// Конструктор за замовчуванням

Animal::Animal() : name("Невідоме ім'я"), species("Невідомий вид"), age(0) {

cout << "Конструктор за замовчуванням викликано" << endl;

}

// Конструктор з параметрами

Animal::Animal(string n, string s, int a) : name(n), species(s), age(a) {

cout << "Конструктор з параметрами викликано" << endl;

}

// Копіюючий конструктор

Animal::Animal(const Animal& other) : name(other.name), species(other.species), age(other.age) {

cout << "Копіюючий конструктор викликано" << endl;

}

// Деструктор

Animal::~Animal() {

cout << "Деструктор викликано для об'єкта з ім'ям: " << name << endl;

}

// Метод для введення імені

void Animal::setName() {

cout << "Введіть ім'я тварини: ";

cin >> name;

}

// Метод для введення виду

void Animal::setSpecies() {

cout << "Введіть вид тварини: ";

cin >> species;

}

// Метод для введення віку

void Animal::setAge() {

cout << "Введіть вік тварини: ";

cin >> age;

}

// Метод для отримання імені

string Animal::getName() const {

return name;

}

// Метод для отримання виду

string Animal::getSpecies() const {

return species;

}

// Метод для отримання віку

int Animal::getAge() const {

return age;

}

Animal.h

#ifndef ANIMAL\_H

#define ANIMAL\_H

#include <string>

using namespace std;

class Animal {

private:

string name;

string species;

int age;

public:

// Конструктор за замовчуванням

Animal();

// Конструктор з параметрами

Animal(string n, string s, int a);

// Копіюючий конструктор

Animal(const Animal& other);

// Деструктор

~Animal();

// Методи встановлення значень через введення з клавіатури

void setName(); // Встановити ім'я

void setSpecies(); // Встановити вид

void setAge(); // Встановити вік

// Методи отримання значень

string getName() const;

string getSpecies() const;

int getAge() const;

};

#endif

Main.cpp

#include <iostream>

#include <Windows.h>

#include "Animal.h"

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

// Створення об'єкта класу "Animal" за допомогою конструктора за замовчуванням

Animal myAnimal;

// Виведення значень полів за замовчуванням (без введення)

cout << "Значення за замовчуванням:" << endl;

cout << "Ім'я: " << myAnimal.getName() << endl;

cout << "Вид: " << myAnimal.getSpecies() << endl;

cout << "Вік: " << myAnimal.getAge() << endl;

// Налаштування властивостей об'єкта через введення користувачем

myAnimal.setName();

myAnimal.setSpecies();

myAnimal.setAge();

// Виведення значень полів після введення користувачем

cout << "Значення після введення користувачем:" << endl;

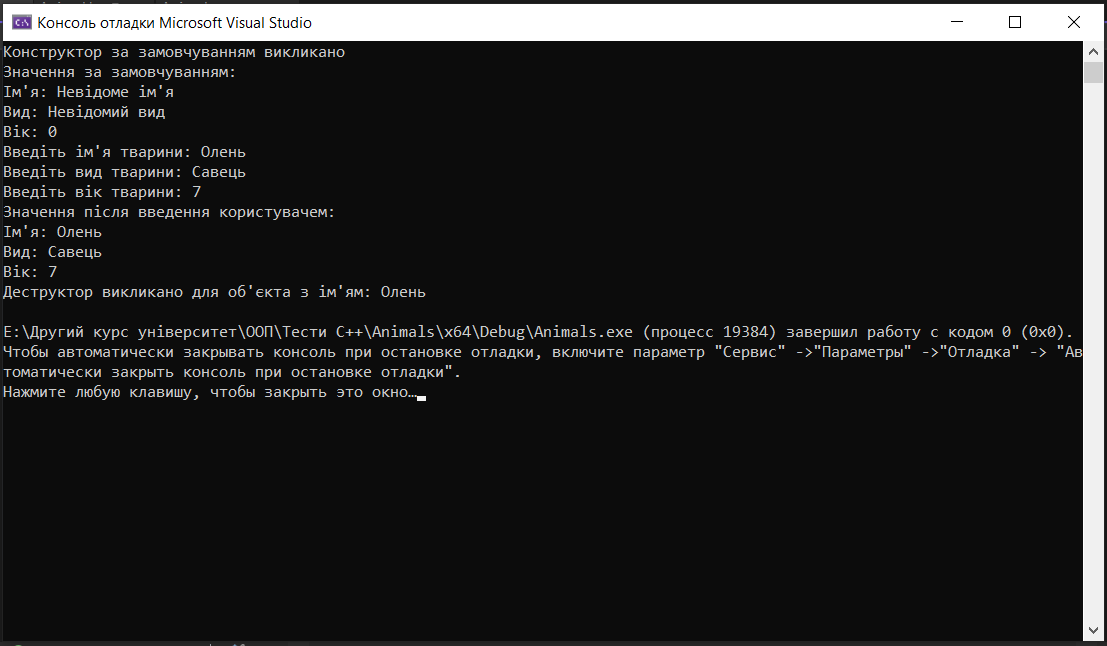
cout << "Ім'я: " << myAnimal.getName() << endl;

cout << "Вид: " << myAnimal.getSpecies() << endl;

cout << "Вік: " << myAnimal.getAge() << endl;

return 0;

}



Завдання 2

Варіант 6 Розробіть клас «Множина (цілих чисел)» – Set потужності 𝑛. Клас повинен містити конструктори. Реалізуйте методи для визначення належності заданого елемента множині, перетину, об'єднання та різниці двох множин. Створіть масив об'єктів і передавайте пари об'єктів у функцію, яка будує множину, що складається з елементів, які входять тільки в одну з заданих множин і повертає її в головну програму.

Лістинг Laba2task2.cpp

#include <iostream>

#include <cstring> // Для memset

#include <Windows.h>

const int MAX\_SIZE = 100; // Максимальний розмір множини

class Set {

private:

int elements[MAX\_SIZE]; // Масив для зберігання елементів множини

int size; // Поточний розмір множини

public:

// Конструктор за замовчуванням

Set() : size(0) {

std::memset(elements, 0, sizeof(elements)); // Ініціалізація масиву нулями

}

// Конструктор, який приймає масив і його розмір

Set(const int arr[], int n) : size(0) {

std::memset(elements, 0, sizeof(elements)); // Ініціалізація масиву нулями

for (int i = 0; i < n; ++i) {

add(arr[i]);

}

}

// Метод для додавання елемента в множину

void add(int elem) {

if (!contains(elem) && size < MAX\_SIZE) {

elements[size++] = elem;

}

}

// Метод для перевірки наявності елемента в множині

bool contains(int elem) const {

for (int i = 0; i < size; ++i) {

if (elements[i] == elem) {

return true;

}

}

return false;

}

// Метод для перетину множин

Set intersection(const Set& other) const {

Set result;

for (int i = 0; i < size; ++i) {

if (other.contains(elements[i])) {

result.add(elements[i]);

}

}

return result;

}

// Метод для об'єднання множин

Set unionSet(const Set& other) const {

Set result = \*this;

for (int i = 0; i < other.size; ++i) {

result.add(other.elements[i]);

}

return result;

}

// Метод для різниці множин

Set difference(const Set& other) const {

Set result;

for (int i = 0; i < size; ++i) {

if (!other.contains(elements[i])) {

result.add(elements[i]);

}

}

return result;

}

// Метод для симетричної різниці (елементи, які є тільки в одній із множин)

Set symmetricDifference(const Set& other) const {

Set result;

for (int i = 0; i < size; ++i) {

if (!other.contains(elements[i])) {

result.add(elements[i]);

}

}

for (int i = 0; i < other.size; ++i) {

if (!contains(other.elements[i])) {

result.add(other.elements[i]);

}

}

return result;

}

// Виведення елементів множини

void print() const {

for (int i = 0; i < size; ++i) {

std::cout << elements[i] << " ";

}

std::cout << std::endl;

}

};

// Функція для симетричної різниці двох множин

Set buildSymmetricDifference(const Set& set1, const Set& set2) {

return set1.symmetricDifference(set2);

}

int main() {

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

// Створення двох масивів для ініціалізації множин

int arr1[] = { 1, 2, 3, 4, 5 };

int arr2[] = { 4, 5, 6, 7, 8 };

// Створення двох множин

Set set1(arr1, 5);

Set set2(arr2, 5);

// Виведення множин

std::cout << "Множина 1: ";

set1.print();

std::cout << "Множина 2: ";

set2.print();

// Перетин множин

Set intersection = set1.intersection(set2);

std::cout << "Перетин: ";

intersection.print();

// Об'єднання множин

Set unionSet = set1.unionSet(set2);

std::cout << "Об'єднання: ";

unionSet.print();

// Різниця множин

Set difference = set1.difference(set2);

std::cout << "Різниця (Множина 1 \\ Множина 2): ";

difference.print();

// Симетрична різниця

Set symmetricDifference = buildSymmetricDifference(set1, set2);

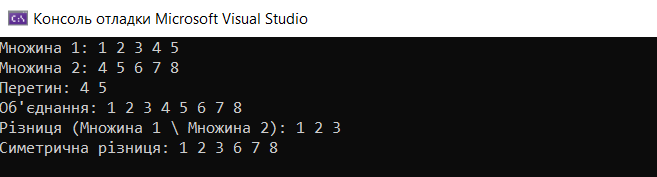
std::cout << "Симетрична різниця: ";

symmetricDifference.print();

return 0;

}

Результат:



**Висновок**

У цій лабораторній роботі було реалізовано клас із конструкторами (за замовчуванням, з параметрами та копіюючим) та деструктором. Конструктори забезпечують гнучкість у створенні об'єктів, а деструктор підтверджує правильність управління пам'яттю, виводячи повідомлення про знищення об'єктів.

У другій частині роботи був розроблений клас «Множина (цілих чисел)», який містить методи для перевірки належності елемента, перетину, об'єднання та різниці між множинами. Також реалізована функція для створення нової множини з елементів, які входять тільки в одну з двох заданих множин.

Ця лабораторна робота дозволила зміцнити знання з об'єктно-орієнтованого програмування, зокрема у використанні конструкторів, деструкторів і роботи з множинами.