

## 1. Цель работы

научиться использовать контейнерные классы в среде Qt Creator.

## 2. Задание

Разработайте форму, подобную приведенной на рисунке, сохранив весь функционал предложенный на форме. А также добавьте 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

Новая последовательность сначала должна содержать все элементы исходной, значение которых по модулю меньше 1, затем - все остальные (с сохранением исходного относительного порядка).

7) Выполните задание 2 согласно варианта

Сумму элементов последовательности от начала до последнего положительного элемента (включительно).

8) Если для выполнения задания потребуются еще дополнительные виджеты, то студент может их добавить на форму

Работа с контейнерами

Пишите здесь

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

Исходная коллекция

Преобразованная коллекция

Индекс элемента

Значение элемента

Сумма

Имя файла

Рисунок 1 – Форма по заданию

### 3. Форма программы

Пишите здесь

Методы контейнеров

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 файл

Очистить коллекцию

Задания по варианту

Специальная сортировка

Сумма от начала до последнего положительного

Индекс элемента

Значение элемента

Сумма

Имя файла

Сумма

Исходная коллекция

Преобразованная коллекция

#### **4. Описание формы**

- QPushButton – addElemButton
- QPushButton – clearCollectionButton
- QPushButton – collectionMerhodsTitle
- QLabel – collectionMethodsTitle
- QLabel – easyProgTitle
- QTextEdit – editedCollectionEdit
- QLabel – editedCollectionTitle
- QPushButton – elemByIndexButton
- QLineEdit – elemIndexEdit
- QLabel – elemIndexTitle
- QLineEdit – elemVarEdit
- QLabel – elemVarTitle
- QcomboBox – fileChooser
- QLabel – fileName
- QLabel – proProg
- QPushButton – removeElemByIndexButton
- QPushButton – removeElemByVarButon
- QPushButton – saveCollectionToFileButton
- QPushButton – sortButton
- QTextEdit – sourceCollectionEdit
- QLabel – sourceCollectionTitle
- QLineEdit – sumEdit
- QPushButton – sumOfElemButton
- QPushButton – sumOfEvenButton
- QPushButton – sumOfPositiveButton
- QLineEdit – sumPositiveEdit
- QLabel – sumPositiveTitle
- QLabel – sumTitle
- QLabel - varTask

#### **5. Текст программы**

### **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 <QFile>
#include <QDir>

using namespace std;

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

    connect(ui->elemIndexEdit, &QLineEdit::textEdited, this, &MainWindow::indexChanged);
    connect(ui->elemVarEdit, &QLineEdit::textEdited, this, &MainWindow::elemChanged);

    connect(ui->saveCollectionToFileButton, &QPushButton::pressed, this,
    &MainWindow::loadToFile);
    connect(ui->collectionFromFileButton, &QPushButton::pressed, this,
    &MainWindow::loadFromFile);
```

```

        connect(ui->clearCollectionButton, &QPushButton::pressed, this,
&MainWindow::clearCollection);

        connect(ui->elemByIndexButton, &QPushButton::pressed, this,
&MainWindow::getElemByIndex);

        connect(ui->removeElemByIndexButton, &QPushButton::pressed, this,
&MainWindow::removeElemByIndex);

        connect(ui->removeElemByVarButton, &QPushButton::pressed, this,
&MainWindow::removeElemByVar);

        connect(ui->addElemButton, &QPushButton::pressed, this,
&MainWindow::addElemToEdited);

        connect(ui->sumOfElemButton, &QPushButton::pressed, this, &MainWindow::summElem);

        connect(ui->sumOfEvenButton, &QPushButton::pressed, this,
&MainWindow::summEvenElem);

        connect(ui->sumOfPositiveButton, &QPushButton::pressed, this,
&MainWindow::summPositiveElem);

        connect(ui->sortButton, &QPushButton::pressed, this, &MainWindow::sortElem);


        connect(ui->fileChooser, &QComboBox::currentTextChanged, this,
&MainWindow::fileChanged);


        connect(ui->editedCollectionEdit, &QTextEdit::textChanged, this,
&MainWindow::editedCollectionChanged);


        fileLoad();
    }

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

void MyList::updateSize(int arg1)
{
    this->listSize += arg1;
}

```

```
void MyList::insert(float arg1)
```

```
{  
    List.append(arg1);  
    updateSize(1);  
}
```

```
int MyList::size()
```

```
{  
    return this->listSize;  
}
```

```
float MyList::at(int arg1)
```

```
{  
    return List.at(arg1);  
}
```

```
void MyList::clear()
```

```
{  
    List.clear();  
    this->listSize = 0;  
}
```

```
bool MyList::isEmpty()
```

```
{  
    return List.isEmpty();  
}
```

```
void MyList::takeAt(int arg1)
```

```
{  
    List.takeAt(arg1);  
    updateSize(-1);  
}
```

```
int MyList::indexOf(float arg1)
```

```
{  
    return List.indexOf(arg1);  
}
```

```
void MyList::copy(MyList arg1)  
{  
    this->clear();  
  
    QList<float>::iterator it = arg1.begin();  
    for (; it != arg1.end(); ++it) {  
        this->insert(*it);  
    }  
}
```

```
QList<float>::iterator MyList::begin()  
{  
    return List.begin();  
}
```

```
QList<float>::iterator MyList::end()  
{  
    return List.end();  
}
```

```
void MainWindow::fileLoad()  
{  
    QDir directory;  
    QStringList files = directory.entryList(QStringList() << "list-*.txt",QDir::Files);  
    ui->fileChooser->addItem(files);  
}
```

```
void MainWindow::indexChanged(QString arg1)
```

```

{
    if ((arg1.toInt() < 0) or (arg1.toInt() >= EditedList.size()))
    {
        if (arg1.toInt() >= EditedList.size())
        {
            if (EditedList.size() == 0)
            {
                ui->elemIndexEdit->clear();
            }
            else
            {
                ui->elemIndexEdit->setText(QString::number(EditedList.size()-1));
            }
        }
        else
        {
            ui->elemIndexEdit->clear();
        }
    }

    if (ui->elemIndexEdit->text().isEmpty() or EditedList.isEmpty())
    {
        ui->elemByIndexButton ->setDisabled(true);
        ui->removeElemByIndexButton->setDisabled(true);
    }
    else
    {
        ui->elemByIndexButton ->setDisabled(false);
        ui->removeElemByIndexButton->setDisabled(false);
    }
}

```

```

void MainWindow::elemChanged()

```



```

{
    if (ui->elemVarEdit->text().isEmpty() or EditedList.isEmpty())
    {
        ui->removeElemByVarButton->setDisabled(true);
        ui->addElemButton->setDisabled(ui->elemVarEdit->text().isEmpty());
    }
    else
    {
        ui->removeElemByVarButton->setDisabled(false);
        ui->addElemButton->setDisabled(false);
    }
}

```

```

void MainWindow::updateSource()
{
    QString str;
    QList<float>::iterator it = SourceList.begin();
    for (; it != SourceList.end(); ++it) {
        str += QString::number(*it) + '\n';
    }
    ui->sourceCollectionEdit->setText(str);
}

```

```

void MainWindow::updateEdited()
{
    QString str;
    QList<float>::iterator it = EditedList.begin();
    for (; it != EditedList.end(); ++it) {
        str += QString::number(*it) + '\n';
    }
    ui->editedCollectionEdit->setText(str);
}

```

```

void MainWindow::loadToFile()

```

```

{
    QFile myfile(ui->fileChooser->currentText());
    if (!myfile.open(QIODevice::WriteOnly))
    {
        qInfo() << "Что-то пошло не так!";
        return;
    }

    QTextStream stream(&myfile);
    for (int i{0}; i<EditedList.size(); i++)
    {
        stream << QString::number(EditedList.at(i)) << "\n";
    }

    myfile.close();
}

```

```

void MainWindow::loadFromFile()
{
    SourceList.clear();
    EditedList.clear();
    QFile myfile(ui->fileChooser->currentText());
    if (!myfile.open(QIODevice::ReadOnly))
    {
        qInfo() << "Что-то пошло не так!";
        return;
    }

    QTextStream stream(&myfile);
    while (!stream.atEnd())
    {
        QString element = stream.readLine();
        SourceList.insert(element.toFloat());
        EditedList.insert(element.toFloat());
    }
}

```

```

updateSource();
updateEdited();

if (not(EditedList.isEmpty()))
    ui->clearCollectionButton->setDisabled(false);
indexChanged(ui->elemIndexEdit->text());
elemChanged();

}

void MainWindow::ClearEdit()
{
    ui->elemIndexEdit->clear();
    ui->elemVarEdit->clear();
    ui->sumEdit->clear();
    ui->sumPositiveEdit->clear();
}

void MainWindow::fileChanged()
{
    SourceList.clear();
    ui->sourceCollectionEdit->clear();
}

void MainWindow::clearCollection()
{
    EditedList.clear();
    ui->editedCollectionEdit->clear();
    ui->clearCollectionButton->setDisabled(true);
}

void MainWindow::editedCollectionChanged()
{
    ClearEdit();
}

```

```

        indexChanged(ui->elemIndexEdit->text());
        elemChanged();
    }

void MainWindow::getElemByIndex()
{
    ui->elemVarEdit->setText(QString::number(EditedList.at(ui->elemIndexEdit-
>text().toInt())));
}

void MainWindow::removeElemByIndex()
{
    EditedList.takeAt(ui->elemIndexEdit->text().toInt());
    updateEdited();
}

void MainWindow::removeElemByVar()
{
    int index = EditedList.indexOf(ui->elemVarEdit->text().toFloat());
    if (index != -1)
    {
        EditedList.takeAt(index);
    }
    updateEdited();
}

void MainWindow::addElemToEdited()
{
    EditedList.insert(ui->elemVarEdit->text().toFloat());
    updateEdited();
}

void MainWindow::summElem()
{
    float sum{0};

```

```

    QList<float>::iterator it = EditedList.begin();
    for (; it != EditedList.end(); ++it) {
        sum += *it;
    }

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

```

```

void MainWindow::summEvenElem()
{
    float sum{0};

```

```

    QList<float>::iterator it = EditedList.begin();
    for (; it != EditedList.end(); ++it) {
        if ((int)*it % 2 == 0)
            sum += *it;
    }

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

```

```

void MainWindow::summPositiveElem()
{

```

```

    QList<float>::iterator it = EditedList.begin();
    QList<float>::iterator endforsum;

    for (; it != EditedList.end(); ++it) {
        if ((*it)>0)
            endforsum = it;
    }

```

```

    float sum{0};

```

```

it = EditedList.begin();

for (; it != endforsum; ++it) {
    sum += *it;
}

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

void MainWindow::sortElem()
{
    MyList TempList;

    QList<float>::iterator it = EditedList.begin();
    for (; it != EditedList.end(); ++it) {
        if (abs(*it)<1)
            TempList.insert(*it);
    }
    it = EditedList.begin();
    for (; it != EditedList.end(); ++it) {
        if (abs(*it)>=1)
            TempList.insert(*it);
    }

    EditedList.copy(TempList);

    updateEdited();
}

```

## **mainwindow.h**

```

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>

```

```
#include <qtextedit.h>
```

```
QT_BEGIN_NAMESPACE
```

```
namespace Ui {
```

```
class MainWindow;
```

```
}
```

```
QT_END_NAMESPACE
```

```
class MyList
```

```
{
```

```
public:
```

```
    void insert(float);
```

```
    int size();
```

```
    float at(int);
```

```
    void clear();
```

```
    bool isEmpty();
```

```
    void takeAt(int);
```

```
    void copy(MyList);
```

```
    int indexOf(float);
```

```
    QList<float>::iterator begin();
```

```
    QList<float>::iterator end();
```

```
private:
```

```
    int listSize = 0;
```

```
    QList<float> List;
```

```
    void updateSize(int);
```

```
};
```

```
class MainWindow : public QMainWindow
```

```
{
```

```
    Q_OBJECT
```

public:

MainWindow(QWidget \*parent = nullptr);

~MainWindow();

private:

Ui::MainWindow \*ui;

MyList SourceList;

MyList EditedList;

void fileLoad();

bool editIsEmpty(QTextEdit \*);

void indexChanged(QString);

void elemChanged();

void fileChanged();

void ClearEdit();

void loadToFile();

void loadFromFile();

void updateSource();

void updateEdited();

void clearCollection();

void getElemByIndex();

void removeElemByIndex();

void editedCollectionChanged();

void removeElemByVar();

void addElemToEdited();

void summElem();

void summEvenElem();

void summPositiveElem();

void sortElem();



```
};
```

```
#endif // MAINWINDOW_H
```

## 6. Пример работы программы

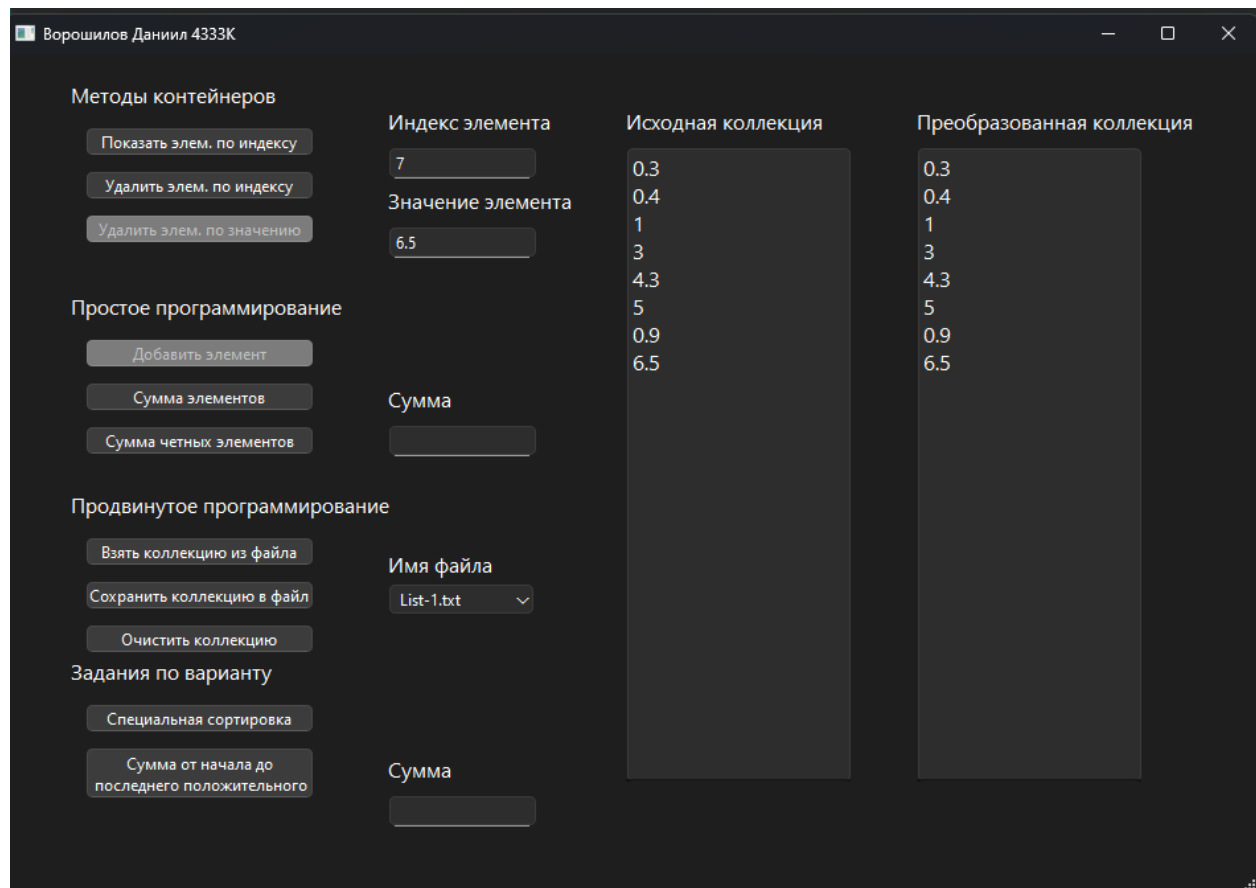


Рисунок 3 – Поиск по индексу

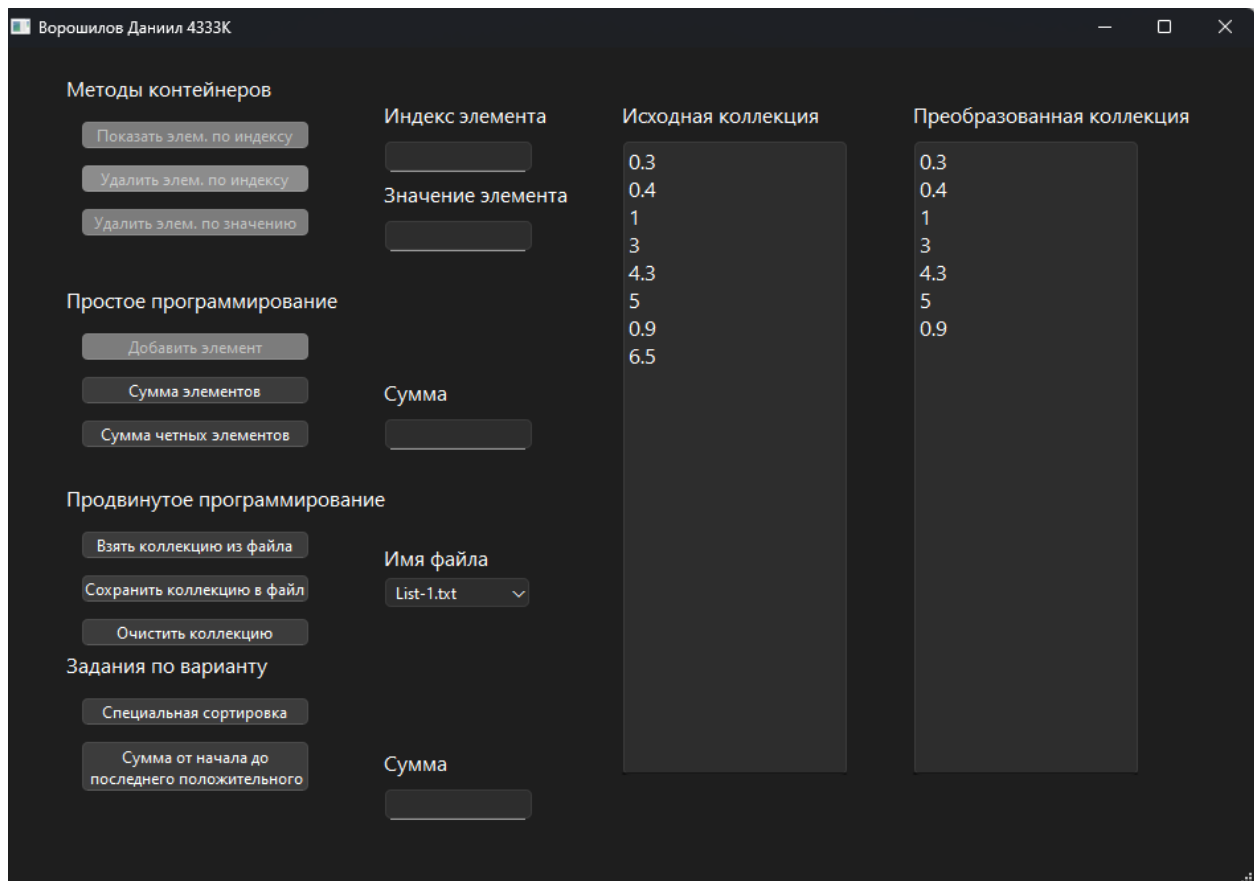


Рисунок 4 – Удаление по индексу 7

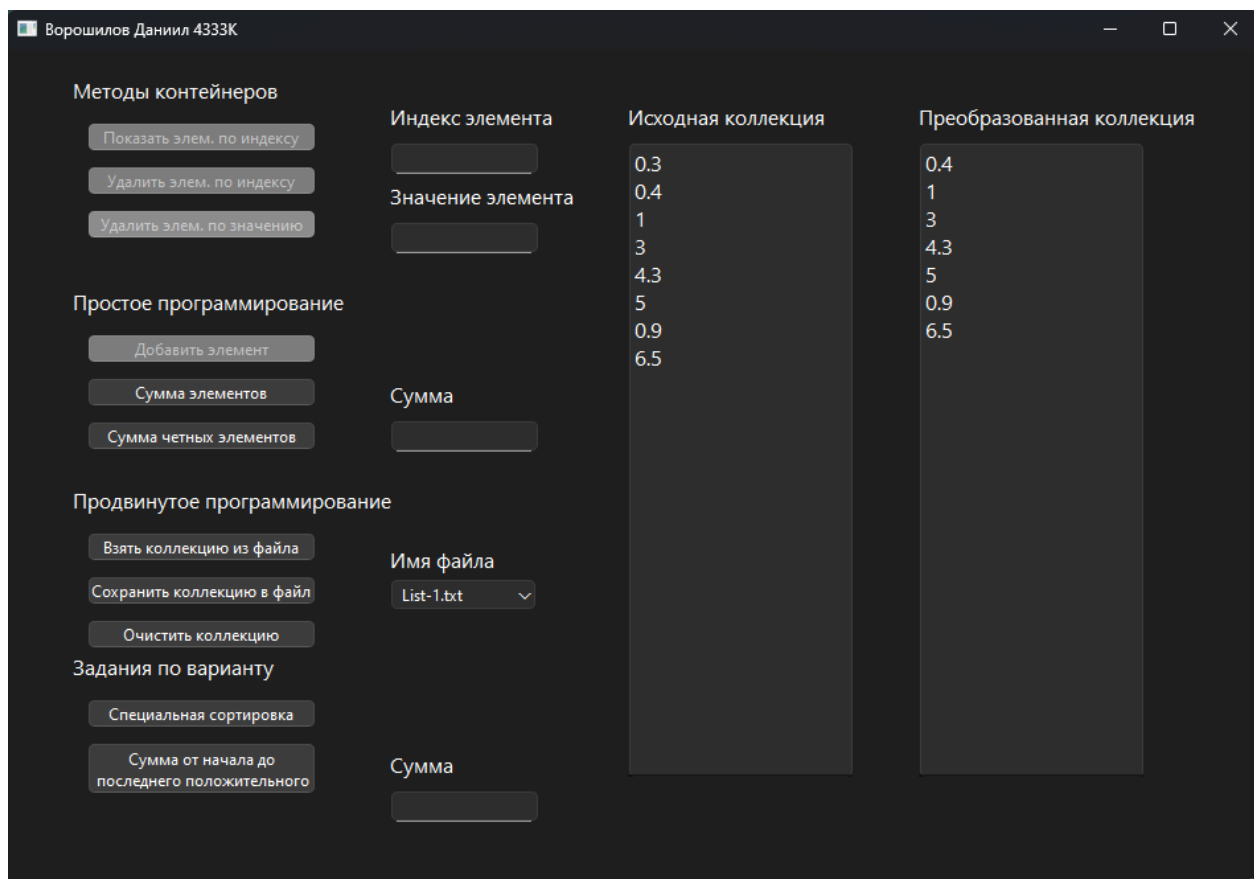


Рисунок 5 – Удаление по значению 0.3

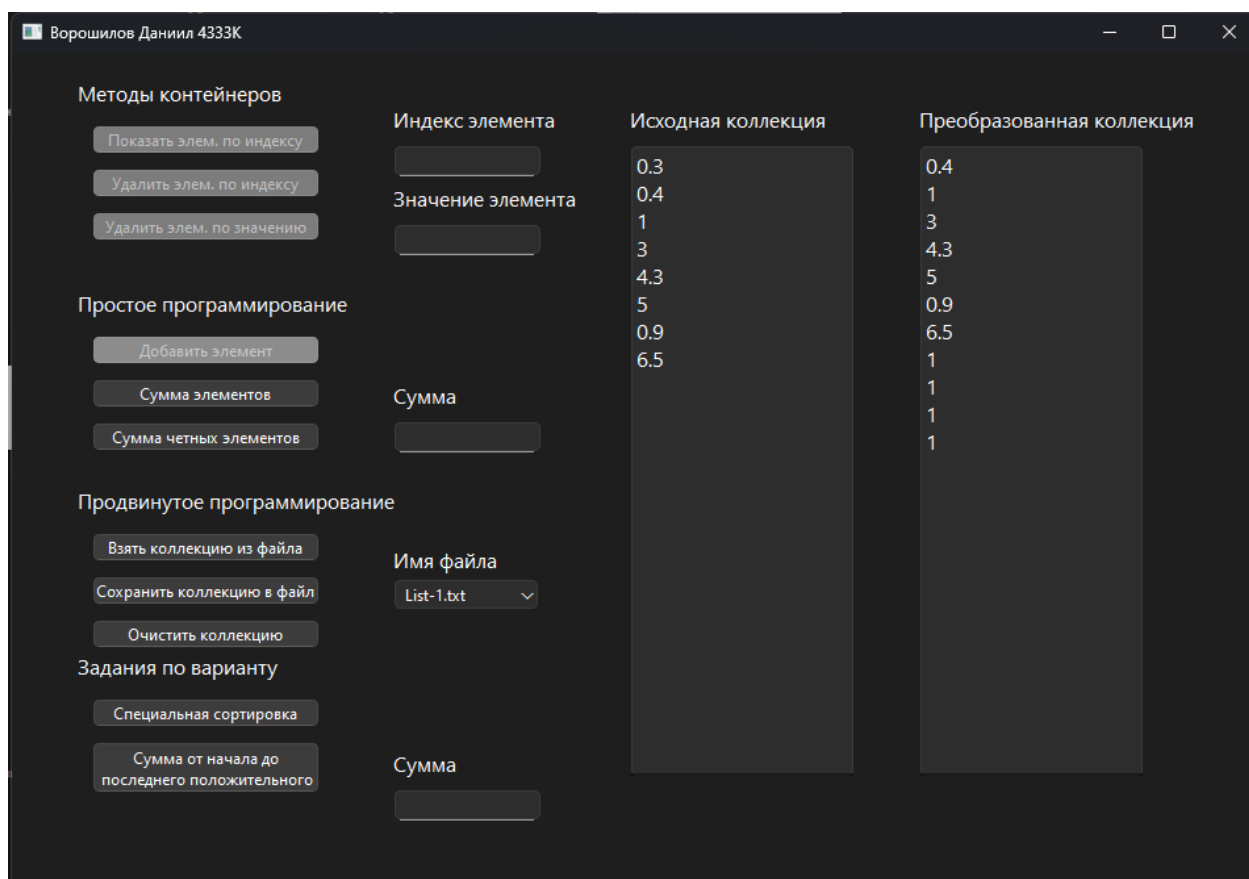


Рисунок 6 – Добавление элемента

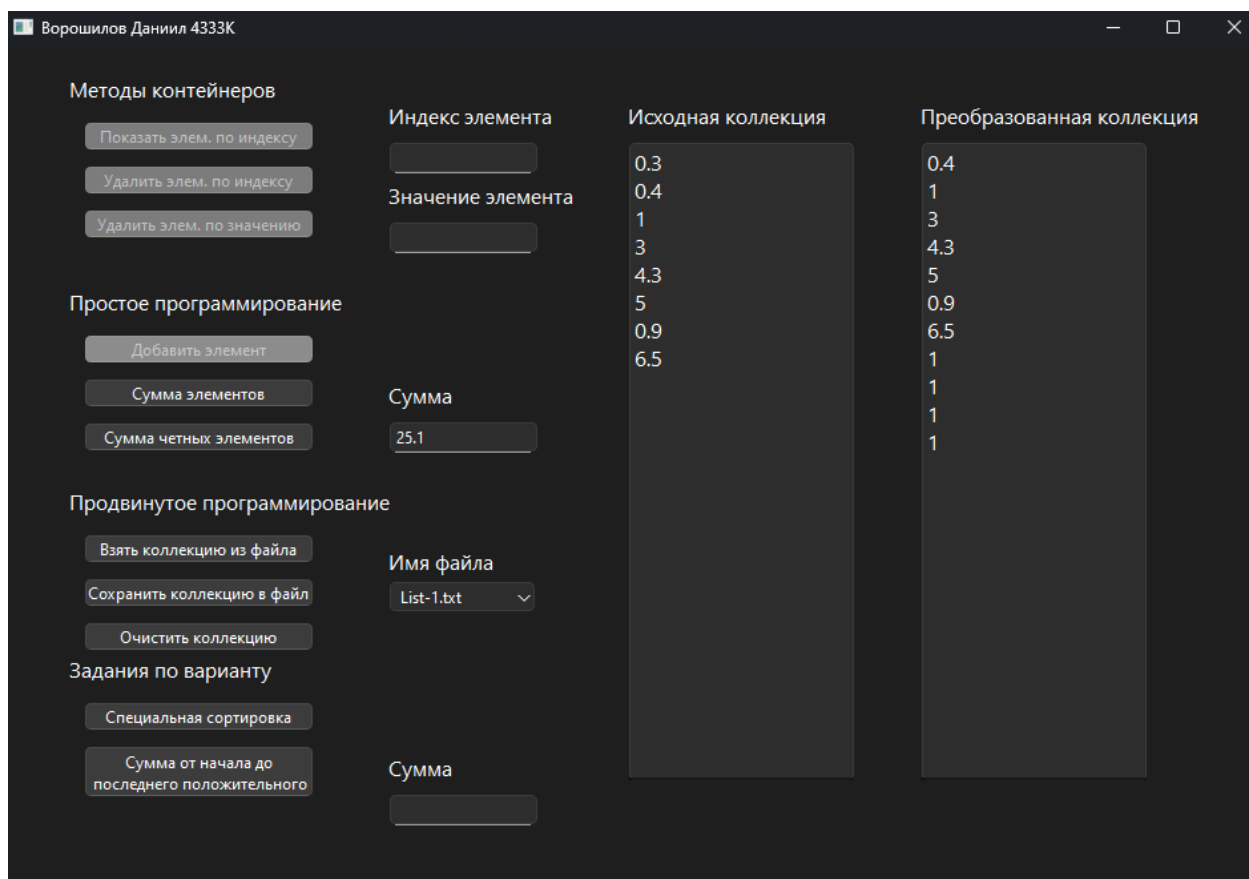


Рисунок 7 – Сумма

Ворошилов Даниил 4333K

Методы контейнеров

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 файл

Очистить коллекцию

Задания по варианту

Специальная сортировка

Сумма от начала до последнего положительного

Индекс элемента

Значение элемента

Сумма

Имя файла

List-3.txt

Исходная коллекция

1  
2  
3  
4  
5

Преобразованная коллекция

1  
2  
3  
4  
5

Сумма

6

Рисунок 8 – Сумма чётных

Ворошилов Даниил 4333K

Методы контейнеров

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 файл

Очистить коллекцию

Задания по варианту

Специальная сортировка

Сумма от начала до последнего положительного

Индекс элемента

Значение элемента

Сумма

Имя файла

List-5.txt

Исходная коллекция

-0.5  
0.3  
0.1  
3  
90  
0.9  
10  
-0.3

Преобразованная коллекция

-0.5  
0.3  
0.1  
0.9  
-0.3  
3  
90  
10

Сумма

Рисунок 9 – Сортировка по варианту

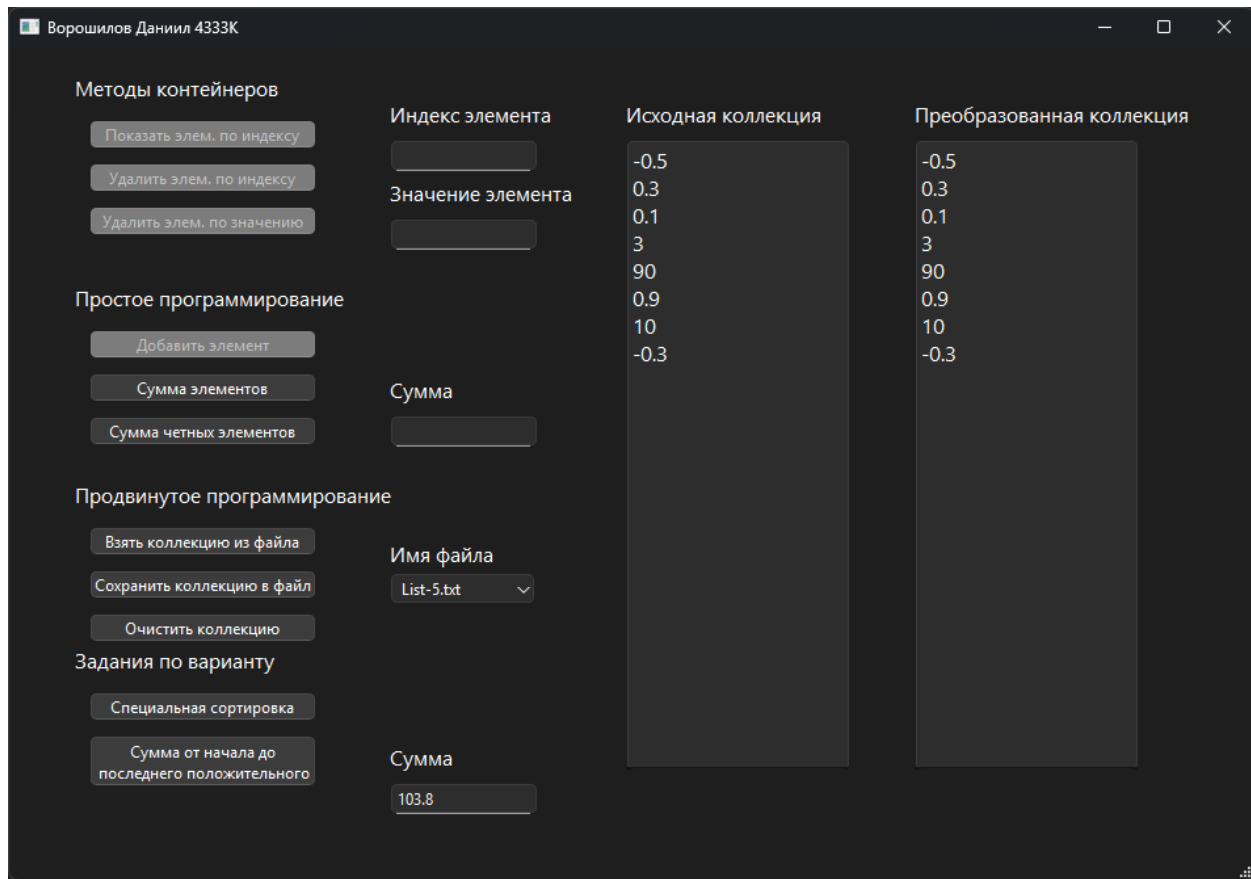


Рисунок 10 – Сумма от начала до положительного

## 7. Выводы

В процессе лабораторной работы были изучены контейнерные классы в QT