

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5

«Использование контейнерных классов в среде Qt Creator»

по курсу: Объектно-ориентированное программирование

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

подпись, дата

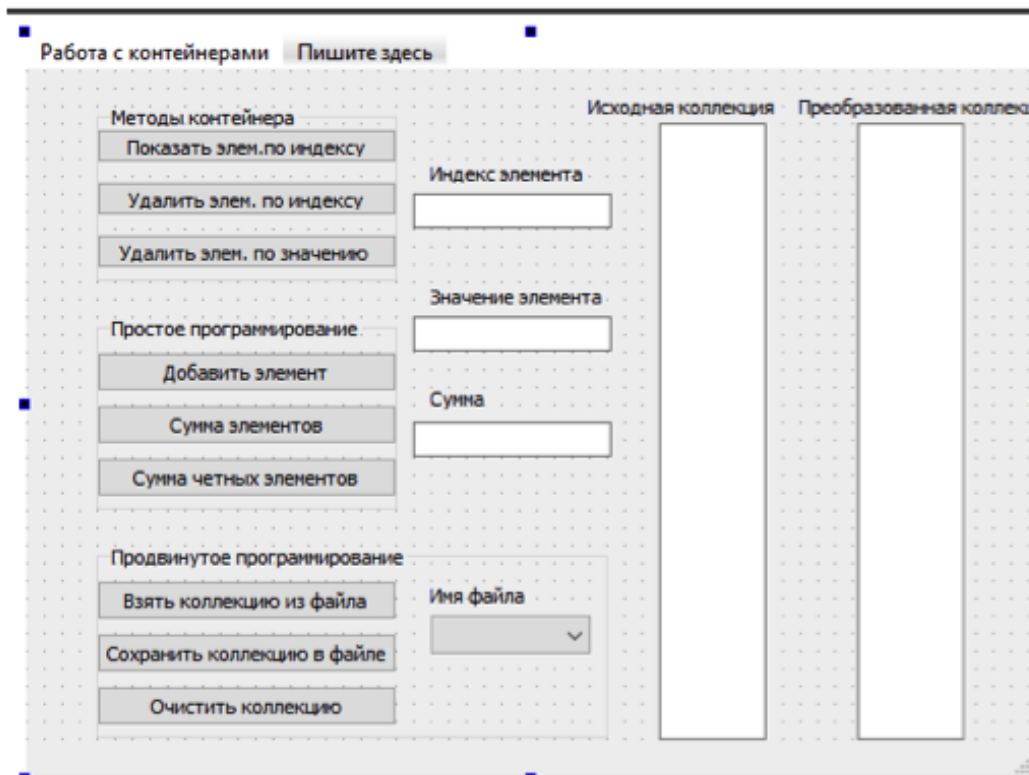
инициалы, фамилия

1. Цель работы

Научиться использовать контейнерные классы в среде Qt Creator.

2. Задание

- 1) Разработайте форму, подобную приведенной на рисунке, добавив на форму кнопки для выполнения Ваших индивидуальных заданий (согласно варианта). **Шрифт на форме не должен быть менее 12.** В качестве заголовка формы пропишите свою фамилию, имя и номер группы.



- 2) Добавьте в проект свой класс для работы с коллекцией.
- 3) Для представления исходной и результирующей коллекции используйте шаблон QList.
- 4) Данные исходной коллекции загрузите из файла.
- 5) Реализуйте все функции, указанные на форме
- 6) Выполните задание 1 согласно варианта
- 7) Выполните задание 2 согласно варианта
- 8) Если для выполнения задания потребуются еще дополнительные виджеты, то студент может их добавить на форму.

Задание 1. Дана последовательность действительных чисел. Необходимо сформировать новую последовательность по некоторому правилу. Для

2. В новой последовательности сначала должны идти все отрицательные элементы исходной, затем все нулевые элементы, затем все положительные (с сохранением исходного относительного порядка).

Задание 2. Дана последовательность действительных чисел. Необходимо вычислить:

2. Сумму элементов последовательности, расположенных между первым и вторым отрицательными элементами.

3. Форма

The screenshot shows a web application interface with a dark background and a grid of buttons. The interface is organized into several sections:

- Методы контейнера (Container Methods):** Includes buttons for "Показать элем. по индексу" (Show element by index), "Удалить элем. по индексу" (Delete element by index), and "Удалить элем. по значению" (Delete element by value).
- Простое программирование (Simple Programming):** Includes buttons for "Добавить элемент" (Add element), "Сумма элементов" (Sum of elements), and "Сумма четных элементов" (Sum of even elements).
- Продвинутое программирование (Advanced Programming):** Includes buttons for "Взять коллекцию из файла" (Load collection from file), "Сохранить коллекцию в файл" (Save collection to file), and "Очистить коллекцию" (Clear collection).
- Задания (Tasks):** Includes buttons for "Задание 1" (Task 1) and "Задание 2" (Task 2).

On the right side, there are two large vertical panels labeled "Исходная коллекция" (Original collection) and "Преобразованная коллекция" (Transformed collection). Above these panels are input fields for "Индекс элемента" (Element index), "Значение элемента" (Element value), and "Сумма" (Sum). A file selection dropdown labeled "Имя файла" (File name) is also present, showing "test.txt".

Были использованы следующие виджеты:

6 – label: label, label_2, label_3, label_4, label_5, label_6 которые относятся к наименованию, пояснительным надписям для полей,

11 – pushButton: pushButton, pushButton_2, pushButton_3, pushButton_4, pushButton_5, pushButton_6, pushButton_7, pushButton_8, pushButton_9, pushButton_10, pushButton_11 которые отвечают за три метода контейнеров – показать и удалить элемент по индексу, либо удалить элемент по значению, три pushButton для простого программирования – добавить и суммировать элементы, и суммировать только чётные, также три pushButton для продвинутого программирования – работа с файлами – взять, сохранить и очистить коллекцию, и 2 pushButton для 2 заданий соответственно.

1 – comboBox: для выбора файла

3 – lineEdit: index_input, value_input, summ: ввод индекса, значения, вывод суммы

2 – listWidget: listWidget, listWidget _2, для вывода исходных и преобразованных коллекций

4.1 Текст программы (*.h)

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H
```

```
#include <QMainWindow>
#include <QFile>
#include <QDir>
#include <QTextStream>
```

```
QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE
```

```
class MainWindow : public QMainWindow
{
    Q_OBJECT
    QList<QString> list;
    QFile fin;
```

```
public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();
```

```
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 on_pushButton_6_clicked();
```

```
void on_pushButton_9_clicked();
```

```
void on_pushButton_7_clicked();
```

```
void on_pushButton_8_clicked();
```

```
void on_pushButton_10_clicked();
```

```
void on_pushButton_11_clicked();
```

```
private:
```

```
    Ui::MainWindow *ui;
```

```
};
```

```
#endif // MAINWINDOW_H
```

Текст программы (*.cpp)

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

```
#include "./ui_mainwindow.h"
```

```
#include <QFile>
```

```
#include <QDir>
```

```
#include <QTextStream>
```

```
MainWindow::MainWindow(QWidget *parent)
```

```
    : QMainWindow(parent)
```

```
    , ui(new Ui::MainWindow)
```

```
{
```

```
    ui->setupUi(this);
```

```
        this->setWindowTitle("Захаров 4133К");  
    }  
}
```

```
MainWindow::~MainWindow()
```

```
{  
    delete ui;  
}
```

```
//элемент по индексу
```

```
void MainWindow::on_pushButton_clicked()
```

```
{  
    if (list.count() > ui->index_input->text().toInt())  
    {  
        ui->value_input->setText(list.at(ui->index_input->text().toInt()));  
        ui->index_input->clear();  
    }  
    else  
    {  
        ui->value_input->setText("Индекс вне списка");  
        ui->index_input->clear();  
    }  
    ui->summ->clear();  
}
```

```
//удалить по индексу
```

```
void MainWindow::on_pushButton_2_clicked()
```

```
{  
    if (list.count() > ui->index_input->text().toInt())  
    {  
        ui->value_input->setText(list.at(ui->index_input->text().toInt()));  
        list.removeAt(ui->index_input->text().toInt());  
        ui->listWidget_2->clear();  
        ui->listWidget_2->addItems(list);  
    }  
}
```

```

        ui->value_input->clear();
        ui->index_input->clear();
    }
    else
    {
        ui->value_input->setText("Индекс вне списка");
        ui->index_input->clear();
    }
    ui->summ->clear();
}

```

//удалить по значению

```

void MainWindow::on_pushButton_3_clicked()
{
    if (list.contains(ui->value_input->text()))
    {
        ui->index_input->setText(QString::number(list.indexOf(ui->value_input->text())));
        list.removeAt(ui->index_input->text().toInt());
        ui->listWidget_2->clear();
        ui->listWidget_2->addItems(list);
        ui->value_input->clear();
        ui->index_input->clear();
    }
    else
    {
        ui->index_input->setText("--");
        ui->value_input->clear();
    }
    ui->summ->clear();
}

```

//добавить элемент

```

void MainWindow::on_pushButton_4_clicked()

```

```

{
    if (ui->value_input->text() != "")
        list += ui->value_input->text();
    ui->listWidget_2->clear();
    ui->listWidget_2->addItems(list);
    ui->value_input->clear();
    ui->index_input->clear();
    ui->summ->clear();
}

```

//сумма

```

void MainWindow::on_pushButton_5_clicked()
{
    int sum_ = 0;
    foreach(const QString &str, list)
        sum_ += str.toInt();
    ui->summ->setText(QString::number(sum_));
    ui->value_input->clear();
    ui->index_input->clear();
}

```

//сумма чётных

```

void MainWindow::on_pushButton_6_clicked()
{
    int sum_ = 0;
    foreach(const QString& str, list)
        if (!(str.toInt() % 2))
            sum_ += str.toInt();
    ui->summ->setText(QString::number(sum_));
    ui->value_input->clear();
    ui->index_input->clear();
}

```



```
//очистить коллекцию
```

```
void MainWindow::on_pushButton_9_clicked()
```

```
{  
    list.clear();  
    ui->listWidget_2->clear();  
    ui->value_input->clear();  
    ui->index_input->clear();  
    ui->summ->clear();  
}
```

```
//взять из файла
```

```
void MainWindow::on_pushButton_7_clicked()
```

```
{  
    switch(ui->comboBox->currentIndex())  
    {  
        case 0:  
        {  
            ui->listWidget->clear();  
            ui->listWidget_2->clear();  
            QFile fin("/Users/andrey/Documents/QTproj/test.txt");  
            fin.open(QFile::ReadOnly | QFile::Text);  
            QString buffer = fin.readLine();  
            list = buffer.split(" ");  
            ui->listWidget->addItems(list);  
            fin.close();  
            break;  
        }  
        case 1:  
        {  
            ui->listWidget->clear();  
            ui->listWidget_2->clear();  
            QFile fin("/Users/andrey/Documents/QTproj/input.txt");  
            fin.open(QFile::ReadOnly | QFile::Text);
```

```

        QString buffer = fin.readLine();

        list = buffer.split(" ");

        ui->listWidget->addItems(list);

        fin.close();

        break;
    }
    case 2:
    {
        ui->listWidget->clear();

        ui->listWidget_2->clear();

        QFile fin("/Users/andrey/Documents/QTproj/output.txt");

        fin.open(QFile::ReadOnly | QFile::Text);

        QString buffer = fin.readLine();

        list = buffer.split(" ");

        ui->listWidget->addItems(list);

        fin.close();

        break;
    }
}
}
}

```

//сохранить в файл

```

void MainWindow::on_pushButton_8_clicked()
{
    switch(ui->comboBox->currentIndex())
    {
        case 0:
        {
            QFile fin("/Users/andrey/Documents/QTproj/test.txt");

            fin.open(QFile::WriteOnly | QFile::Text);

            QTextStream stream(&fin);

            QList<QString>::iterator it = list.begin();

            for (; it != list.end() - 1; ++it)

```

```

        stream << *it << " ";

    stream << *it;

    fin.flush();

    fin.close();

    break;
}

case 1:
{
    QFile fin("/Users/andrey/Documents/QTproj/input.txt");

    fin.open(QFile::WriteOnly | QFile::Text);

    QTextStream stream(&fin);

    QList<QString>::iterator it = list.begin();

    for (; it != list.end() - 1; ++it)

        stream << *it << " ";

    stream << *it;

    fin.flush();

    fin.close();

    break;
}

case 2:
{
    QFile fin("/Users/andrey/Documents/QTproj/output.txt");

    fin.open(QFile::WriteOnly | QFile::Text);

    QTextStream stream(&fin);

    QList<QString>::iterator it = list.begin();

    for (; it != list.end() - 1; ++it)

        stream << *it << " ";

    stream << *it;

    fin.flush();

    fin.close();

    break;
}
}

```

```
}
```

```
//задание 1 - неотрицательные в том же порядке
```

```
void MainWindow::on_pushButton_10_clicked()
```

```
{
```

```
    QString str;
```

```
    int i = 0;
```

```
    int j = 0;
```

```
    foreach(str, list)
```

```
    {
```

```
        if (str.toInt() >= 0)
```

```
        {
```

```
            list.swapItemsAt(i++, j);
```

```
        j++;
```

```
        }
```

```
    else
```

```
        list.takeAt(j);
```

```
    }
```

```
    ui->listWidget_2->clear();
```

```
    ui->listWidget_2->addItems(list);
```

```
    ui->value_input->clear();
```

```
    ui->index_input->clear();
```

```
    ui->summ->clear();
```

```
}
```

```
// сумма между 1 и 2 отрицательным
```

```
void MainWindow::on_pushButton_11_clicked()
```

```
{
```

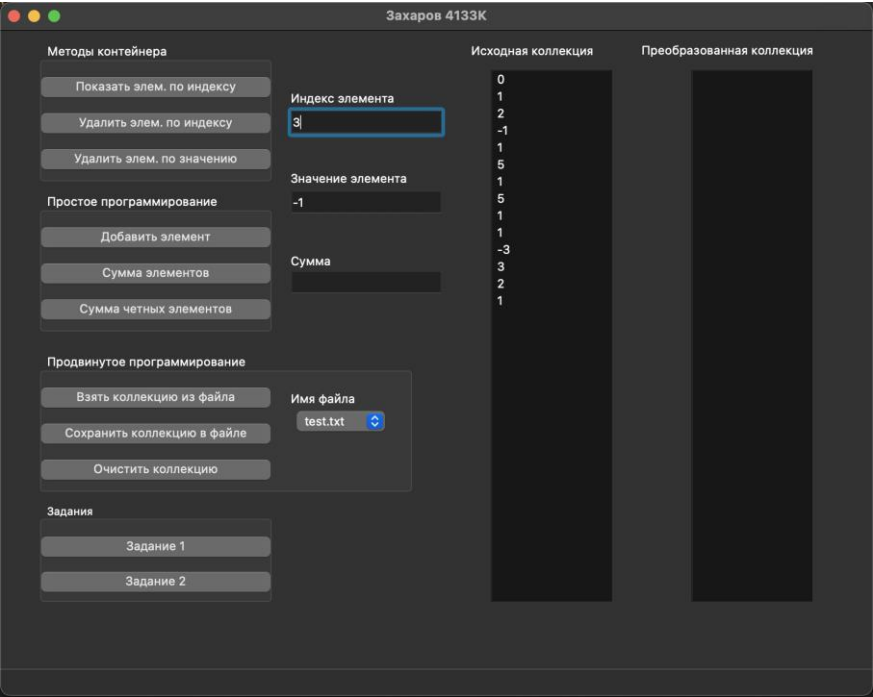
```
int sum_ = 0;
bool flag_1 = false, flag_2 = false;
foreach(QString str, list)
{
    if (str.toInt() < 0)
    {
        if (flag_1)
            flag_2 = true;
        else
            flag_1 = true;
    }
    else
        if ((flag_1) && !(flag_2))
            sum_ += str.toInt();
}
ui->summ->clear();
if ((flag_1) && (flag_2))
    ui->summ->setText(QString::number(sum_));
ui->value_input->clear();
ui->index_input->clear();

}
```

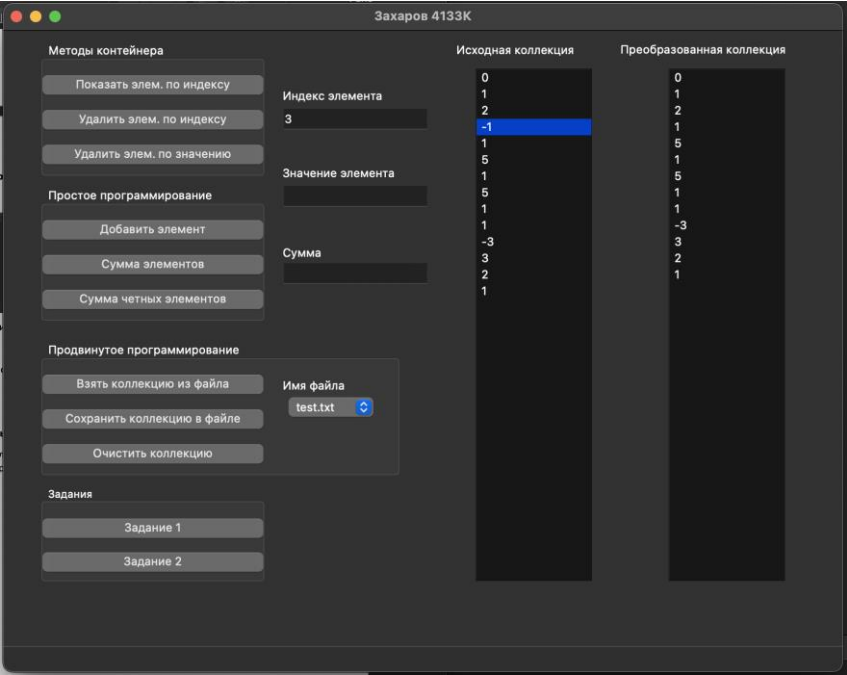
4. Пример выполнения программы

Методы контейнера:

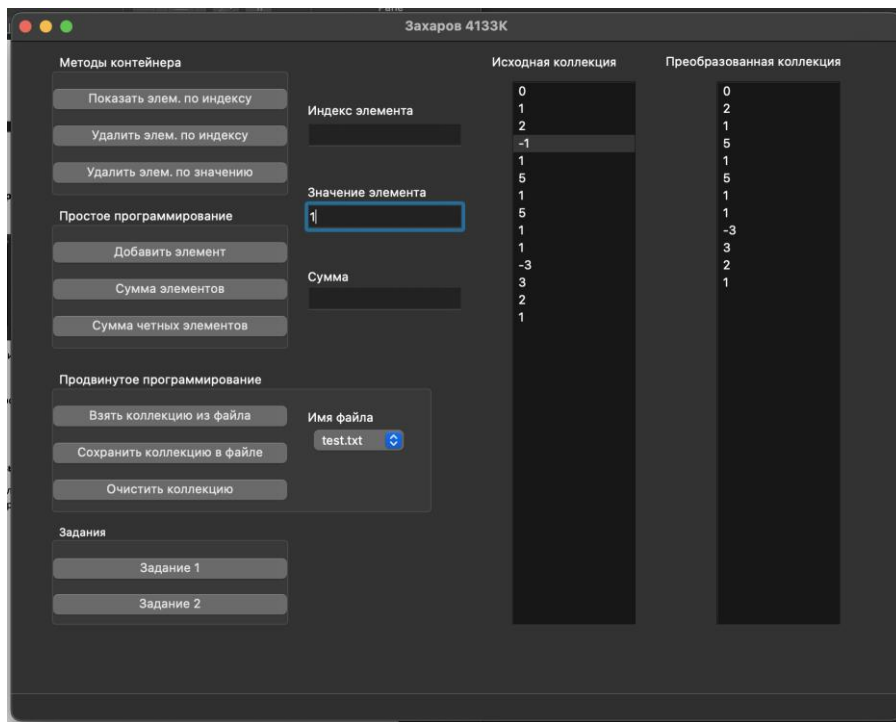
Показываем элемент по индексу



Удаляем элемент по индексу

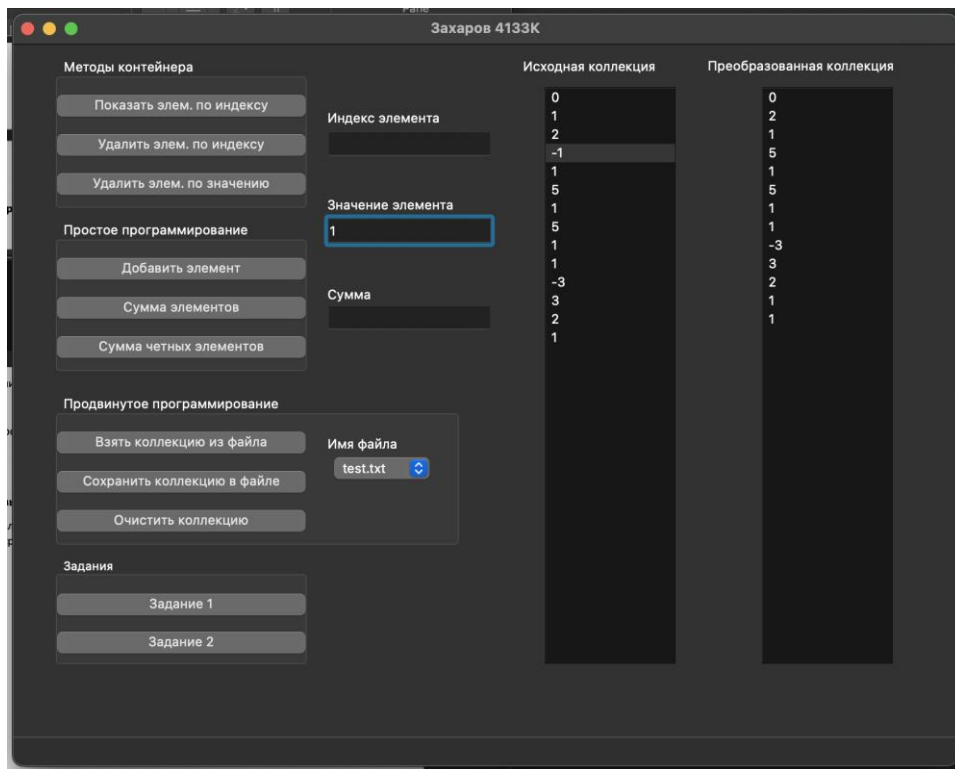


Удаляем элемент по значению



Простое программирование:

Добавляем элемент



Считаем сумму элементов

Методы контейнера

Показать элем. по индексу

Удалить элем. по индексу

Удалить элем. по значению

Простое программирование

Добавить элемент

Сумма элементов

Сумма четных элементов

Продвинутое программирование

Взять коллекцию из файла

Сохранить коллекцию в файле

Очистить коллекцию

Задания

Задание 1

Задание 2

Имя файла

test.txt

Индекс элемента

Значение элемента

Сумма

20

Исходная коллекция

0

1

2

-1

1

5

1

5

1

5

1

-3

3

2

1

Преобразованная коллекция

0

2

1

5

1

5

1

-3

3

2

1

-3

3

2

1

Считаем сумму чётных элементов

Методы контейнера

Показать элем. по индексу

Удалить элем. по индексу

Удалить элем. по значению

Простое программирование

Добавить элемент

Сумма элементов

Сумма четных элементов

Продвинутое программирование

Взять коллекцию из файла

Сохранить коллекцию в файле

Очистить коллекцию

Задания

Задание 1

Задание 2

Имя файла

test.txt

Индекс элемента

Значение элемента

Сумма

4

Исходная коллекция

0

1

2

-1

1

5

1

5

1

5

1

-3

3

2

1

Преобразованная коллекция

0

2

1

5

1

5

1

-3

3

2

1

-3

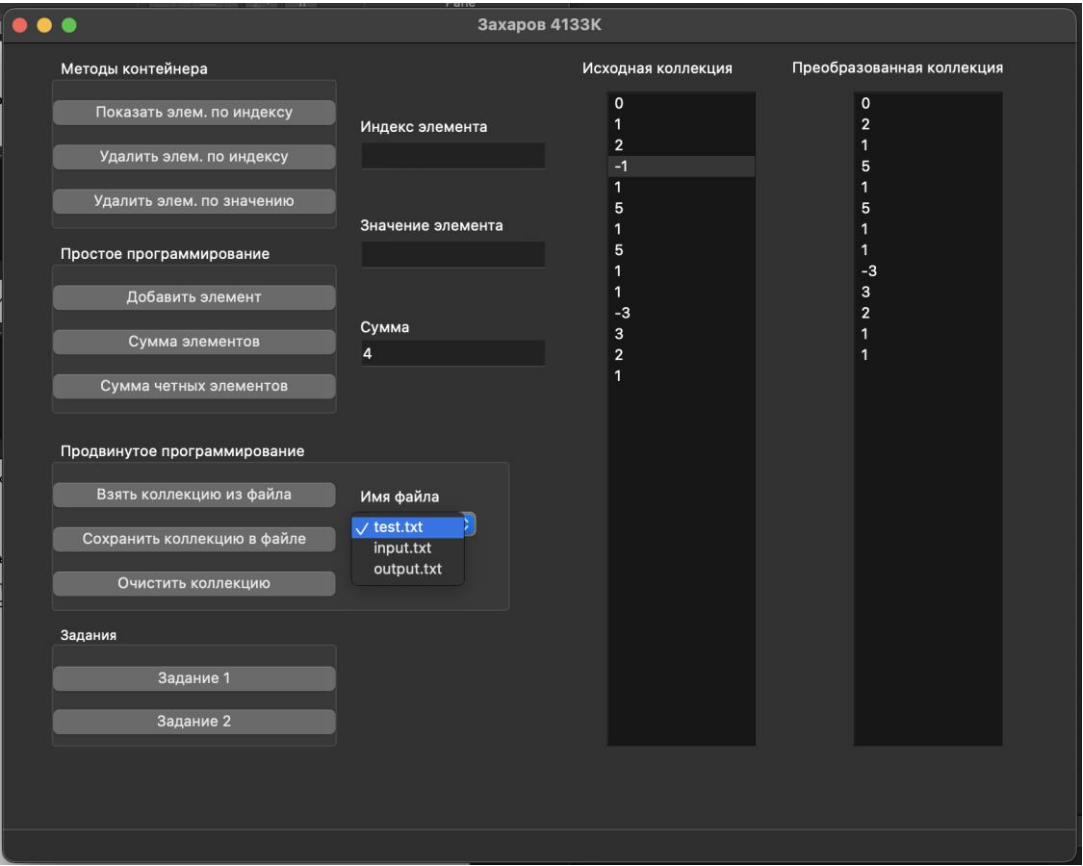
3

2

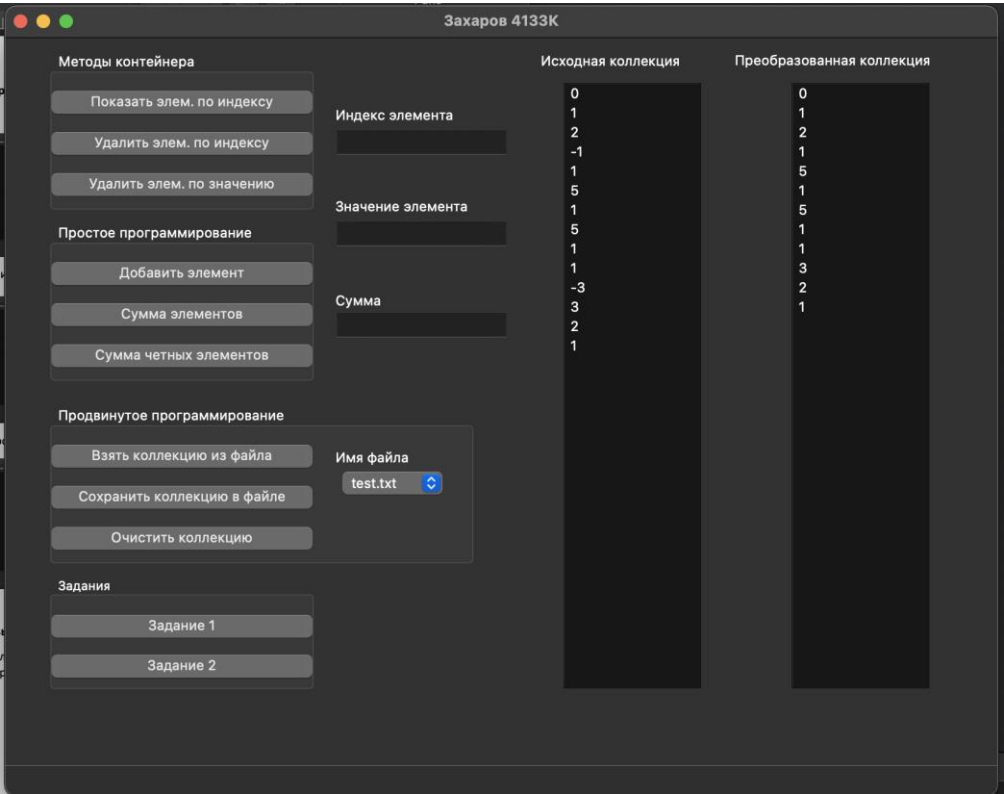
1

Продвинутое программирование, задание 1 и 2:

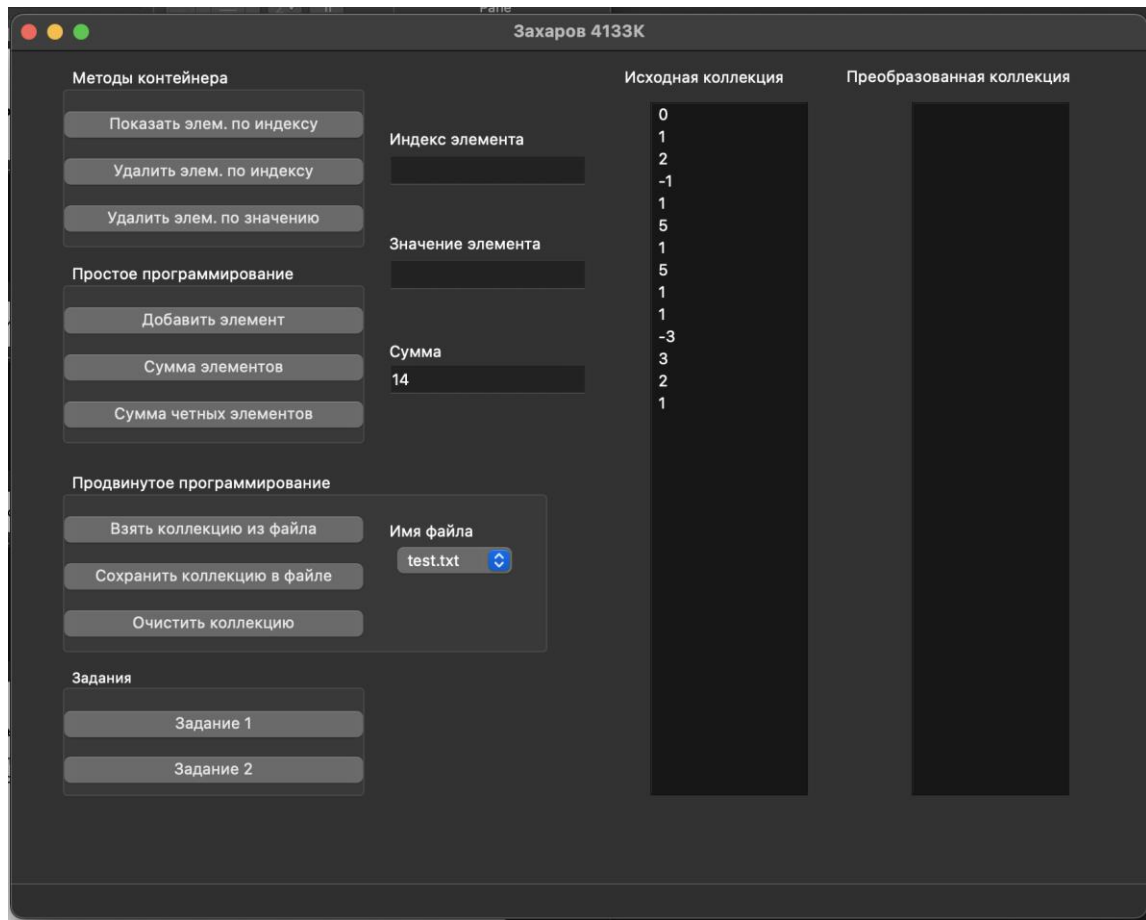
Берём из файла/сохраняем в файл



Задание 1



Задание 2



5. Анализ результатов и выводы

В результате выполнения лабораторной работы я ознакомился с использованием контейнерных классов в среде Qt Creator.