

ЛАБОРАТОРНА РОБОТА №10. ШАБЛОННІ ФУНКЦІЇ

Тема. Шаблонні функції.

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

1 ВИМОГИ

1.1 Розробник

- Котенко Сергій Миколайович;
- Студент групи КІТ 102.8(а);
- 02-06-2019р..

1.2 Загальне завдання

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

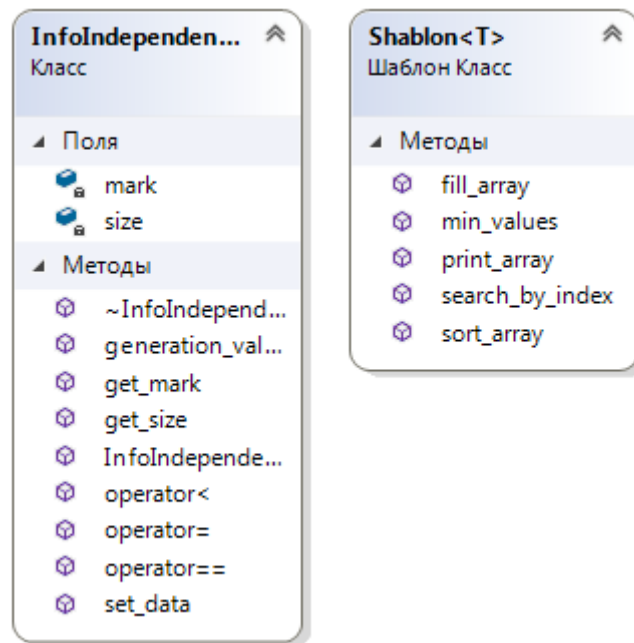
- виводити вміст масиву на екран;
- визначати індекс переданого елементу в заданому масиві;
- сортувати елементи масиву;
- визначати значення мінімального елементу масиву. При цьому необхідно продемонструвати роботу програми як з використанням стандартних типів даних, так і типів, створених користувачем.

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 роботи «using» кожного необхідного класу: using std::string, using std::cout.

2 ОПИС ПРОГРАМИ

2.1 Опис логічної структури



Діаграма класу InfoIndependentsWork:

- ✓ ~InfoIndependentsWork - Деструктор класу;
- ✓ generation_values – Генерація випадкових значень;
- ✓ get_mark , get_size - Отримання даних;
- ✓ InfoIndependentsWork - Конструктор класу;
- ✓ operator< , operator== , operator= - Перевантаження операторів
- ✓ set_data - Встановлення значень .

Діаграма класу Shablon<T> :

- ✓ fill_array – Заповнення масиву;
- ✓ min_values – Знаходження мінімального значення;
- ✓ print_array – Вивід масиву на екран;
- ✓ search_by_index – Пошук числа за індексом;
- ✓ sort_array – Сортювання масиву;

2.2 Фрагменти коду

```
template <class T> <T>
class Shablon {
public:

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

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

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

    void sort_array(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;
                }
            }
        }
    }

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

Рисунок 2.1 – Шаблонні функції

3 ВАРІАНТИ ВИКОРИСТАННЯ

3.2 Ілюстрація роботи програми

```
Enter size : 5
Enter marks:
3
3
5
1
4
Choose option:
0 - Exit
1 - Print array
2 - Sort array
3 - Search by index
4 - Search by min values
```

Рисунок 3.1 – Створення масиву даних та можливі дії над ним

```
1
3
3
4
5
Choose option:
0 - Exit
1 - Print array
2 - Sort array
3 - Search by index
4 - Search by min values
```

Рисунок 3.2 – Сортвання

```
1
Choose option:
0 - Exit
1 - Print array
2 - Sort array
3 - Search by index
4 - Search by min values
```

Рисунок 3.2 – Мінімальне значення

ВИСНОВОК

В інтегрованому середовищі *Visual Studio* розроблена програма мовою C++. Виконання програми дозволяє продемонструвати коректність роботи шаблонних функцій та коректність їх роботи.