

Лабораторна робота №10. Шаблонні функції

Тема: Шаблонні функції.

Мета: Отримати базові знання про шаблонізацію (узагальнення) на основі шаблонних функцій.

ВИМОГИ

1.1 Інформація про розробника:

- Кліщов Б. Р.
- КІТ 102.8а

1.2 Загальне завдання

Створити клас, який не має полів, а усі необхідні дані передаються безпосередньо у функції. Клас має виконувати наступні дії:

- виводити вміст масиву на екран;
- визначати індекс переданого елементу в заданому масиві;
- сортувати елементи масиву;
- визначати значення мінімального елементу масиву. При цьому необхідно продемонструвати роботу програми як з використанням стандартних типів даних, так і типів, створених користувачем.

1.3 Додаткові умови виконання завдання:

- продемонструвати відсутність витоків пам'яті;
- продемонструвати роботу розроблених методів за допомогою модульних тестів;
- не використовувати конструкцію «`using namespace std;`», замість цього слід робити «`using`» кожного необхідного класу: `using std::string`, `using std::cout`.

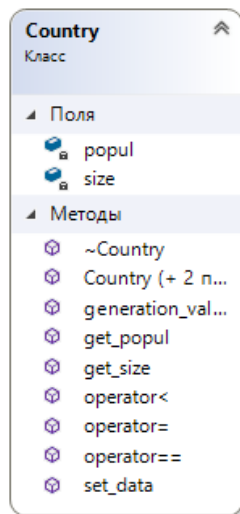
2. ОПИС ПРОГРАМИ

2.1 Функціональне призначення

Програма призначена щоб отримувати та зберігати інформацію щодо різних країн світу.

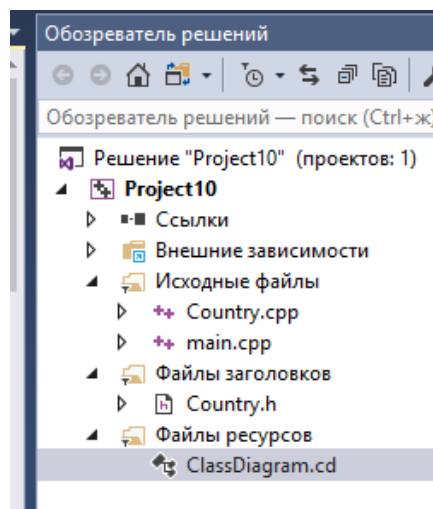
2.2 Опис логічної структури

На рисунку № 1 зображена діаграма класу



Малюнок №1. Діаграма класу

На рисунку № 2 зображена структура програми:



Малюнок №2: Структура програми

2.3 Важливі фрагменти програми

Код програми :

```
namespace Functions {
```

```

template <typename T>
void enterData(T* arr, int size) {
    std::cout << "Enter population: " << std::endl;
    for (int i = 0; i < size; i++) {
        std::cin >> arr[i];
    }
}

template <typename T>
void print(T* arr, int size) {
    for (int i = 0; i < size; i++) {
        std::cout << arr[i] << std::endl;
    }
}

template <typename T>
T minimum(T* arr, int size) {
    T min = arr[0];
    for (int i = 0; i < size; i++) {
        if (arr[i] < min) {
            min = arr[i];
        }
    }
    return min;
}

template <typename T>
void sortArr(T* arr, int size) {
    T temp;
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            if (arr[i] < arr[j]) {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}

template <typename T>
int getByIndex(T* arr, int size, T values) {
    for (int i = 0; i < size; i++) {
        if (arr[i] == values) {
            return i;
        }
    }
    return -1;
}

};

int main() {
    int size = 0;
    std::cout << "Enter size : ";
    std::cin >> size;
    int* arr = new int[size];

    {
        int values;
        Functions::enterData(arr, size);
        int option;
        do {

```

```

        std::cout << "Choose option:" << std::endl << "0 - Exit " <<
std::endl << "1 - Print array" << std::endl << "2 - Sort array" << std::endl << "3 -
Search by index" << std::endl << "4 - Search by min values" << std::endl;
        std::cout << std::endl;
        std::cin >> option;

        switch (option) {
        case 1: {
            system("cls");
            Functions::print(arr, size);
            break;
        }
        case 2: {
            system("cls");
            Functions::sortArr(arr, size);
            Functions::print(arr, size);
            break;
        }
        case 3: {
            std::cout << "Enter values: ";
            std::cin >> values;
            system("cls");
            Functions::getByIndex(arr, size, values);
            std::cout << values << std::endl << std::endl;
            break;
        }
        case 4: {
            int min = Functions::minimum(arr, size);
            system("cls");
            std::cout << min << std::endl << std::endl;
            break;
        }
        default: {
            break;
        }
        }
    } while (option != 0);
    system("cls");
}
delete[]arr;

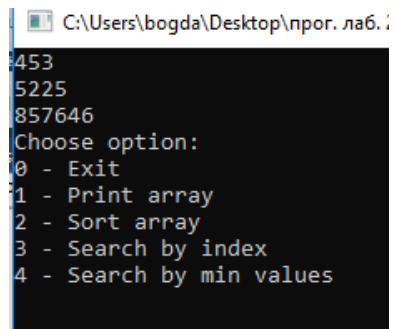
_CrtSetReportMode(_CRT_WARN, _CRTDBG_MODE_FILE);
_CrtSetReportFile(_CRT_WARN, _CRTDBG_FILE_STDERR);
_CrtSetReportMode(_CRT_ERROR, _CRTDBG_MODE_FILE);
_CrtSetReportFile(_CRT_ERROR, _CRTDBG_FILE_STDERR);
_CrtSetReportMode(_CRT_ASSERT, _CRTDBG_MODE_FILE);
_CrtSetReportFile(_CRT_ASSERT, _CRTDBG_FILE_STDERR);
_CrtDumpMemoryLeaks();
system("pause");
return _CrtDumpMemoryLeaks();
}

```

3 ВАРІАНТИ ВИКОРИСТАННЯ

3.1 Результат роботи функцій

На рисунку № 3 зображено результат роботи програми



```
C:\Users\bogda\Desktop\пpor. лаб.1
453
5225
857646
Choose option:
0 - Exit
1 - Print array
2 - Sort array
3 - Search by index
4 - Search by min values
```

Рисунок № 3. Результат роботи програми

Висновок: Отримати базові знання про шаблонізацію (узагальнення) на основі шаблонних функцій.