МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА 43

ОЦЕНКА

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | К.А. Кочин |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ЛАБОРАТОРНАЯ РАБОТА №5 |
| Использование контейнерных классов в среде Qt Creator |
| по дисциплине: Объектно-ориентированное программирование |
|  |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4136 |  |  |  | Бобрович Н.С. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

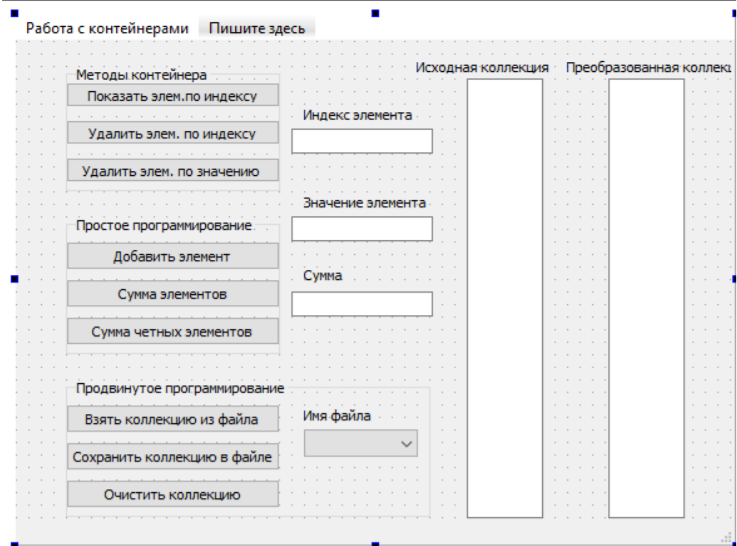
РАБОТУ ВЫПОЛНИЛ

Санкт-Петербург 2023

***Цель работы:*** Научиться использовать контейнерные классы в среде Qt Creator.

***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 пропищите 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.

***Задания по варианту:***

1) Новая последовательность должна содержать все неотрицательные элементы исходной (с сохранением исходного относительного порядка).

2) Сумму элементов последовательности, расположенных между первым и последним нулевыми элементами.

***Листинг:***

***Mainwindow.h***

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include<QList>

#include <QFileDialog>

#include <QFile>

#include <QVector>

#include <algorithm>

QT\_BEGIN\_NAMESPACE

namespace Ui { class MainWindow; }

QT\_END\_NAMESPACE

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

MainWindow(QWidget \* parent = nullptr);

~MainWindow();

QList<int> list;

QList<QString> lists;

QList<int> list1;

QString path;

int index;

int data;

void tostr();

bool test\_znach();

bool test\_ind();

private slots:

void on\_pushButton\_3\_clicked();

void on\_lineEdit\_textChanged(const QString &arg1);

void on\_lineEdit\_2\_textChanged(const QString &arg1);

void on\_pushButton\_4\_clicked();

void on\_pushButton\_clicked();

void on\_pushButton\_2\_clicked();

void on\_pushButton\_5\_clicked();

void on\_pushButton\_6\_clicked();

void on\_pushButton\_7\_clicked();

void on\_pushButton\_11\_clicked();

void on\_pushButton\_10\_clicked();

void on\_pushButton\_8\_clicked();

void on\_pushButton\_9\_clicked();

void on\_pushButton\_12\_clicked();

void on\_pushButton\_13\_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"

MainWindow::MainWindow(QWidget \* parent)

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~MainWindow()

{

delete ui;

}

void MainWindow::tostr()

{

lists.reserve(list.size());

for (int i : list)

{

lists.append(QString::number(i));

}

}

bool MainWindow::test\_znach()

{

if (ui->lineEdit\_2->text() == "")

{

ui->lineEdit\_2->setText("Введите значение!");

return false;

}

else

return true;

}

bool MainWindow::test\_ind()

{

if (ui->lineEdit->text() == "")

{

ui->lineEdit->setText("Введите значение!");

return false;

}

else

return true;

}

void MainWindow::on\_pushButton\_3\_clicked()

{

if (test\_znach())

{

for (int i = 0; i < list.size(); i++)

{

if (list.at(i) == data)

{

list.removeAt(i);

i--;

}

}

ui->listWidget\_2->clear();

tostr();

ui->listWidget\_2->addItems(lists);

lists.clear();

ui->lineEdit->clear();

ui->lineEdit\_2->clear();

ui->lineEdit\_3->clear();

}

}

void MainWindow::on\_lineEdit\_textChanged(const QString &arg1)

{

index = ui->lineEdit->text().toInt();

}

void MainWindow::on\_lineEdit\_2\_textChanged(const QString &arg1)

{

data = ui->lineEdit\_2->text().toInt();

}

void MainWindow::on\_pushButton\_4\_clicked()

{

ui->listWidget->clear();

list.push\_back(data);

tostr();

ui->listWidget->addItems(lists);

lists.clear();

ui->lineEdit->clear();

ui->lineEdit\_2->clear();

ui->lineEdit\_3->clear();

}

void MainWindow::on\_pushButton\_clicked()

{

if (test\_ind())

{

ui->listWidget\_2->clear();

for (int i = 0; i < list.size(); i++)

{

if (i == index)

{

ui->listWidget\_2->addItem(QString::number(list.at(i)));

}

}

ui->lineEdit->clear();

ui->lineEdit\_3->clear();

}

}

void MainWindow::on\_pushButton\_2\_clicked()

{

if (test\_ind())

{

ui->listWidget\_2->clear();

list.removeAt(index);

tostr();

ui->listWidget\_2->addItems(lists);

lists.clear();

ui->lineEdit->clear();

ui->lineEdit\_3->clear();

}

}

void MainWindow::on\_pushButton\_5\_clicked()

{

ui->lineEdit\_3->clear();

int summ = 0;

for (int i = 0; i < list.size(); i++)

{

summ = summ + list.at(i);

}

ui->lineEdit\_3->setText(QString::number(summ));

ui->lineEdit->clear();

ui->lineEdit\_2->clear();

}

void MainWindow::on\_pushButton\_6\_clicked()

{

ui->lineEdit\_3->clear();

int summ = 0;

for (int i = 0; i < list.size(); i++)

{

if (list.at(i) % 2 == 0)

summ = summ + list.at(i);

}

ui->lineEdit\_3->setText(QString::number(summ));

ui->lineEdit->clear();

ui->lineEdit\_2->clear();

}

void MainWindow::on\_pushButton\_7\_clicked()

{

ui->listWidget\_2->clear();

for (int i = 0; i < list.size(); i++)

{

if (list.at(i) >= 0)

{

list1.push\_back(list.at(i));

}

}

lists.reserve(list1.size());

for (int j : list1)

{

lists.append(QString::number(j));

}

ui->listWidget\_2->addItems(lists);

lists.clear();

list1.clear();

ui->lineEdit->clear();

ui->lineEdit\_2->clear();

ui->lineEdit\_3->clear();

}

void MainWindow::on\_pushButton\_11\_clicked()

{

QString k = "/";

QString k1 = "\\";

path = QFileDialog::getOpenFileName(0, "Open Dialog", "", "\*txt");

path.replace(k1, k);

}

void MainWindow::on\_pushButton\_10\_clicked()

{

QString str;

QFile file(path);

if (!file.open(QIODevice::ReadOnly))

return;

str = file.readAll();

qDebug() << str;

QStringList lst;

lst = str.split('|');

list.clear();

ui->listWidget->clear();

for (int i = 0; i < lst.size(); i++)

{

list.push\_back(lst.at(i).toInt());

}

file.close();

ui->listWidget->addItems(lst);

lst.clear();

}

void MainWindow::on\_pushButton\_8\_clicked()

{

QString str = "";

QFile file(path);

if (!file.open(QIODevice::WriteOnly)) // Проверяем, возможно ли открыть наш файл для чтения

return;

for (int i = 0; i < list.size(); i++)

{

str = str + QString::number(list.at(i));

str = str + "|";

}

str.resize(str.size() - 1);

file.seek(0);

file.write(str.toUtf8());

file.close();

}

void MainWindow::on\_pushButton\_9\_clicked()

{

list.clear();

ui->listWidget->clear();

ui->listWidget\_2->clear();

}

void MainWindow::on\_pushButton\_12\_clicked()

{

int first\_zero = 0;

int last\_zero = 0;

int res = 0;

for (int i = 0; i < list.size(); i++)

{

if (list.at(i) == 0)

{

first\_zero = i;

break;

}

}

for (int i = list.size() - 1; i > 0; i--)

{

if (list.at(i) == 0)

{

last\_zero = i;

break;

}

}

if (first\_zero < last\_zero)

{

for (int i = first\_zero; i < last\_zero; i++)

{

res += list.at(i);

}

ui->lineEdit\_3->setText(QString::number(res));

}

else

{

ui->lineEdit\_3->setText("Нет нулей");

}

}

void MainWindow::on\_pushButton\_13\_clicked()

{

ui->listWidget->clear();

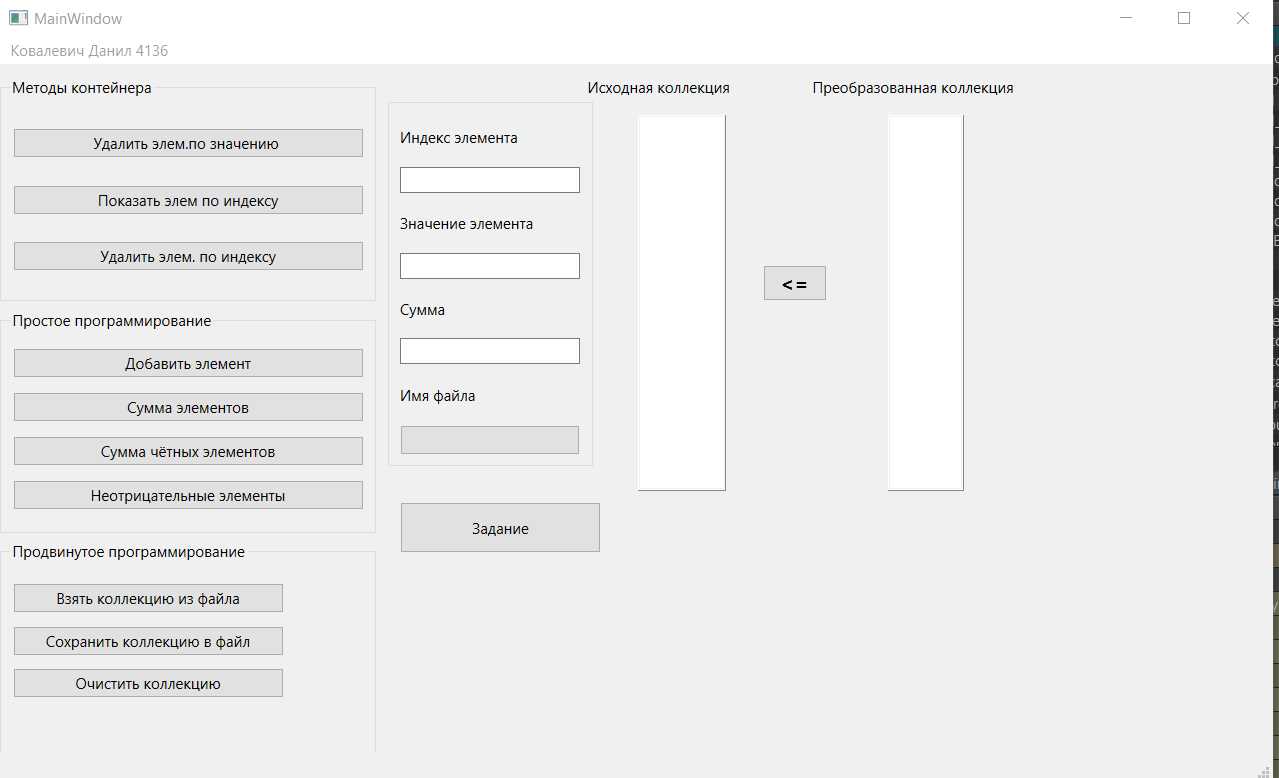
ui->listWidget\_2->clear();

tostr();

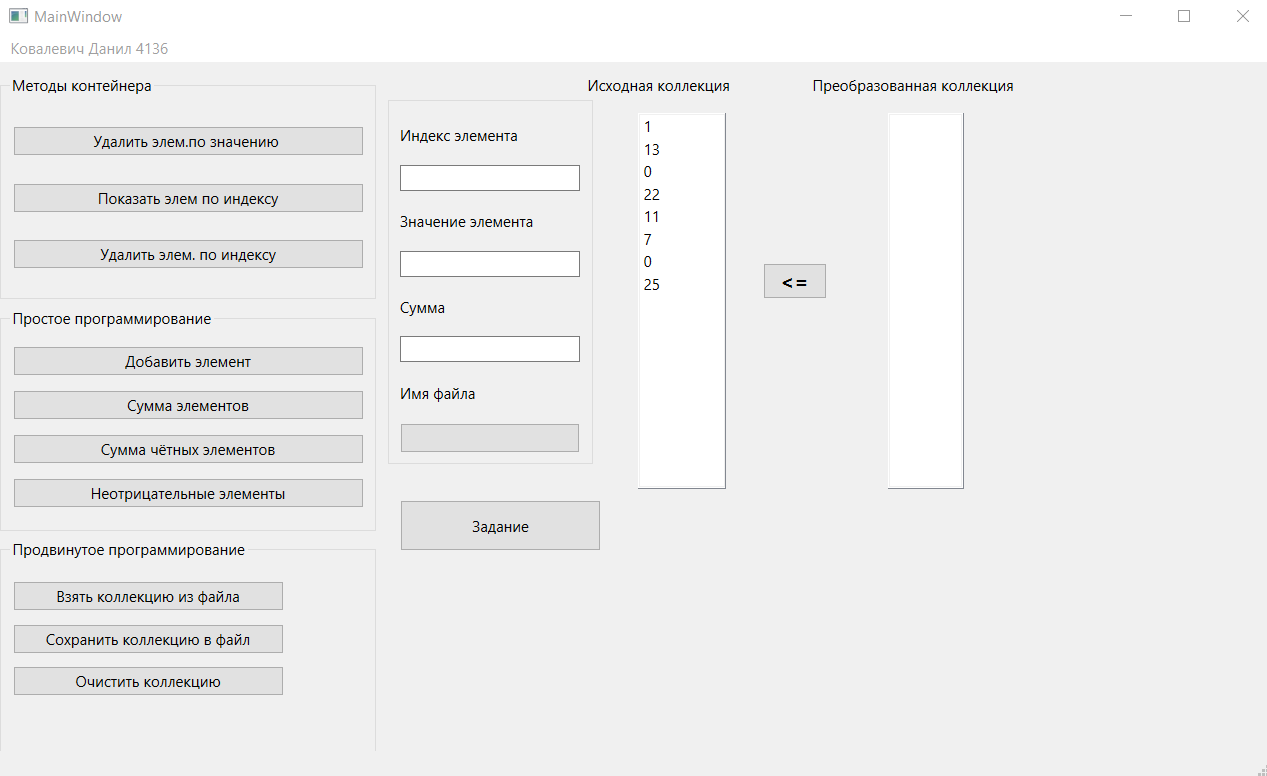
ui->listWidget->addItems(lists);

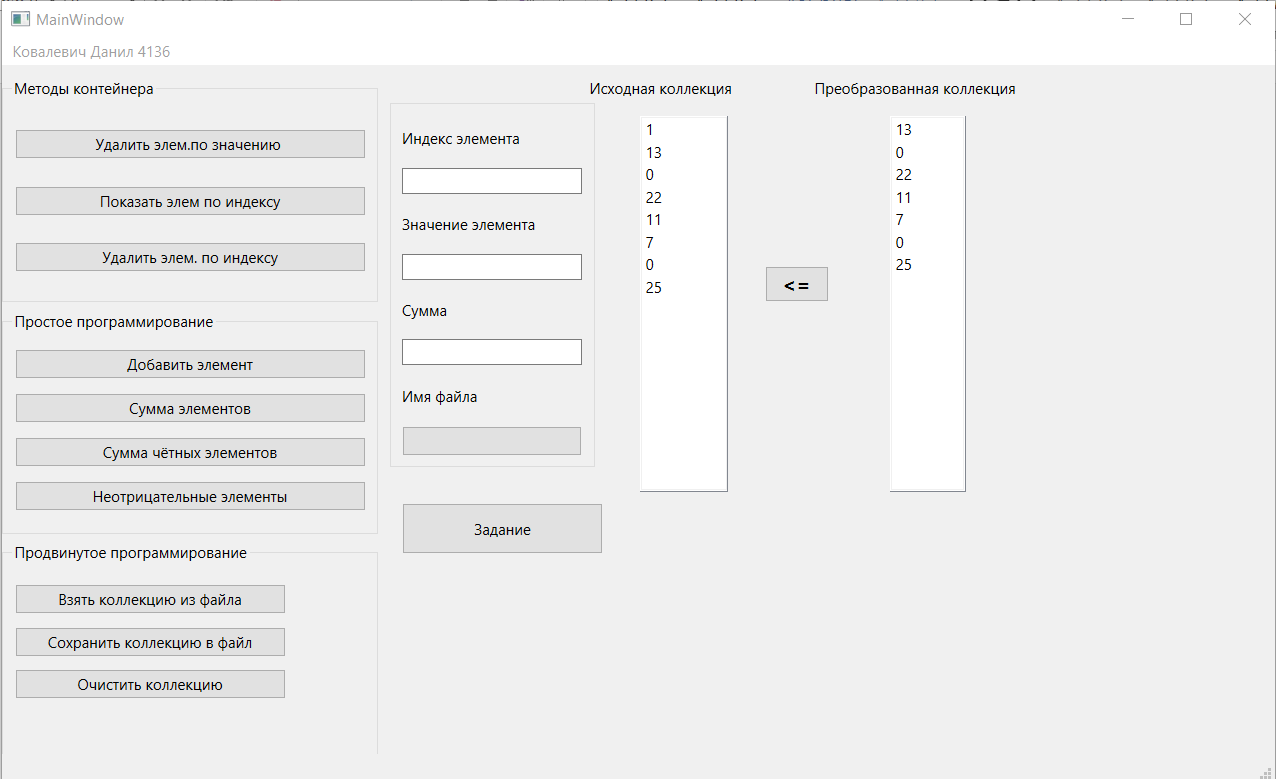
lists.clear();

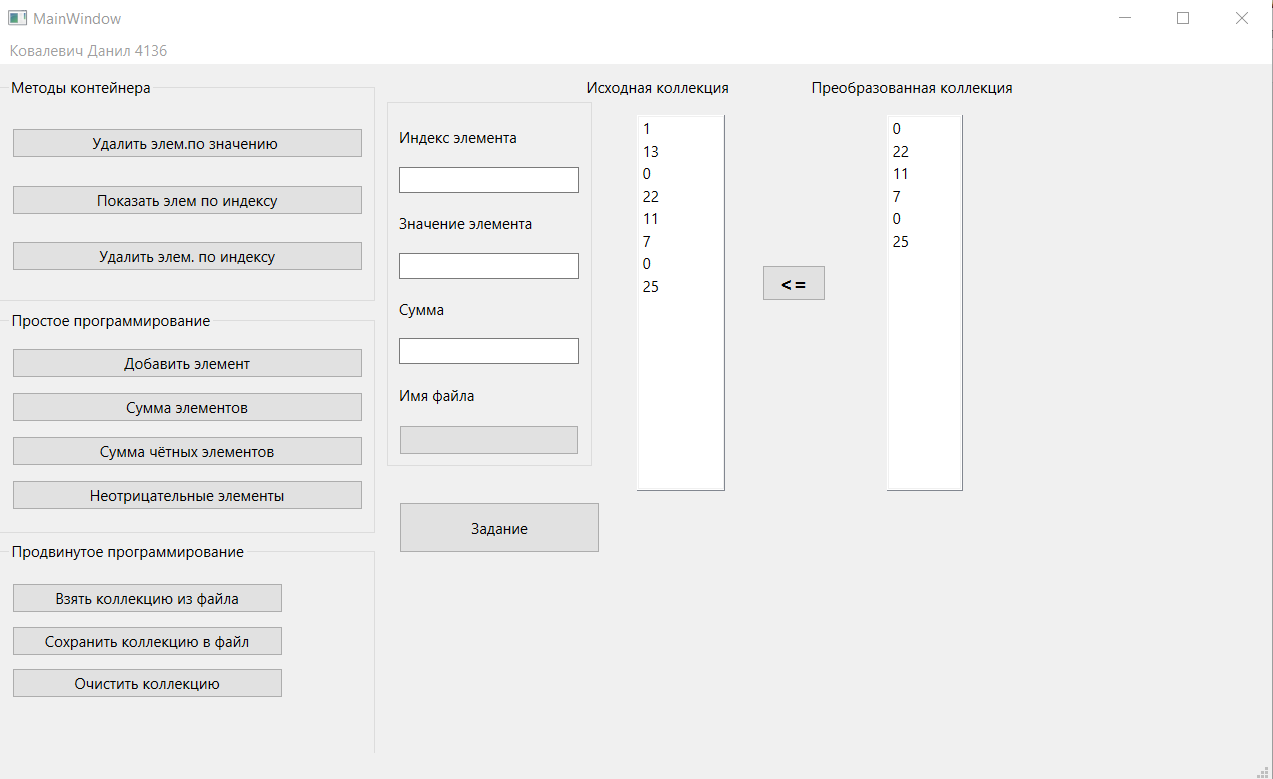
}

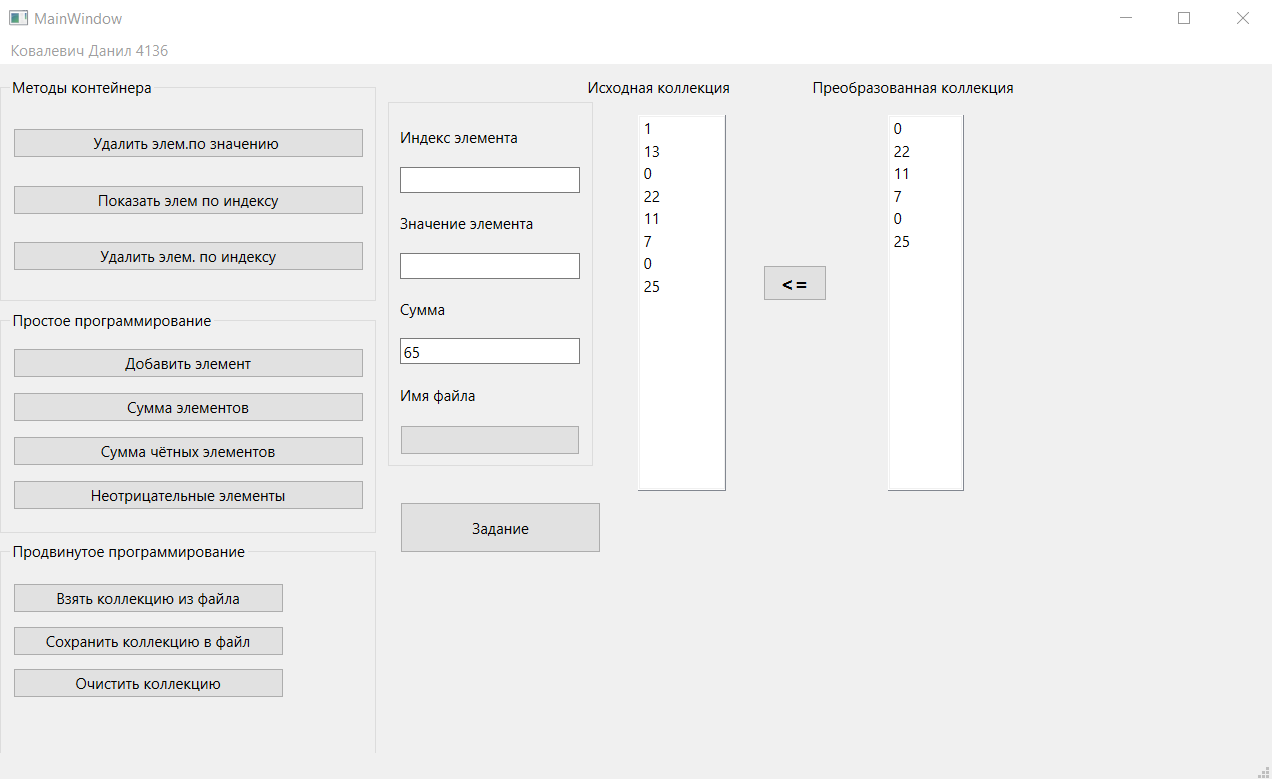


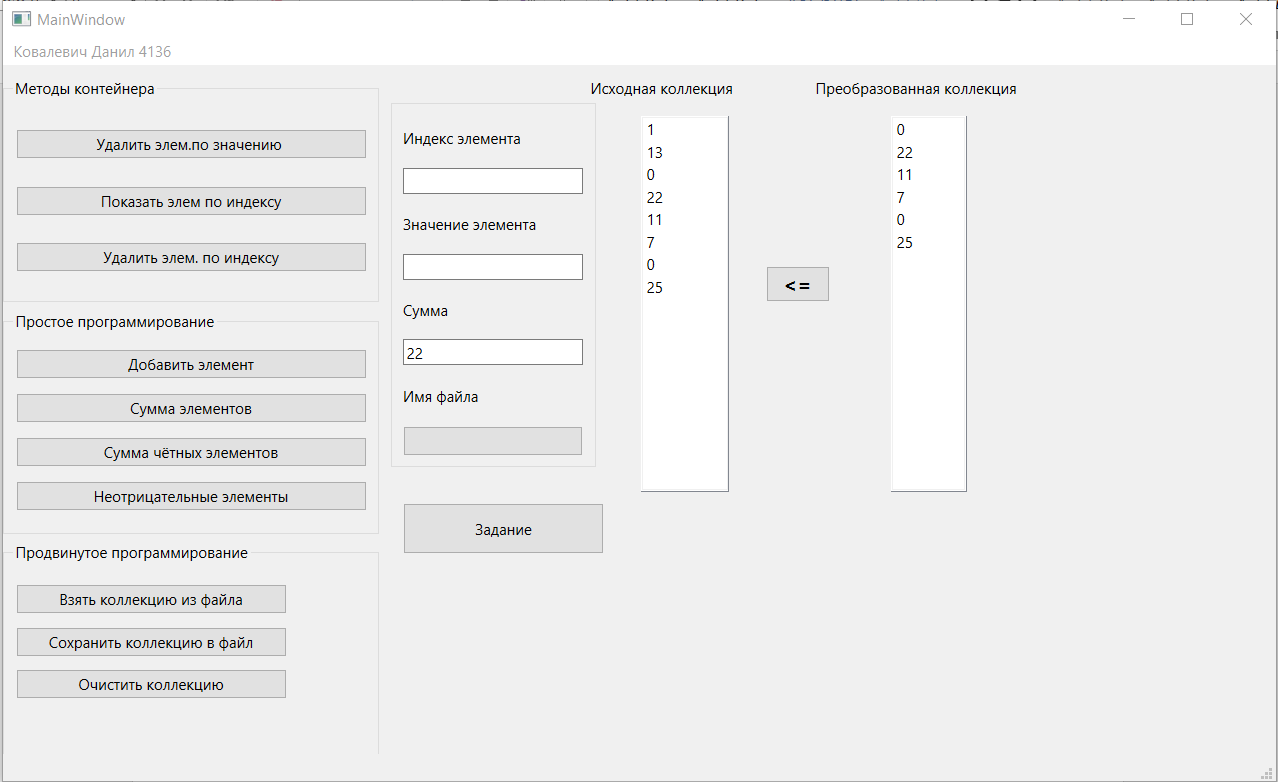
***Результат работы программы:***

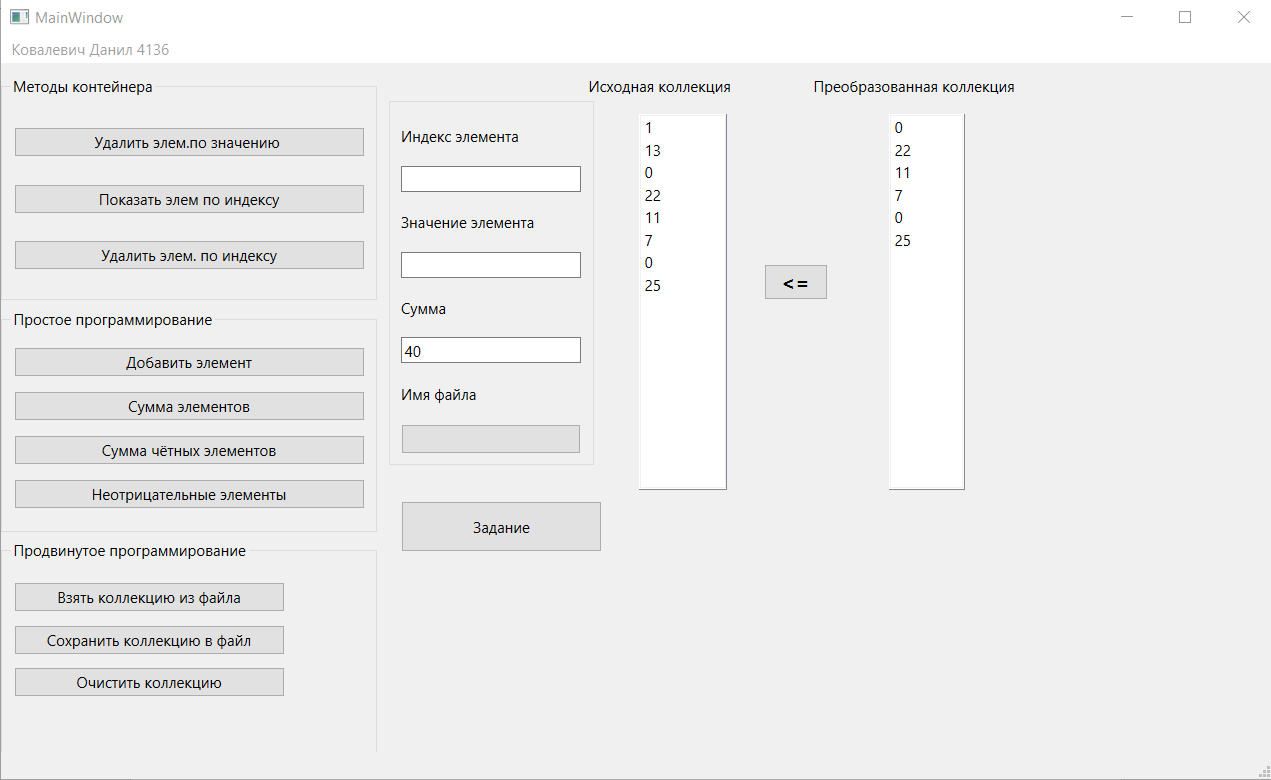












***Вывод:***

Я научился работать с контейнерными классами в среде QT Creator, закрепил полученные знания, написав desktop приложение.