

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Інститут КНІТ
Кафедра ПЗ

ЗВІТ

До лабораторної роботи № 11

На тему: "Стандартна бібліотека шаблонів. Контейнери та алгоритми"
З дисципліни: "Об'єктно-орієнтоване програмування"

Лектор:

доцент кафедри ПЗ
Коротєєва Т.О.

Виконав:

студент групи ПЗ-16
Коваленко Д.М.

Прийняв:

доцент кафедри ПЗ
Яцишин С.І.

«_____» _____ 2022 р.
 Σ = _____

Тема. Стандартна бібліотека шаблонів. Контейнери та алгоритми.

Мета. Навчитись використовувати контейнери стандартної бібліотеки шаблонів та вбудовані алгоритми.

Лабораторне завдання

Написати програму з використанням бібліотеки STL.

В програмі реалізувати наступні функції:

1. Створити об'єкт-контейнер (1) у відповідності до індивідуального варіанту і заповнити його даними користувацького типу, згідно варіанту.
2. Вивести контейнер.
3. Змінити контейнер, видаливши з нього одні елементи і замінивши інші.
4. Проглянути контейнер, використовуючи для доступу до його елементів ітератори.
5. Створити другий контейнер цього ж класу і заповнити його даними того ж типу, що і перший контейнер.
6. Змінити перший контейнер, видаливши з нього n елементів після заданого і добавивши опісля в нього всі елементи із другого контейнера.
7. Вивести перший і другий контейнери.
8. Відсортувати контейнер по спаданню елементів та вивести результати.
9. Використовуючи необхідний алгоритм, знайти в контейнері елемент, який задовільняє заданій умові.
10. Перемістити елементи, що задовільняють умові в інший, попередньо пустий контейнер (2).

Тип цього контейнера визначається згідно варіанту.

11. Проглянути другий контейнер.
13. Відсортувати перший і другий контейнери по зростанню елементів, вивести результати.
15. Отримати третій контейнер шляхом злиття перших двох.
16. Вивести на екран третій контейнер.
17. Підрахувати, скільки елементів, що задовільняють заданій умові, містить третій контейнер.

Оформити звіт до лабораторної роботи. Звіт має містити варіант завдання, код розробленої програми, результати роботи програми (скріншоти), висновок.

9
priority_queue
stack
int

Код програми

Назва файлу: *main.cpp*

```
#include "mainwindow.h"

#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
    return a.exec();
}
```

Назва файлу: *mainwindow.cpp*

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

#include <queue>
#include <stack>
#include <iostream>

using namespace std;
```

```

priority_queue<int> * q;
priority_queue<int> * q2;
stack<int> * s = new stack<int>();
vector<int> * v = new vector<int>();

MainWindow::MainWindow(QWidget *parent)
: QMainWindow(parent)
, ui(new Ui::MainWindow) {
    ui->setupUi(this);
}

MainWindow::~MainWindow() {
    delete ui;
}

void MainWindow::on_b1_clicked() {
    q = new priority_queue<int>();
    for (QString item : ui->edit1->text().split(" ")) q->push(item.toInt());
}

void MainWindow::on_b2_clicked() {
    priority_queue<int> queue = *q;
    ui->label2->clear();
    for (; !queue.empty(); queue.pop())
        ui->label2->setText(ui->label2->text() + " " + QString::number(queue.top()));
}

void MainWindow::on_b3_clicked() {
    q->pop();
    q->pop();
    q->push(142);
}

void MainWindow::on_b4_clicked() {
    priority_queue<int> queue = *q;
    ui->label4->clear();
    vector<int> v;
    while (!queue.empty()) {
        v.push_back(queue.top());
        queue.pop();
    }
    for (vector<int>::iterator i = v.begin(); i != v.end(); i++)
        ui->label4->setText(ui->label4->text() + " " + QString::number(*i));
}

void MainWindow::on_b5_clicked() {
    q2 = new priority_queue<int>();
    for (QString item : ui->edit5->text().split(" ")) q2->push(item.toInt());
}

void MainWindow::on_b6_clicked() {
    priority_queue<int> queue = *q;
    priority_queue<int> queue2 = *q2;
    priority_queue<int> * nq = new priority_queue<int>();
    for (int i = 0; i<ui->edit62->text().toInt(); i++) {
        nq->push(queue.top());
        queue.pop();
    }
    for (int n = 0; n<ui->edit6->text().toInt(); n++) {

```

```

        queue.pop();
    }
    while (!queue.empty()) {
        nq->push(queue.top());
        queue.pop();
    }
    while (!queue2.empty()) {
        nq->push(queue2.top());
        queue2.pop();
    }
    q = new priority_queue<int>(*nq);
}

void MainWindow::on_b7_clicked() {
    priority_queue<int> queue = *q;
    priority_queue<int> queue2 = *q2;
    ui->label7->clear();
    ui->label7->setText("(1) ");
    for (; !queue.empty(); queue.pop())
        ui->label7->setText(ui->label7->text() + " " + QString::number(queue.top()));
    ;
    ui->label7->setText(ui->label7->text() + " (2) ");
    for (; !queue2.empty(); queue2.pop())
        ui->label7->setText(ui->label7->text() + " " + QString::number(queue2.top()));
    );
}

void MainWindow::on_b8_clicked() {
    priority_queue<int> queue = *q;

    ui->label8->clear();
    for (; !queue.empty(); queue.pop())
        ui->label8->setText(ui->label8->text() + " " + QString::number(queue.top()));
    ;
}

void MainWindow::on_b9_clicked() {
    priority_queue<int> queue = *q;
    ui->label9->clear();
    vector<int> v;
    while (!queue.empty()) {
        if (queue.top() > ui->edit9->text().toInt()) v.push_back(queue.top());
        queue.pop();
    }
    for (int i : v)
        ui->label9->setText(ui->label9->text() + " " + QString::number(i));
}

void MainWindow::on_b10_clicked() {
    priority_queue<int> queue = *q;
    while (!queue.empty()) {
        if (queue.top() > ui->edit10->text().toInt()) s->push(queue.top());
        queue.pop();
    }
}

void MainWindow::on_b11_clicked() {
    stack<int> stack = *s;

    ui->label11->clear();
    for (; !stack.empty(); stack.pop())

```

```

        ui->label11->setText(ui->label11->text() + " " + QString::number(stack.top())
    );
}

void MainWindow::on_b12_clicked() {
    priority_queue<int> queue = *q;

    vector<int> v;
    while (!queue.empty()) {
        v.push_back(queue.top());
        queue.pop();
    }
    std::priority_queue<int, std::vector<int>, std::greater<int>> sq(v.begin(),
v.end());
    ui->label13->clear();
    for (; !sq.empty(); sq.pop())
        ui->label13->setText(ui->label13->text() + " " + QString::number(sq.top()));
}

void MainWindow::on_b13_clicked() {
    priority_queue<int> queue = *q;
    stack<int> stack = *s;

    while (!queue.empty()) {
        v->push_back(queue.top());
        queue.pop();
    }
    while (!stack.empty()) {
        v->push_back(stack.top());
        stack.pop();
    }
}

void MainWindow::on_b14_clicked() {
    ui->label16->clear();
    for (int i : *v)
        ui->label16->setText(ui->label16->text() + " " + QString::number(i));
}

void MainWindow::on_b15_clicked() {
    int n = 0;
    for (int i : *v) {
        if (i > ui->edit17->text().toInt()) n++;
    }
    ui->label17->setText(QString::number(n));
}

```

Робота програми

MainWindow (on toolbox) x

1	11 22 33 44 55
2	55 44 33 22 11
3	
4	142 33 22 11
5	1 2 3
6	2 2
7	(1) 142 33 3 2 1 (2) 3 2 1
8	142 33 3 2 1
9	2 142 33 3
10	2
11	3 33 142
13	1 2 3 33 142
15	
16	142 33 3 2 1 3 33 142
17	2 6

Висновок

Під час виконання лабораторної роботи я навчивсь використовувати контейнери стандартної бібліотеки шаблонів та вбудовані алгоритми