Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

**Лабораторна робота №1**

**Тема:** «Конструктори і деструктори»

|  |
| --- |
| Виконав: ст. гр. КН-23 |
| Кобзар М. В. |
| Перевірила:  Козірова Наталія Леонідівна |
|  |

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

Розробіть клас «Множина (рядків)» – 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

**Варіант 8**

**Лістинг :**

*Main.cpp*

#include <iostream>

#include "Set.h"

int main() {

Set set1, set2;

std::cout << "Input elements for Set 1:" << std::endl;

set1.inputElements();

std::cout << "Input elements for Set 2:" << std::endl;

set2.inputElements();

std::cout << "Set 1: ";

set1.display();

std::cout << "Set 2: ";

set2.display();

std::cout << "Intersection: ";

set1.intersection(set2).display();

std::cout << "Union: ";

set1.unionSet(set2).display();

std::cout << "Difference (Set 1 - Set 2): ";

set1.difference(set2).display();

std::cout << "Symmetric Difference: ";

set1.symmetricDifference(set2).display();

return 0;

}

*Set.cpp*

#include "Set.h"

Set::Set() {}

void Set::addElement(const std::string& element) {

if (!contains(element)) {

elements.push\_back(element);

}

}

bool Set::contains(const std::string& element) const {

for (const auto& elem : elements) {

if (elem == element) {

return true;

}

}

return false;

}

void Set::inputElements() {

std::string input;

while (true) {

std::getline(std::cin, input);

if (input == "done") break;

addElement(input);

}

}

void Set::display() const {

for (const auto& elem : elements) {

std::cout << elem << " ";

}

std::cout << std::endl;

}

Set Set::intersection(const Set& other) const {

Set result;

for (const auto& elem : elements) {

if (other.contains(elem)) {

result.addElement(elem);

}

}

return result;

}

Set Set::unionSet(const Set& other) const {

Set result;

for (const auto& elem : elements) {

result.addElement(elem);

}

for (const auto& elem : other.elements) {

result.addElement(elem);

}

return result;

}

Set Set::difference(const Set& other) const {

Set result;

for (const auto& elem : elements) {

if (!other.contains(elem)) {

result.addElement(elem);

}

}

return result;

}

Set Set::symmetricDifference(const Set& other) const {

Set result = this->unionSet(other).difference(this->intersection(other)); //

return result;

}

*Set.h*

#ifndef SET\_H

#define SET\_H

#include <iostream>

#include <vector>

#include <string>

class Set {

private:

std::vector<std::string> elements;

public:

Set(); // Конструктор

void addElement(const std::string& element);

bool contains(const std::string& element) const;

void inputElements();

void display() const;

Set intersection(const Set& other) const;

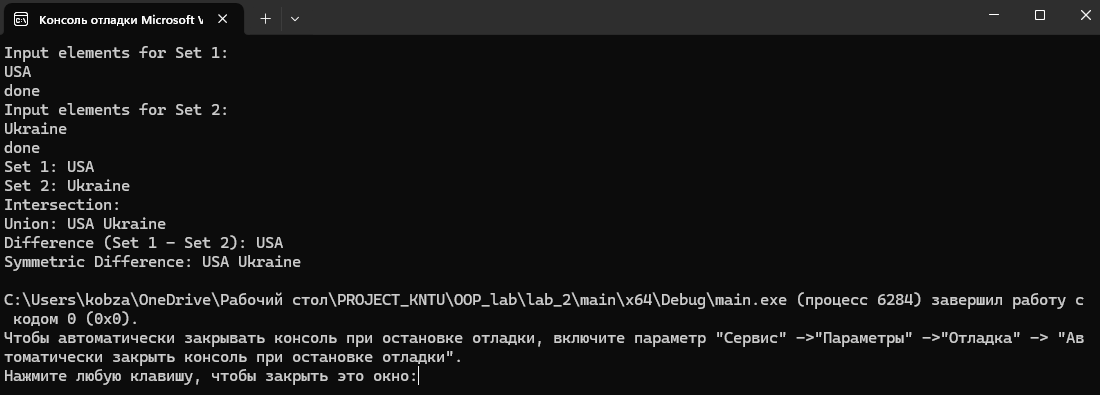
Set unionSet(const Set& other) const;

Set difference(const Set& other) const;

Set symmetricDifference(const Set& other) const;

};

#endif // SET\_H

**Результат:**

**Висновок :** Робота демонструє основи об'єктно-орієнтованого програмування, зокрема використання класів, динамічного управління пам'яттю, а також реалізацію алгоритмів для роботи з множинами.

Запровадження можливості ручного введення елементів множини робить програму більш інтерактивною та зручною для користувача.

**Опис класу Set**

1. **Поле даних**:
   * **elements**: динамічний масив, що містить рядки, які представляють елементи множини.
   * **size**: кількість елементів у множині.
   * **capacity**: максимальна ємність масиву, для динамічного управління пам'яттю.
2. **Конструктори**:
   * **Конструктор за замовчуванням**: ініціалізує множину з нульовим розміром і ємністю.
   * **Конструктор з параметрами**: дозволяє ініціалізувати множину з вказаною ємністю.
   * **Копіюючий конструктор**: копіює значення з іншої множини.
3. **Деструктор**: звільняє пам'ять, виділену для масиву елементів, та виводить повідомлення про знищення об'єкта.
4. **Методи**:
   * **add**: додає елемент до множини, перевіряючи наявність.
   * **contains**: перевіряє, чи входить елемент до множини.
   * **intersection**: обчислює перетин з іншою множиною.
   * **unionSet**: обчислює об'єднання з іншою множиною.
   * **difference**: обчислює різницю між двома множинами.
   * **symmetricDifference**: обчислює симетричну різницю між двома множинами.
   * **display**: виводить елементи множини на екран.

**Основна програма**

У функції main:

1. Запитується у користувача введення елементів для двох множин. Користувач може ввести елементи до тих пір, поки не введе «done».
2. Виводяться елементи обох множин.
3. Обчислюються і виводяться результати для перетину, об'єднання, різниці та симетричної різниці.