

ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

Старший преподаватель

должность, уч. степень, звание

подпись, дата

Е.О. Шумова

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5

«Использование контейнерных классов в среде Qt Creator»

по курсу: ОБЪЕКТНО ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4134к

подпись, дата

Костяков Н.А.

инициалы, фамилия

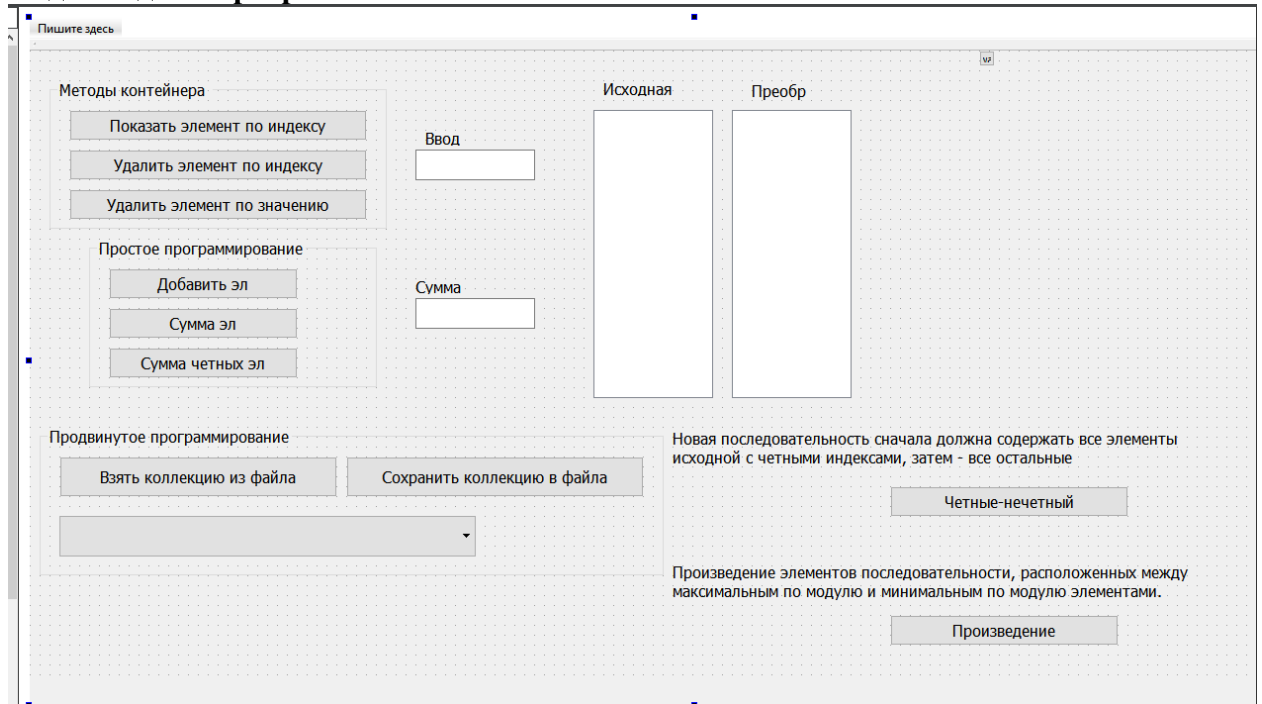
Санкт-Петербург 2022

Цель работы: научиться использовать контейнерные классы в среде Qt Creator.

Задание 4. Новая последовательность сначала должна содержать все элементы исходной с четными индексами, затем - все остальные (с сохранением исходного относительного порядка).

Произведение элементов последовательности, расположенных между максимальным по модулю и минимальным по модулю элементами

Вид исходной программы



Листинг программы

Mainwindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <vector>

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = 0);
    ~MainWindow();
    std::vector<int> vec = {32, 1, 45, 34, 23};
    std::string paths[2] = {"C:\\qtprojects\\laba5\\file.txt",
"C:\\qtprojects\\laba5\\file2.txt"};

private slots:
    void on_pushButton_clicked();

    void on_pushButton_2_clicked();
}
```

```

void on_pushButton_3_clicked();

void on_pushButton_4_clicked();

void on_pushButton_5_clicked();

void refresh();

void on_pushButton_6_clicked();

void on_pushButton_7_clicked();

void load(std::string path);
void save(std::string path);

void on_pushButton_8_clicked();

void on_pushButton_9_clicked();

void on_pushButton_11_clicked();

void on_pushButton_10_clicked();

private:
    Ui::MainWindow *ui;
};

#endif // MAINWINDOW_H

```

Main.cpp

```

#include "mainwindow.h"
#include <QApplication>

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

    return a.exec();
}

```

Mainwindow.cpp

```

#include "mainwindow.h"
#include "ui_mainwindow.h"
#include "string"
#include "fstream"

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    setWindowTitle("KostyakovNikita 4134k");
    refresh();

    for(auto i : paths) ui->comboBox->addItem(QString::fromStdString(i));
    ui->listWidget_2->clear();
    for(auto i: vec) ui->listWidget_2->addItem(QString::number(i));
}

MainWindow::~MainWindow()

```

```

{
    delete ui;
}

void MainWindow::refresh() {
    ui->listWidget->clear();
    for(auto i: vec) ui->listWidget->addItem(QString::number(i));
}

void MainWindow::load(std::string path) {
    std::ifstream input;
    //C:\\qtprojects\\laba5\\file.txt
    input.open(path);
    vec.clear();
    while (!input.eof())
    {
        std::string s;
        std::getline(input, s);
        if(s!="") {

            int buf = std::stoi(s);
            vec.push_back(buf);

        }
    }

    input.close();
    refresh();
    ui->listWidget_2->clear();
    for(auto i: vec) ui->listWidget_2->addItem(QString::number(i));
}

void MainWindow::save(std::string path) {
    std::ofstream file;
    file.open(path);
    for (auto i : vec) {
        file<<i<<"\n";
    }
    file.close();
}

void MainWindow::on_pushButton_clicked()
{
    if (ui->lineEdit->text()=="") return;
    ui->listWidget->clear();
    int index = std::stoi((ui->lineEdit->text()).toStdString());
    ui->lineEdit->setText("");

    ui->label_7->setText(QString::number(vec[index]));

    refresh();
}

void MainWindow::on_pushButton_2_clicked()
{
    if (ui->lineEdit->text()=="") return;
    ui->listWidget->clear();
    int index = std::stoi((ui->lineEdit->text()).toStdString());
    ui->lineEdit->setText("");

    auto iterator = std::begin(vec);
    for (int i = 0; i<index ;i++) iterator = std::next(iterator);
    vec.erase(iterator);
}

```

```

        refresh();
    }

void MainWindow::on_pushButton_3_clicked()
{
    if (ui->lineEdit->text()=="") return;
    ui->listWidget->clear();
    int index = std::stoi(ui->lineEdit->text()).toString();
    ui->lineEdit->setText("");
    auto iterator = 0;

    while (index!=vec[iterator]){
        iterator ++;
    }
    vec.erase(vec.begin()+iterator);

    refresh();
}

void MainWindow::on_pushButton_4_clicked()
{
    if (ui->lineEdit->text()=="") return;
    ui->listWidget->clear();
    int index = std::stoi(ui->lineEdit->text()).toString();
    ui->lineEdit->setText("");

    vec.push_back(index);

    refresh();
}

void MainWindow::on_pushButton_5_clicked()
{
    int sum=0;
    for(auto i : vec)
    {
        sum+=i;
    }
    ui->lineEdit_3->setText(QString::number(sum));
    refresh();
}

void MainWindow::on_pushButton_6_clicked()
{
    int sum=0;

    for(auto i : vec) if(i%2==0) sum+=i;

    ui->lineEdit_3->setText(QString::number(sum));
    refresh();
}

void MainWindow::on_pushButton_7_clicked()
{
    load((ui->comboBox->currentText()).toString());
}

void MainWindow::on_pushButton_8_clicked()
{
    save((ui->comboBox->currentText()).toString());
}

```

```

void MainWindow::on_pushButton_9_clicked()
{
    ui->listWidget->clear();
    for (int i = 0; i<vec.size(); i+=2){
        ui->listWidget->addItem(QString::number(vec[i]));
    }
    for (int i = 1; i<vec.size(); i+=2){
        ui->listWidget->addItem(QString::number(vec[i]));
    }
}

void MainWindow::on_pushButton_11_clicked()
{
}

void MainWindow::on_pushButton_10_clicked()
{
    int min = 10000;
    int max = -10000;
    int i_max=0, i_min=0, it=0;

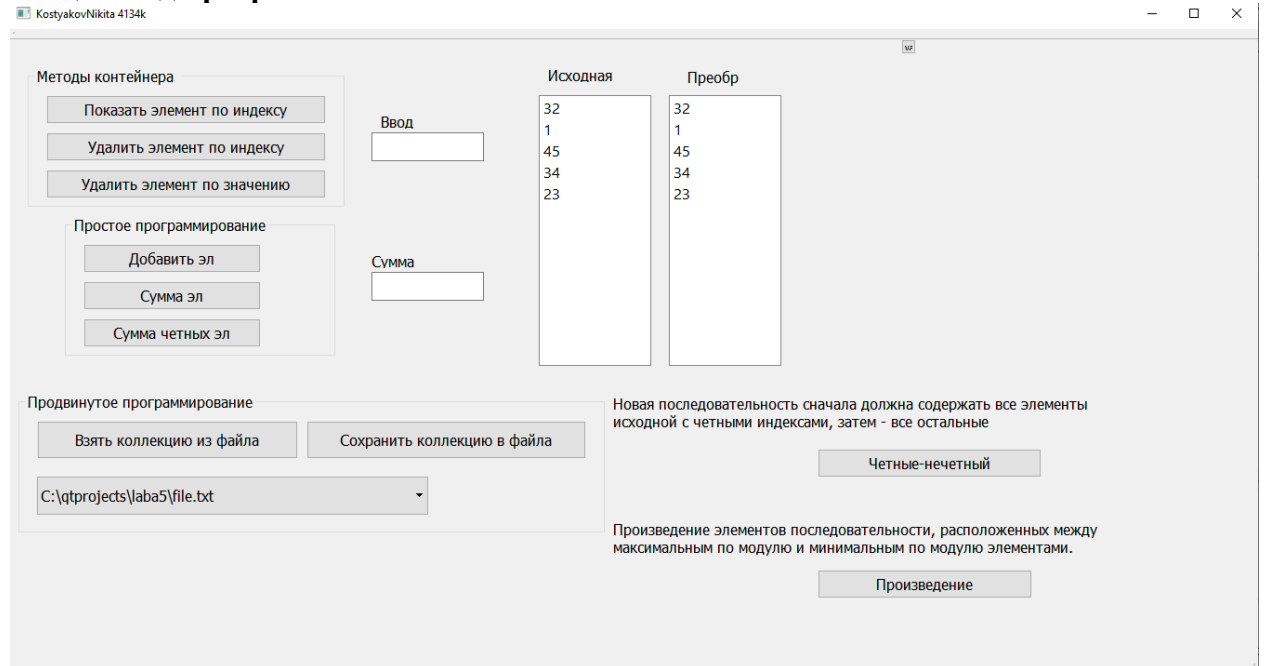
    for (auto i: vec){
        if(i>i_max) {
            max = i;
            i_max = it;
        }
        if(i<i_min) {
            min = i;
            i_min = it;
        }
        it++;
    }
    int mult = 1;
    if (i_min>i_max){
        int temp = i_min;
        i_min = i_max;
        i_max = temp;
    }

    for(int q = i_min+1; q<i_max; q++) mult*=vec[q];
    ui->listWidget->clear();
    ui->listWidget->addItem(QString::number(mult));
}

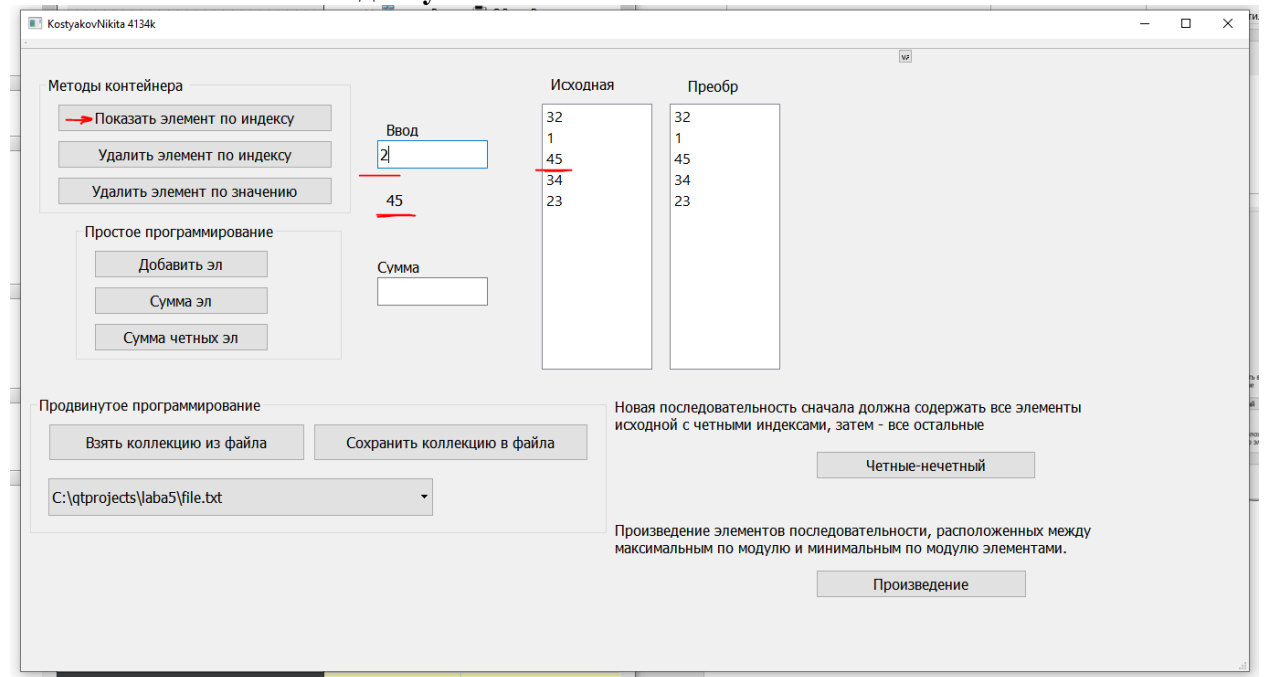
```

Результат работы

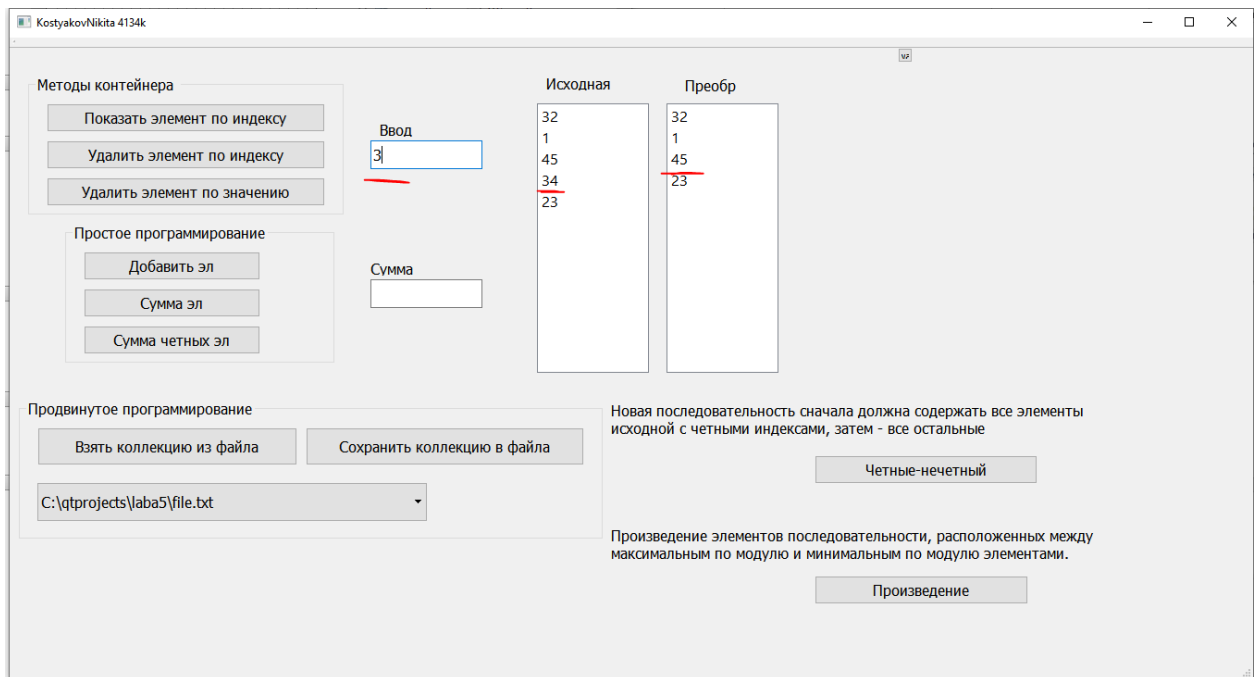
Общий вид программы



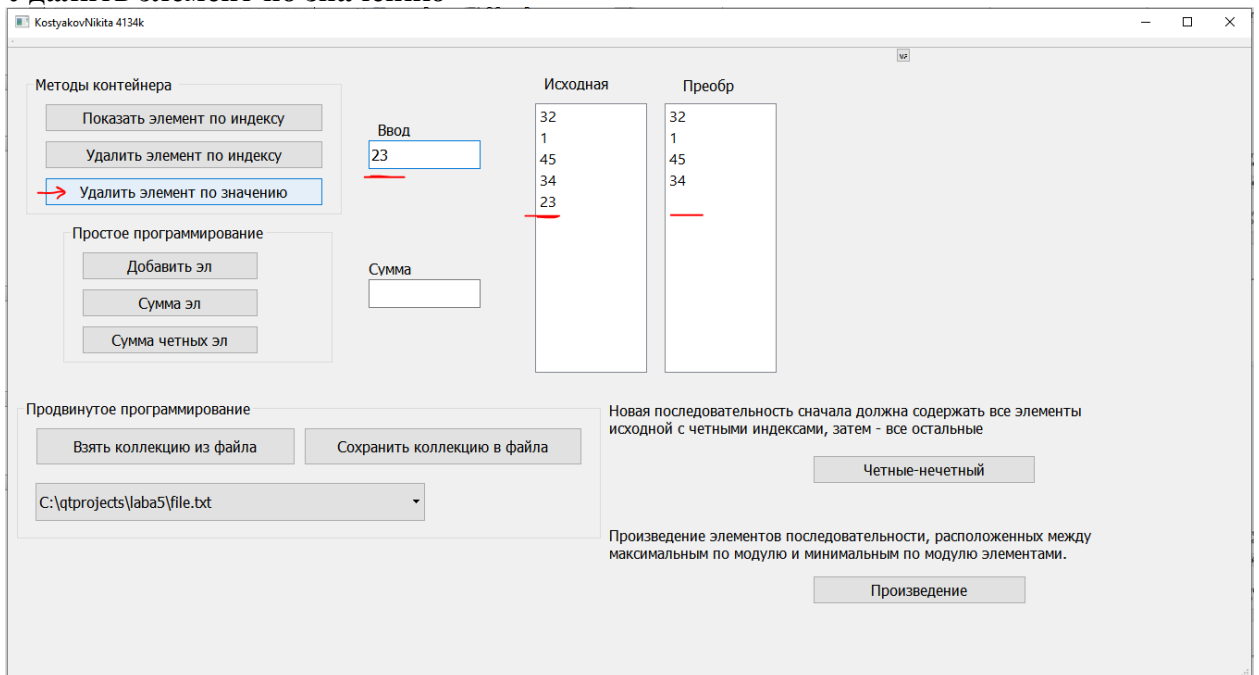
Показать элемент по индексу



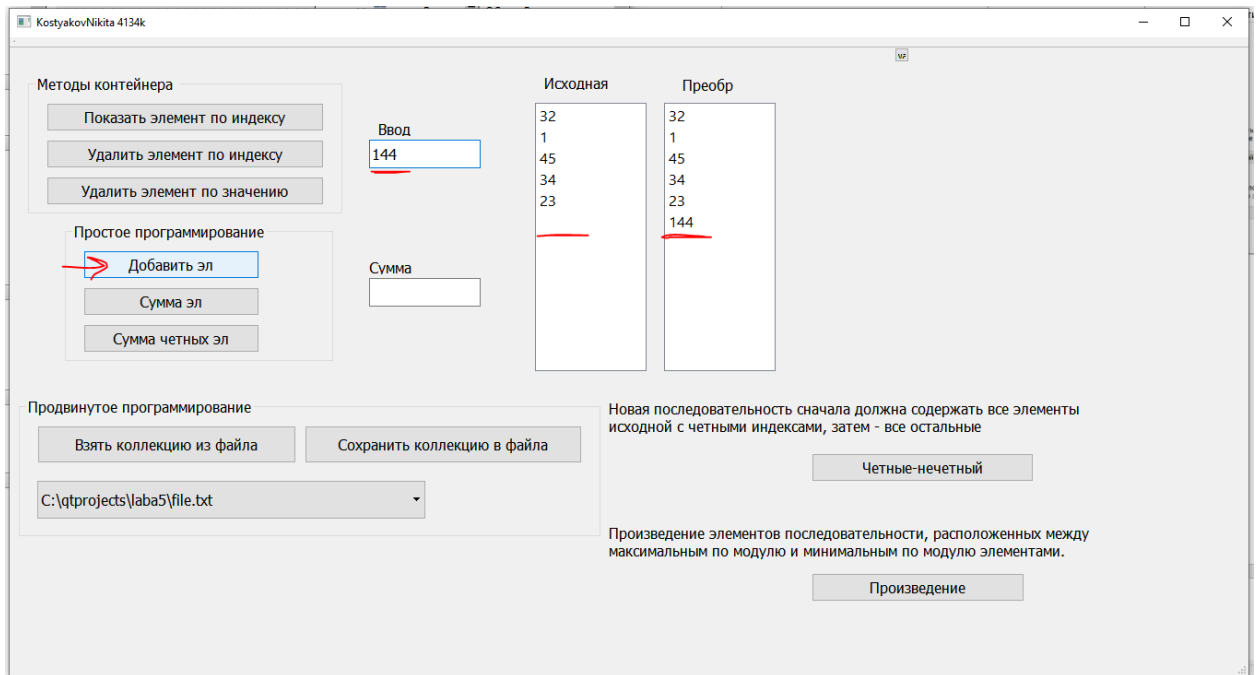
Удалить элемент по индексу



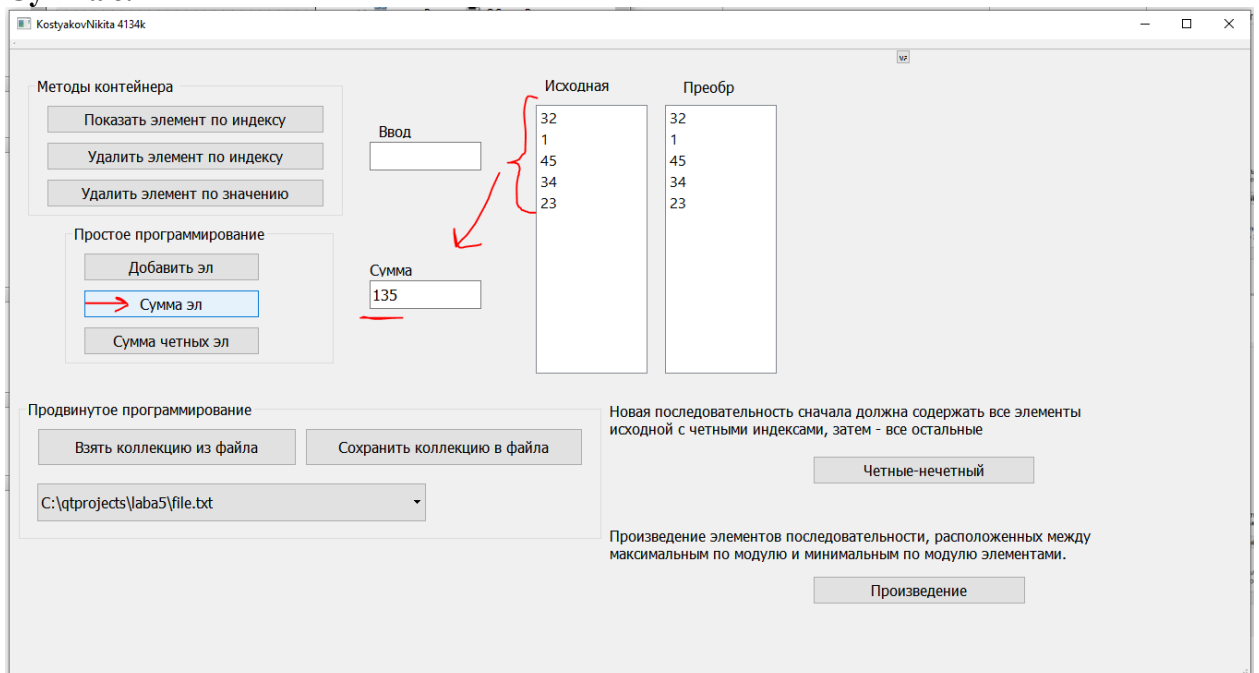
Удалить элемент по значению



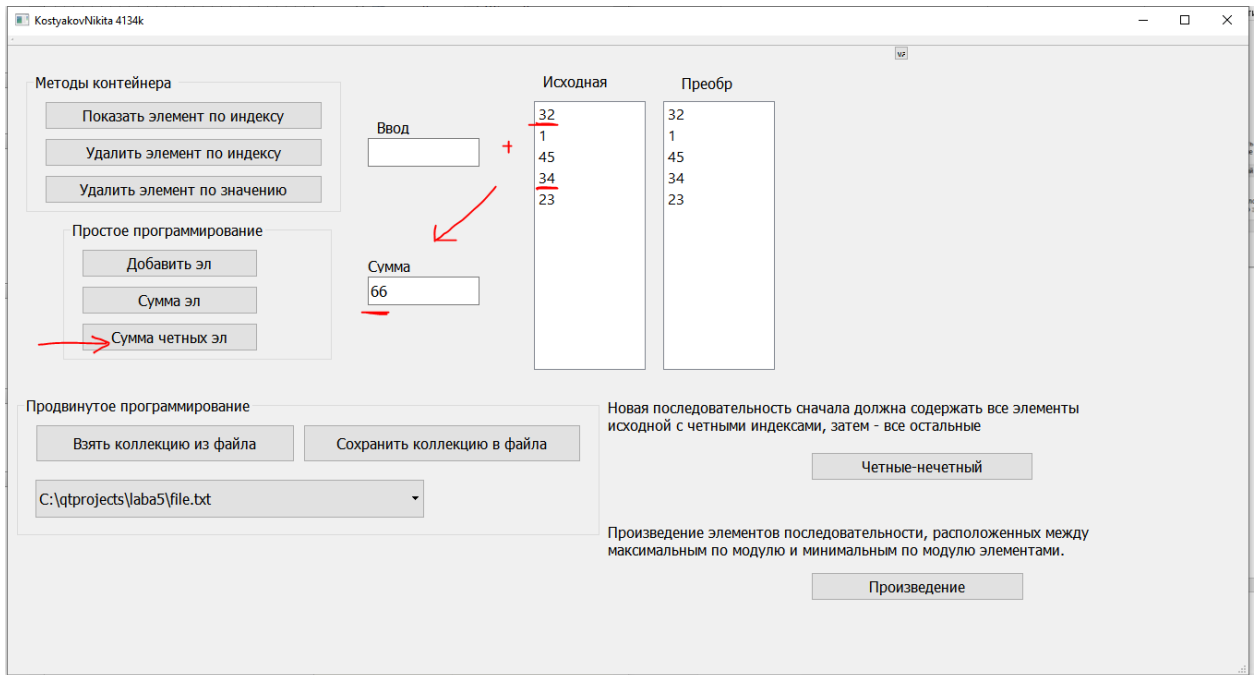
Добавить эл



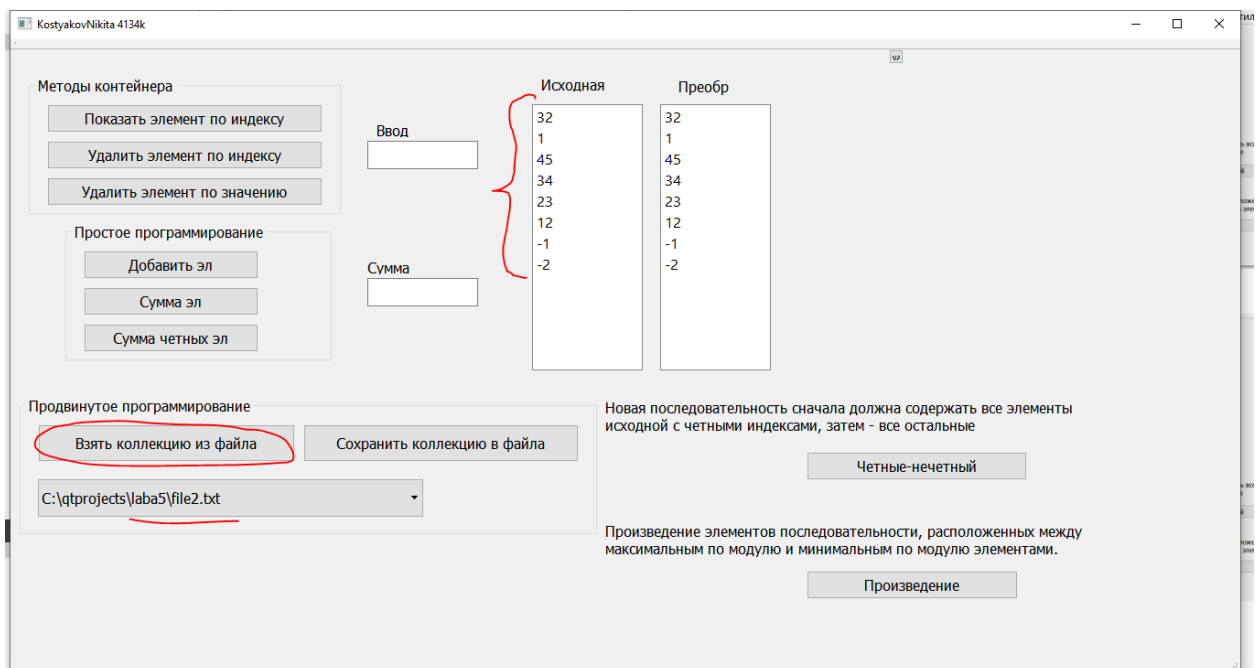
Сумма эл



Сумма четных эл



Взять коллекцию из файла



Четный-нечетный

KostyakovNikita 4134k

Методы контейнера

Показать элемент по индексу

Удалить элемент по индексу

Удалить элемент по значению

Простое программирование

Добавить эл

Сумма эл

Сумма четных эл

Ввод

Сумма

Исходная

Преобр

Четные-нечетный

Произведение

Новая последовательность сначала должна содержать все элементы исходной с четными индексами, затем - все остальные

Произведение элементов последовательности, расположенных между максимальным по модулю и минимальным по модулю элементами.

Исходная	Преобр
32	32
1	45
45	23
34	-1
23	1
12	34
-1	12
-2	-2

Произведение

KostyakovNikita 4134k

Методы контейнера

Показать элемент по индексу

Удалить элемент по индексу

Удалить элемент по значению

Простое программирование

Добавить эл

Сумма эл

Сумма четных эл

Ввод

Сумма

Исходная

Преобр

Четные-нечетный

Произведение

Новая последовательность сначала должна содержать все элементы исходной с четными индексами, затем - все остальные

Произведение элементов последовательности, расположенных между максимальным по модулю и минимальным по модулю элементами.

Исходная	Преобр
32	-9384
1	
45	
34	
23	
12	
-1	
-2	

$-(34 \cdot 23 \cdot 12 \cdot 1)$

max

min

Выводы

Я научился использовать контейнерные классы в среде qtcreator