Лабораторна робота №21. ООП. Шаблонні функції та класи

1 ВИМОГИ

1.1 Розробник

- Хелемендик Дмитро Олегович;
- студент групи КН-921д;
- 15-трав-2022.

1.2 Загальне завдання

• Зробити шаблоний клас-список (на базі динамічного масиву), що має шаблоноване поле масиву (для будь-якого існуючого типу даних);

• Створити наступні методи:

– вивод вмісту масиву на екран;

– визначити індекс переданого елемента в заданому масиві;

– відсортувати елементи масиву;

– визначити значення мінімального елемента масиву;

– додати елемент до кінця масиву;

– видалити елемент з масиву за індексом.

2 ОПИС ПРОГРАМИ

2.1 Функціональне призначення:

Програма призначена для додавання, видалення елементів різних типів даних, знаходження мінімального значення масиву, відсортування елементів, виводу масиву на екран та отримання індекса елемента за його значенням. Програма працює за допомогою функцій, що задекларовані в *list.hpp, iostream*
Результат зберігається у змінних *int_arr* та *double_arr.*
Демонстрація знайдених результатів передбачає у вікні консолі.

2.2 Опис логічної структури

Створюю шаблонний список List, який містить динамічний масив та його розмір. Список має такі методи: конструктор, деструктор, додавання елементів у масив, видалення з нього, вивод даних на екран, отримання індексу за значенням елемента, сортування та знаходження мінімального елемента.

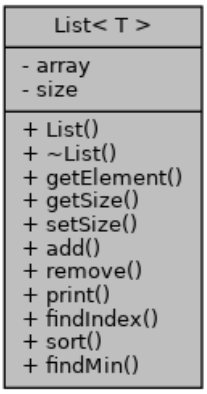

Рисунок 1 — поля класа List

Опис розроблених структур і функцій наводиться на базі результатів роботи системи автодокументування *Doxygen*.

Додавання елемента у список

void add(const T element);

Призначення: додавання елемента у динамічний масив.
Опис роботи: функція виділяє пам'ять для більшого масива, переписує в нього старі елементи(якщо вони є) та додає новий елемент в кінець динамічного масива.
Аргументи:
- *element* — константне посилання на лампочку.

Видалення елемента зі списку

void remove(size_t pos);

Призначення: видалення елемента з динамічного масива.
Опис роботи: функція видаляє елемент зі списку шляхом виділення пам'яті для меншого масива та заповненням в нього всі елементи окрім видаленого.
Аргументи:
- *pos* — позиція для видалення.

Вивод списка

void print() const;

Призначення: виведення всіх елементів на екран.
Опис роботи: функція друкує список на екран.

Отримання індекса за значенням елемента

size_t findIndex(const T element) const;

Призначення: отримання індекса за значенням елемента.
Опис роботи: функція повертає індекс шуканого елемента.
Аргументи:
- *element* — елемент, індекс якого потрібно знайти.

Сортування масиву

void sort() const;

Призначення: сортування всіх елементів динамічного масива.
Опис роботи: функція сортує масив елементів за збільшенням значення.

Знаходження мінімального елемента

T findMin() const;

Призначення: отримання мінімального елемента в динамічному масиві.
Опис роботи: функція повертає мінімальний елемент.

Основна функція

int main()

Призначення: головна функція.

Опис роботи:

- створюю список int та додаю в нього 5 значень функцією add;
- друкую масив за допомогою print;
- видаляю елемент з 0 індексом шляхом виклику функції remove та знову друкую масив функцією print;
- далі знаходю індекс елемента зі значенням 2 за допомогою findIndex;
- потім сортую масив функцією sort;
- також знаходю мінімальний елемент масива функцією findMin;
- тепер виконую всі ці дії знову, але з масивом типу double;
- успішний код повернення з програми (0).

Структура проекту:

```
└── lab21
    ├── main.bin
    └── test.bin
    ├── doc
    │   ├── assets
    │   │   └── list-fields.png
    │   ├── lab21.docx
    │   ├── lab21.md
    │   └── lab21.pdf
    ├── Doxyfile
    ├── Makefile
    ├── README.md
    ├── src
    │   ├── list.hpp
    │   └── main.cpp
    └── test
        └── test.cpp
```

2.3 Важливі фрагменти програми

Додавання елемента у спискок

```
T **new_array = new T *[size + 1];
for (size_t i = 0; i < size; i++) {
        new_array[i] = array[i];
```

```
}
new_array[size] = new T(element);
delete[] array;
array = new_array;
size++;
```

Видалення елемента зі списка

```
if (size == 0)
    return;
T **new_array = new T *[size - 1];
if (pos >= size)
    pos = size - 1;
for (size_t i = 0; i < pos; i++) {
    new_array[i] = array[i];
}
for (size_t i = pos; i < size - 1; i++) {
    new_array[i] = array[i + 1];
}
delete array[pos];
delete[] array;
array = new_array;
size--;
```

3. Варіанти використання

Для демонстрації результатів кожної задачі використовується:

- виконання програми у вікні консолі.

Варіант використання 1: запуск програми у вікні консолі:

- запустити програму у консолі з трьома аргументами;
- подивитись результат програми.

Elements before removing first element:

8 3 2 1 -6

Elements after removing first element:

3 2 1 -6

Getting index of 2

1

Sorting array

-6 1 2 3

Min element: -6

Elements before removing first element:

8.8 3.1 2.3 1.2 -6.9

Elements after removing first element:

3.1 2.3 1.2 -6.9

Getting index of 2.3

1

Sorting array

-6.9 1.2 2.3 3.1

Min element: -6.9

ВИСНОВКИ

При виконанні даної лабораторної роботи було набуто практичного досвіду у роботі з шаблонними функціями та класами.