



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное
образовательное учреждение высшего образования
**«Балтийский государственный технический университет «ВОЕНМЕХ»
им. Д.Ф. Устинова»**
(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова)

Факультет	<u>О</u> шифр	<u>Естественнонаучный</u> наименование
Кафедра	<u>О7</u> шифр	<u>Информационные системы и программная инженерия</u> наименование
Дисциплина	<u>Визуальное программирование</u>	

Лабораторная работа №2

«Работа с файлами XML и JSON»

ВЫПОЛНИЛ студент группы И596

Орехов. Р. В.
Фамилия И.О.

ПРЕПОДАВАТЕЛЬ

Васюков В.М.
Фамилия И.О.

САНКТ-ПЕТЕРБУРГ
2021 г.

ЗАДАНИЕ

Общая постановка задачи.

Необходимо разработать приложение с графическим пользовательским интерфейсом согласно индивидуальному варианту. В ходе работы приложение должно хранить свои данные в файле формата XML или JSON и обращаться к ним по мере необходимости.

Варианты заданий (номер варианта 10).

Список студентов для людей с аллергией на БД. В файле хранятся студенты, поделённые на группы, поделённые по направлениям подготовки. Для каждого студента хранится его ФИО, и номер зачётки. На форме представлены три QListWidget и текстовое поле. Первый хранит список направлений подготовки. После выбора направления, во втором появляются группы, относящиеся к нему. После выбора группы, в третьем списке появляются ФИО студентов из этой группы. Выбор студента приводит к отображению в текстовом поле его номера зачётки. Также должна быть реализована возможность добавления нового студента, группы и направления подготовки.

ЭТАПЫ ВЫПОЛНЕНИЯ РАБОТЫ

Текст файла main.cpp:

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

#include "DEFINE.h"

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

    w.setFixedHeight(600);
    w.setFixedWidth(800);

    w.show();
    return a.exec();
}
```

Текст файла mainwindow.h:

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QListWidget>
#include <QLabel>
#include <QLineEdit>
#include <QPushButton>

#include <QJsonDocument>
#include <QJsonObject>
#include <QJsonParseError>
#include <QJsonArray>
#include <QFile>
#include <QFileDialog>

#include "DEFINE.h"

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow {
    Q_OBJECT

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

private:
    QJsonDocument doc;
    QJsonParseError docError;
    QString pathDoc;

    QFile file;

    QMenu * actionMenu;

    QAction * actionClearAll;
    QAction * actionOpenDoc;
    QAction * actionSaveDoc;
}
```

```

    QAction * actionExitProgram;

    QLabel * textHelp;
    QLabel * textDirection;
    QLabel * textGroup;
    QLabel * textStudent;
    QLabel * textNumber;
    QLabel * strNumber;

    QString number = "";

    QLineEdit * lineText;

    QPushButton * addDirection;
    QPushButton * addGroup;
    QPushButton * addStudent;
    QPushButton * addNumber;

    QListWidget * direction;
    QListWidget * group;
    QListWidget * student;

    QStringList listDirection;
    QStringList listGroup;
    QStringList listStudent;

    Ui::MainWindow * ui;

    int numberDirection = -1;
    int numberGroup = -1;
    int numberStudent = -1;

    void clearWidgetDirection();
    void clearWidgetGroup();
    void clearWidgetStudent();
    void clearWidgetNumber();

    void viewWidgetDirection();
    void viewWidgetGroup();
    void viewWidgetStudent();
    void viewWidgetNumber();

    void buttonOpenDoc();
    void buttonSaveDoc();
    void buttonClearDoc();

    void selectedDirection();
    void selectedGroup();
    void selectedStudent();

    void buttonAddDirection();
    void buttonAddGroup();
    void buttonAddStudent();
    void buttonAddNumberStudent();

    void clickDeleteDirection();
    void clickDeleteGroup();
    void clickDeleteStudent();
};

#endif // MAINWINDOW_H

```

Текст файла mainwindow.cpp:

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

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

    actionOpenDoc = new QAction(tr("Открыть файл"), this);
    connect(actionOpenDoc, &QAction::triggered, this,
&MainWindow::buttonOpenDoc);
    actionSaveDoc = new QAction(tr("Сохранить файл"), this);
    connect(actionSaveDoc, &QAction::triggered, this,
&MainWindow::buttonSaveDoc);
    actionClearAll = new QAction(tr("Очистить файл"), this);
    connect(actionClearAll, &QAction::triggered, this,
&MainWindow::buttonClearDoc);
    actionExitProgram = new QAction(tr("Выход"), this);
    connect(actionExitProgram, &QAction::triggered, this,
&MainWindow::close);

    actionMenu = menuBar()->addMenu(tr("Действие"));
    actionMenu->addAction(actionOpenDoc);
    actionMenu->addAction(actionSaveDoc);
    actionMenu->addAction(actionClearAll);
    actionMenu->addAction(actionExitProgram);

    lineText = new QLineEdit(this);
    lineText->setGeometry(QRect(QPoint(100, 30), QSize(500, 20)));

    textHelp = new QLabel(this);
    textHelp->setGeometry(QRect(QPoint(11, 30), QSize(80, 20)));
    textHelp->setText(tr("Введите текст:"));
    textDirection = new QLabel(this);
    textDirection->setGeometry(QRect(QPoint(240, 120), QSize(130, 20)));
    textDirection->setText(tr("Направления подготовки"));
    textGroup = new QLabel(this);
    textGroup->setGeometry(QRect(QPoint(660, 30), QSize(80, 20)));
    textGroup->setText(tr("Номер группы"));
    textStudent = new QLabel(this);
    textStudent->setGeometry(QRect(QPoint(260, 254), QSize(80, 20)));
    textStudent->setText(tr("ФИО студента"));
    textNumber = new QLabel(this);
    textNumber->setGeometry(QRect(QPoint(660, 550), QSize(80, 20)));
    textNumber->setText(tr("Номер зачетки: "));
    strNumber = new QLabel(this);
    strNumber->setGeometry(QRect(QPoint(680, 570), QSize(50, 20)));
    strNumber->setText(number);

    addDirection = new QPushButton(this);
    addDirection->setGeometry(QRect(QPoint(10, 60), QSize(290, 23)));
    addDirection->setText(tr("Добавить направление"));
    connect(addDirection, &QPushButton::clicked, this,
&MainWindow::buttonAddDirection);
    addGroup = new QPushButton(this);
    addGroup->setGeometry(QRect(QPoint(310, 60), QSize(290, 23)));
    addGroup->setText(tr("Добавить группу"));
    connect(addGroup, &QPushButton::clicked, this,
&MainWindow::buttonAddGroup);
    addStudent = new QPushButton(this);
    addStudent->setGeometry(QRect(QPoint(10, 90), QSize(290, 23)));
    addStudent->setText(tr("Добавить студента"));
```

```

        connect(addStudent, &QPushButton::clicked, this,
&MainWindow::buttonAddStudent);
        addNumber = new QPushButton(this);
        addNumber->setGeometry(QRect(QPoint(310, 90), QSize(290, 23)));
        addNumber->setText(tr("Указать номер зачетки"));
        connect(addNumber, &QPushButton::clicked, this,
&MainWindow::buttonAddNumberStudent);

        direction = new QListWidget(this);
        direction->setGeometry(QRect(QPoint(10, 150), QSize(590, 100)));
        connect(direction, &QListWidget::itemClicked, this,
&MainWindow::selectedDirection);
        connect(direction, &QListWidget::itemDoubleClicked, this,
&MainWindow::clickDeleteDirection);
        group = new QListWidget(this);
        group->setGeometry(QRect(QPoint(610, 60), QSize(180, 480)));
        connect(group, &QListWidget::itemClicked, this,
&MainWindow::selectedGroup);
        connect(group, &QListWidget::itemDoubleClicked, this,
&MainWindow::clickDeleteGroup);
        student = new QListWidget(this);
        student->setGeometry(QRect(QPoint(10, 280), QSize(590, 310)));
        connect(student, &QListWidget::itemClicked, this,
&MainWindow::selectedStudent);
        connect(student, &QListWidget::itemDoubleClicked, this,
&MainWindow::clickDeleteStudent);

        numberDirection = -1;
        numberGroup = -1;
        numberStudent = -1;

        clearWidgetDirection();
        clearWidgetGroup();
        clearWidgetStudent();
        clearWidgetNumber();
    }

MainWindow::~MainWindow() {
    delete ui;

    delete actionMenu;
    delete actionClearAll;
    delete actionOpenDoc;
    delete actionExitProgram;

    delete textHelp;
    delete textDirection;
    delete textGroup;
    delete textStudent;
    delete textNumber;
    delete strNumber;

    delete lineText;

    delete addDirection;
    delete addGroup;
    delete addStudent;
    delete addNumber;

    delete direction;
    delete group;
    delete student;
}

```

```

void MainWindow::buttonOpenDoc() {
    clearWidgetDirection();
    clearWidgetGroup();
    clearWidgetStudent();
    clearWidgetNumber();

    numberDirection = -1;
    numberGroup = -1;
    numberStudent = -1;

    pathDoc = QFileDialog::getOpenFileName(nullptr, "", "C:\\lab_QT\\LAB_2",
    "*.json");
    file.setFileName(pathDoc);

    if(file.open(QIODevice::ReadOnly | QFile::Text)) {
        doc = QJsonDocument::fromJson(QByteArray(file.readAll()), &docError);
    }

    file.close();

    viewWidgetDirection();

    strNumber->setText("Загружен");
}

void MainWindow::buttonSaveDoc() {
    if(pathDoc.isEmpty()) return;

    file.setFileName(pathDoc);
    file.open(QIODevice::WriteOnly | QFile::Text);
    file.write(doc.toJson());

    file.close();

    strNumber->setText("Сохранен");
}

void MainWindow::buttonClearDoc() {
    if(pathDoc.isEmpty()) return;

    file.setFileName(pathDoc);
    file.open(QIODevice::WriteOnly | QFile::Text);

    clearWidgetDirection();
    clearWidgetGroup();
    clearWidgetStudent();
    clearWidgetNumber();

    file.write(0);
    file.close();

    strNumber->setText("Очищен");
}

void MainWindow::clearWidgetDirection() {
    listDirection.clear();
    direction->clear();
}

void MainWindow::clearWidgetGroup() {
    listGroup.clear();
    group->clear();
}

```

```

}

void MainWindow::clearWidgetStudent() {
    listStudent.clear();
    student->clear();
}

void MainWindow::clearWidgetNumber() {
    number = "";
    strNumber->setText(number);
}

void MainWindow::viewWidgetDirection() {
    clearWidgetDirection();

    QJsonObject allDirection = doc.object();
    QJsonArray arrayDirection = allDirection.value("DIRECTION").toArray();

    QJsonObject temp;
    for(int i = 0; i < arrayDirection.count(); i++) {
        temp = arrayDirection.at(i).toObject();
        listDirection.append(temp.value("NAME_DIRECTION").toString());
    }

    direction->addItem(listDirection);
}

void MainWindow::viewWidgetGroup() {
    clearWidgetGroup();

    numberDirection = direction->currentRow();

    QJsonObject allDirection = doc.object();
    QJsonArray arrayDirection = allDirection.value("DIRECTION").toArray();

    QJsonObject allGroup = arrayDirection.at(numberDirection).toObject();
    QJsonArray arrayGroup = allGroup.value("LIST_GROUP").toArray();

    QJsonObject temp;
    for(int i = 0; i < arrayGroup.count(); i++) {
        temp = arrayGroup.at(i).toObject();
        listGroup.append(temp.value("NAME_GROUP").toString());
    }

    group->addItem(listGroup);
}

void MainWindow::viewWidgetStudent() {
    clearWidgetStudent();

    numberDirection = direction->currentRow();
    numberGroup = group->currentRow();

    QJsonObject allDirection = doc.object();
    QJsonArray arrayDirection = allDirection.value("DIRECTION").toArray();

    QJsonObject allGroup = arrayDirection.at(numberDirection).toObject();
    QJsonArray arrayGroup = allGroup.value("LIST_GROUP").toArray();

    QJsonObject allStudent = arrayGroup.at(numberGroup).toObject();
    QJsonArray arrayStudent = allStudent.value("LIST_STUDENT").toArray();

    QJsonObject temp;

```



```

        for(int i = 0; i < arrayStudent.count(); i++) {
            temp = arrayStudent.at(i).toObject();
            listStudent.append(temp.value("NAME_STUDENT").toString());
        }

        student->addItem(listStudent);
    }

void MainWindow::viewWidgetNumber() {
    clearWidgetNumber();

    numberDirection = direction->currentRow();
    numberGroup = group->currentRow();
    numberStudent = student->currentRow();

    QJsonObject allDirection = doc.object();
    QJsonArray arrayDirection = allDirection.value("DIRECTION").toArray();

    QJsonObject allGroup = arrayDirection.at(numberDirection).toObject();
    QJsonArray arrayGroup = allGroup.value("LIST_GROUP").toArray();

    QJsonObject allStudent = arrayGroup.at(numberGroup).toObject();
    QJsonArray arrayStudent = allStudent.value("LIST_STUDENT").toArray();

    QJsonObject temp = arrayStudent.at(numberStudent).toObject();

    number = temp.value("NUMBER_STUDENT").toString();

    strNumber->setText(number);
}

void MainWindow::selectedDirection() {
    clearWidgetGroup();
    clearWidgetStudent();
    clearWidgetNumber();

    numberGroup = -1;
    numberStudent = -1;

    viewWidgetGroup();
}

void MainWindow::selectedGroup() {
    clearWidgetStudent();
    clearWidgetNumber();

    numberStudent = -1;

    viewWidgetStudent();
}

void MainWindow::selectedStudent() {
    clearWidgetNumber();
    viewWidgetNumber();
}

void MainWindow::buttonAddDirection() {
    clearWidgetDirection();
    clearWidgetGroup();
    clearWidgetStudent();
    clearWidgetNumber();

    numberDirection = -1;

```

```

        numberGroup = -1;
        numberStudent = -1;

        QString tempStr = lineText->text();

        QJsonObject newDirection;
        newDirection.insert("NAME_DIRECTION", tempStr);
        newDirection.insert("LIST_GROUP", QJsonArray{});

        QJsonObject allDirection = doc.object();
        QJsonArray arrayDirection = allDirection.value("DIRECTION").toArray();
        arrayDirection.append(newDirection);

        QJsonObject All;
        All.insert("DIRECTION", arrayDirection);

        doc.setObject(All);

        viewWidgetDirection();
    }

void MainWindow::buttonAddGroup() {
    if(numberDirection == -1) {
        strNumber->setText(tr("Неудача"));
        return;
    }

    clearWidgetGroup();
    clearWidgetStudent();
    clearWidgetNumber();

    numberGroup = -1;
    numberStudent = -1;

    QString tempStr = lineText->text();

    QJsonObject newGroup;
    newGroup.insert("NAME_GROUP", tempStr);
    newGroup.insert("LIST_STUDENT", QJsonArray{});

    QJsonObject allDirection = doc.object();
    QJsonArray arrayDirection = allDirection.value("DIRECTION").toArray();

    QJsonObject allGroup = arrayDirection.at(numberDirection).toObject();
    QJsonArray arrayGroup = allGroup.value("LIST_GROUP").toArray();
    arrayGroup.append(newGroup);

    QJsonObject temp;
    temp.insert("NAME_DIRECTION",
arrayDirection.at(numberDirection).toObject().value("NAME_DIRECTION").toString());
    temp.insert("LIST_GROUP", arrayGroup);
    arrayDirection.replace(numberDirection, temp);

    QJsonObject All;
    All.insert("DIRECTION", arrayDirection);
    doc.setObject(All);

    viewWidgetGroup();
}

void MainWindow::buttonAddStudent() {
    if(numberDirection == -1 || numberGroup == -1) {

```

```

        strNumber->setText(tr("Неудача"));
        return;
    }

    clearWidgetStudent();
    clearWidgetNumber();

    numberStudent = -1;

    QString tempStr = lineText->text();

    QJsonObject newStudent;
    newStudent.insert("NAME_STUDENT", tempStr);
    newStudent.insert("NUMBER_STUDENT", "");

    QJsonObject allDirection = doc.object();
    QJsonArray arrayDirection = allDirection.value("DIRECTION").toArray();

    QJsonObject allGroup = arrayDirection.at(numberDirection).toObject();
    QJsonArray arrayGroup = allGroup.value("LIST_GROUP").toArray();

    QJsonObject allStudent = arrayGroup.at(numberGroup).toObject();
    QJsonArray arrayStudent = allStudent.value("LIST_STUDENT").toArray();

    arrayStudent.append(newStudent);

    QJsonObject temp1;
    temp1.insert("NAME_GROUP",
arrayGroup.at(numberGroup).toObject().value("NAME_GROUP").toString());
    temp1.insert("LIST_STUDENT", arrayStudent);
    arrayGroup.replace(numberGroup, temp1);

    QJsonObject temp2;
    temp2.insert("NAME_DIRECTION",
arrayDirection.at(numberDirection).toObject().value("NAME_DIRECTION").toString());
    temp2.insert("LIST_GROUP", arrayGroup);
    arrayDirection.replace(numberDirection, temp2);

    QJsonObject All;
    All.insert("DIRECTION", arrayDirection);
    doc.setObject(All);

    viewWidgetStudent();
}

void MainWindow::buttonAddNumberStudent() {
    if(numberDirection == -1 || numberGroup == -1 || numberStudent == -1) {
        strNumber->setText(tr("Неудача"));
        return;
    }

    clearWidgetNumber();

    QString tempStr = lineText->text();
    number = tempStr;

    QJsonObject allDirection = doc.object();
    QJsonArray arrayDirection = allDirection.value("DIRECTION").toArray();

    QJsonObject allGroup = arrayDirection.at(numberDirection).toObject();
    QJsonArray arrayGroup = allGroup.value("LIST_GROUP").toArray();

```

```

        QJsonObject allStudent = arrayGroup.at(numberGroup).toObject();
        QJsonArray arrayStudent = allStudent.value("LIST_STUDENT").toArray();

        QJsonObject newStudent;
        newStudent.insert("NAME_STUDENT",
arrayStudent.at(numberStudent).toObject().value("NAME_STUDENT").toString());
        newStudent.insert("NUMBER_STUDENT", tempStr);

        arrayStudent.replace(numberStudent, newStudent);

        QJsonObject temp1;
        temp1.insert("NAME_GROUP",
arrayGroup.at(numberGroup).toObject().value("NAME_GROUP").toString());
        temp1.insert("LIST_STUDENT", arrayStudent);
        arrayGroup.replace(numberGroup, temp1);

        QJsonObject temp2;
        temp2.insert("NAME_DIRECTION",
arrayDirection.at(numberDirection).toObject().value("NAME_DIRECTION").toStrin
g());
        temp2.insert("LIST_GROUP", arrayGroup);
        arrayDirection.replace(numberDirection, temp2);

        QJsonObject All;
        All.insert("DIRECTION", arrayDirection);
        doc.setObject(All);

        viewWidgetNumber();
    }

void MainWindow::clickDeleteDirection() {
    clearWidgetDirection();
    clearWidgetGroup();
    clearWidgetStudent();
    clearWidgetNumber();

    numberGroup = -1;
    numberStudent = -1;

    QJsonObject allDirection = doc.object();
    QJsonArray arrayDirection = allDirection.value("DIRECTION").toArray();
    arrayDirection.removeAt(numberDirection);

    QJsonObject All;
    All.insert("DIRECTION", arrayDirection);
    doc.setObject(All);

    viewWidgetDirection();
}

void MainWindow::clickDeleteGroup() {
    if(numberDirection == -1) {
        strNumber->setText(tr("Неудача"));
        return;
    }

    clearWidgetGroup();
    clearWidgetStudent();
    clearWidgetNumber();

    numberStudent = -1;

    QJsonObject allDirection = doc.object();

```

```

        QJsonArray arrayDirection = allDirection.value("DIRECTION").toArray();

        QJsonObject allGroup = arrayDirection.at(numberDirection).toObject();
        QJsonArray arrayGroup = allGroup.value("LIST_GROUP").toArray();
        arrayGroup.removeAt(numberGroup);

        QJsonObject temp;
        temp.insert("NAME_DIRECTION",
arrayDirection.at(numberDirection).toObject().value("NAME_DIRECTION").toString());
        temp.insert("LIST_GROUP", arrayGroup);
        arrayDirection.replace(numberDirection, temp);

        QJsonObject All;
        All.insert("DIRECTION", arrayDirection);
        doc.setObject(All);

        viewWidgetGroup();
    }

void MainWindow::clickDeleteStudent() {
    if(numberDirection == -1 || numberGroup == -1) {
        strNumber->setText(tr("Неудача"));
        return;
    }

    clearWidgetStudent();
    clearWidgetNumber();

    QJsonObject allDirection = doc.object();
    QJsonArray arrayDirection = allDirection.value("DIRECTION").toArray();

    QJsonObject allGroup = arrayDirection.at(numberDirection).toObject();
    QJsonArray arrayGroup = allGroup.value("LIST_GROUP").toArray();

    QJsonObject allStudent = arrayGroup.at(numberGroup).toObject();
    QJsonArray arrayStudent = allStudent.value("LIST_STUDENT").toArray();
    arrayStudent.removeAt(numberStudent);

    QJsonObject temp1;
    temp1.insert("NAME_GROUP",
arrayGroup.at(numberGroup).toObject().value("NAME_GROUP").toString());
    temp1.insert("LIST_STUDENT", arrayStudent);
    arrayGroup.replace(numberGroup, temp1);

    QJsonObject temp2;
    temp2.insert("NAME_DIRECTION",
arrayDirection.at(numberDirection).toObject().value("NAME_DIRECTION").toString());
    temp2.insert("LIST_GROUP", arrayGroup);
    arrayDirection.replace(numberDirection, temp2);

    QJsonObject All;
    All.insert("DIRECTION", arrayDirection);
    doc.setObject(All);

    viewWidgetStudent();
}

```

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ ПРОГРАММЫ

На рисунке 1 представлен интерфейс программы. Справа снизу можно просмотреть номер зачетки выбранного студента. Слева сверху есть кнопка “Действие”. На рисунке 2 представлены какие действия можно совершать. Чтобы программа начала свою работу, нужно загрузить в нее файл с помощью кнопки “Открыть файл”. Будем открывать файл с рисунка 3.

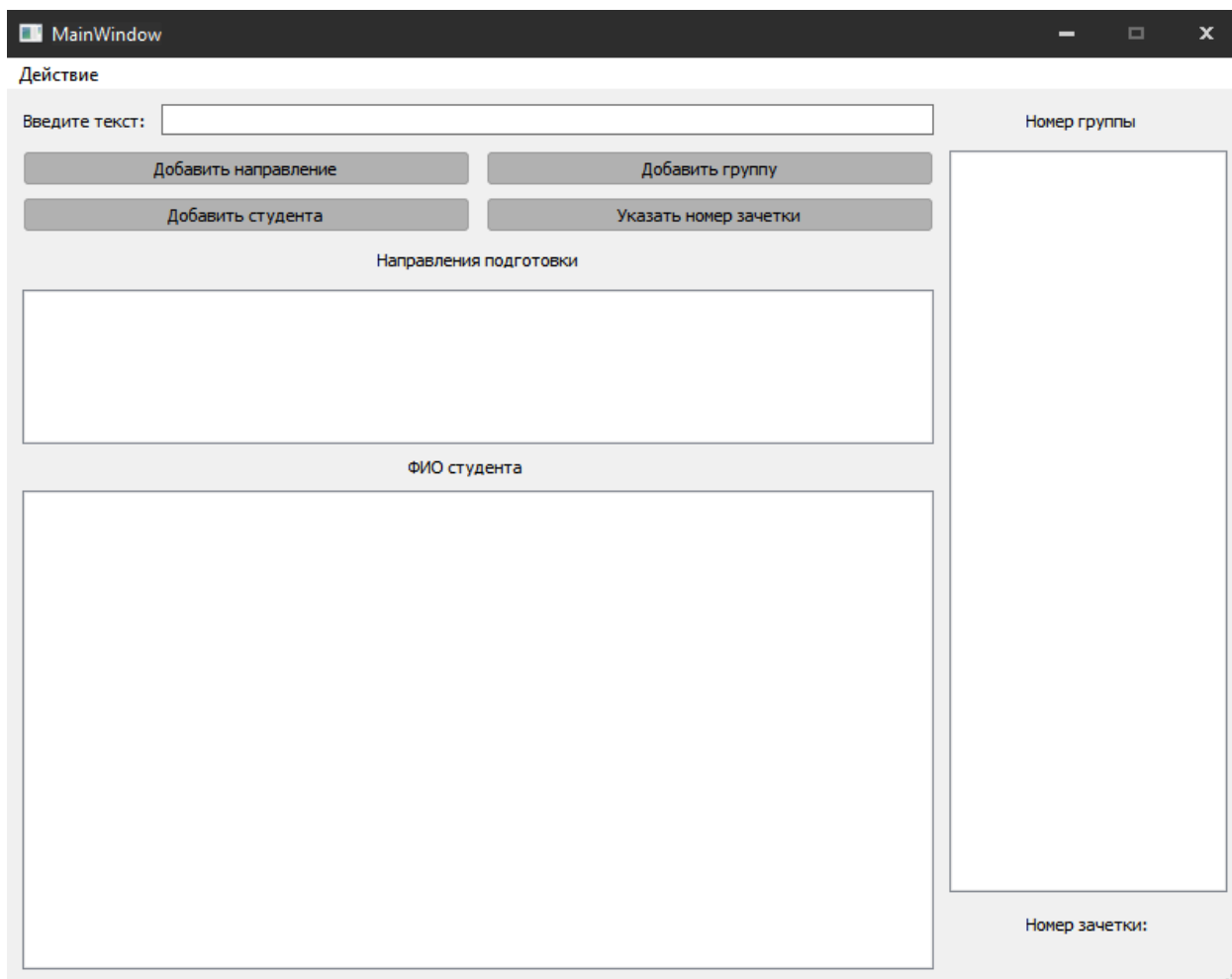


Рисунок 1 – интерфейс программы.

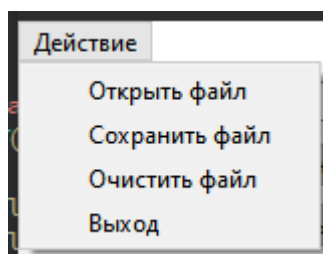


Рисунок 2 – меню “Действие”.

```
doc.json x
1 {
2   "DIRECTION": [
3     {
4       "LIST_GROUP": [
5         {
6           "LIST_STUDENT": [
7             {
8               "NAME_STUDENT": "Лебедев Никита Данилович",
9               "NUMBER_STUDENT": "I59120"
10            },
11            {
12              "NAME_STUDENT": "Козлов Андрей Бедросович",
13              "NUMBER_STUDENT": "I59138"
14            },
15            {
16              "NAME_STUDENT": "Грибов Вадим Валентинович",
17              "NUMBER_STUDENT": "I59125"
18            }
19          ],
20          "NAME_GROUP": "I591"
21        },
22        {
23          "LIST_STUDENT": [
24            "NAME_GROUP": "I592"
25          ],
26          "NAME_GROUP": "I592"
27        },
28        {
29          "LIST_STUDENT": [
30            "NAME_GROUP": "I593"
31          ],
32          "NAME_GROUP": "I593"
33        }
34      ],
35      "NAME_DIRECTION": "Информационные системы и технологии"
36    },
37    {
38      "LIST_GROUP": [
39        {
40          "LIST_STUDENT": [
41            "NAME_GROUP": "I594"
42          ],
43          "NAME_GROUP": "I594"
44        },
45        {
46          "LIST_STUDENT": [
47            "NAME_GROUP": "I595"
48          ],
49          "NAME_GROUP": "I595"
50        },
51        {
52          "LIST_STUDENT": [
53            "NAME_GROUP": "I596"
54          ],
55          "NAME_GROUP": "I596"
56        },
57        {
58          "LIST_STUDENT": [
59            "NAME_GROUP": "I597"
60          ],
61          "NAME_GROUP": "I597"
62        }
63      ],
64      "NAME_DIRECTION": "Программная инженерия"
65    }
66  ]
67 }
```

Рисунок 3 – примерная структура json файла, который будет открыт.

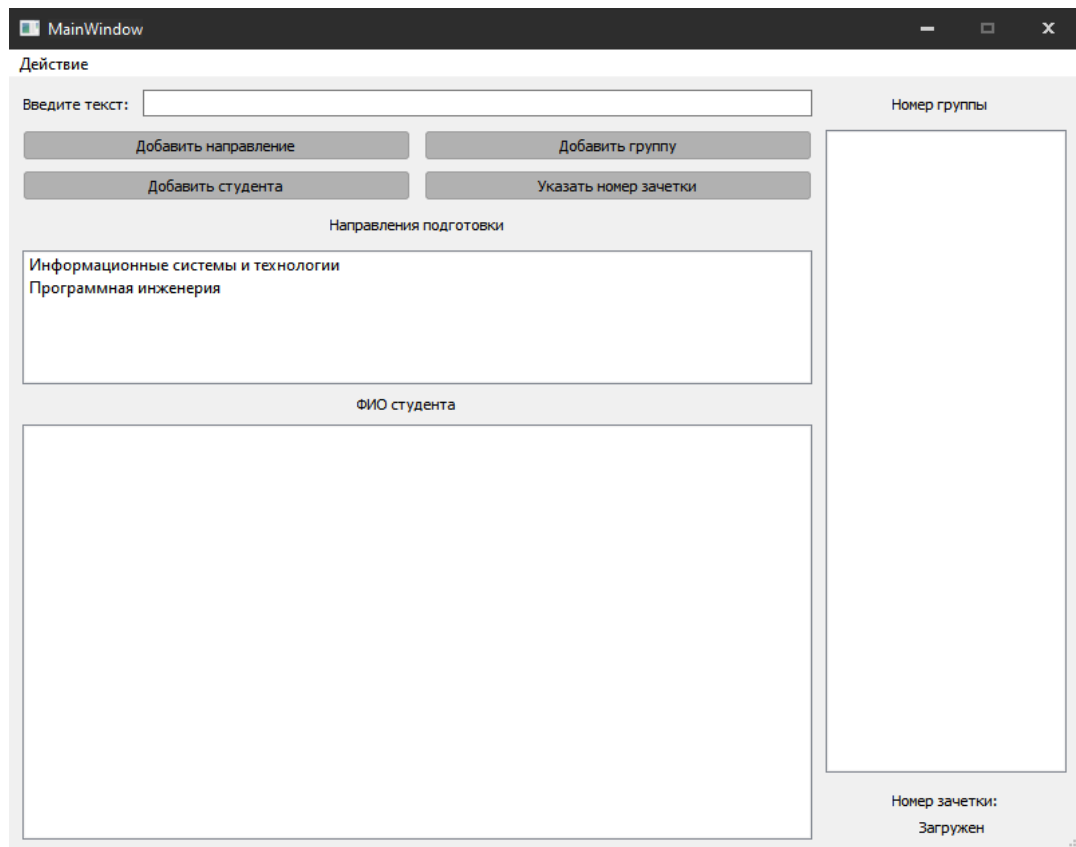


Рисунок 4 – открыли файл.

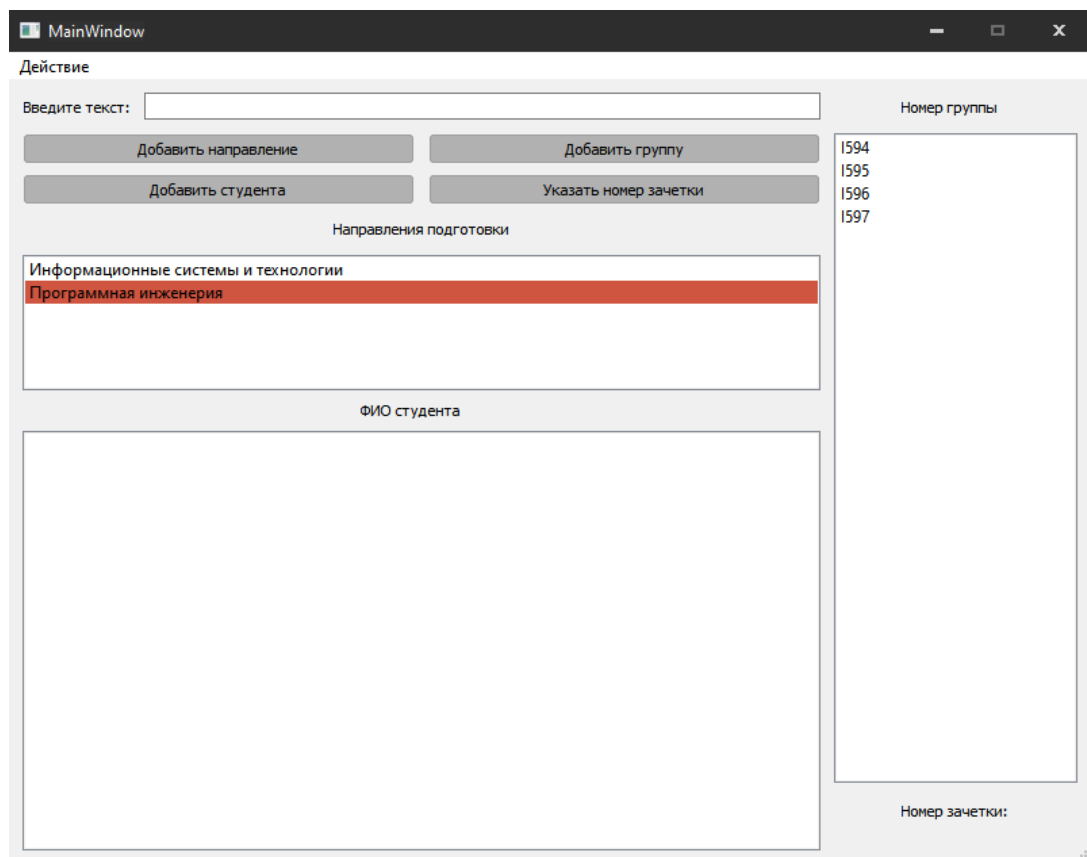


Рисунок 5 – выбрали направление.

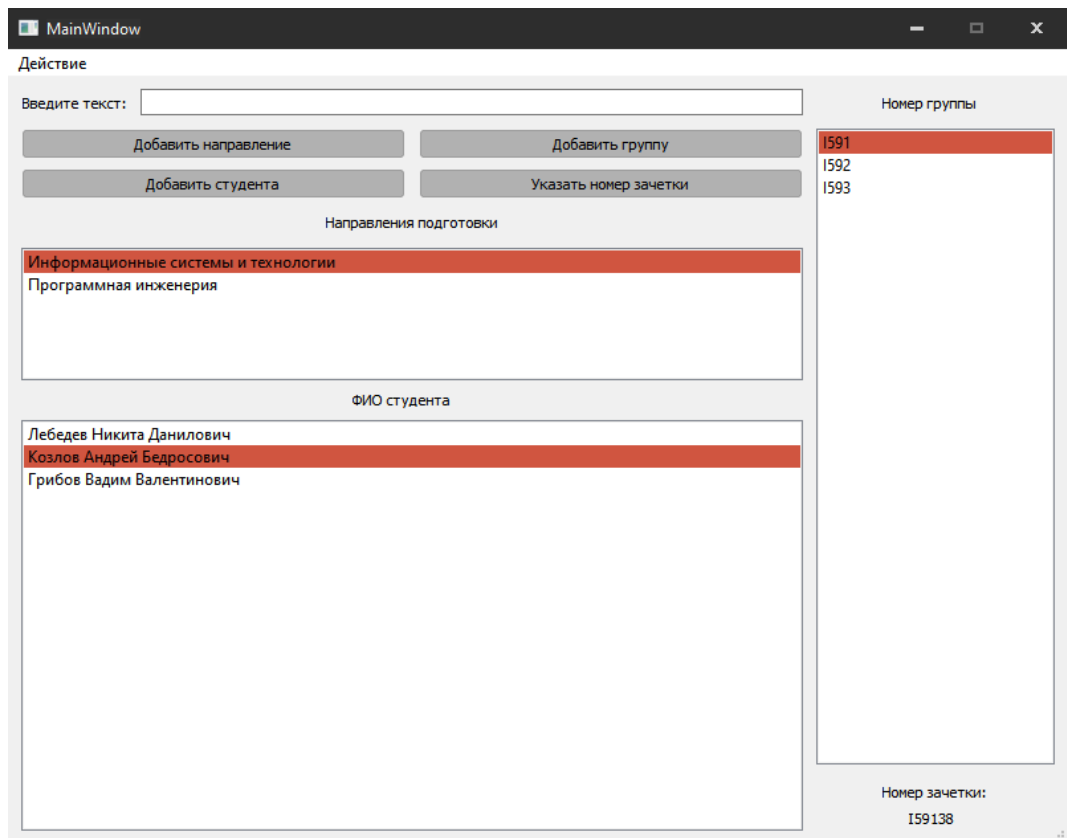


Рисунок 5 – выбрали группу и студента и получили его номер зачетки.

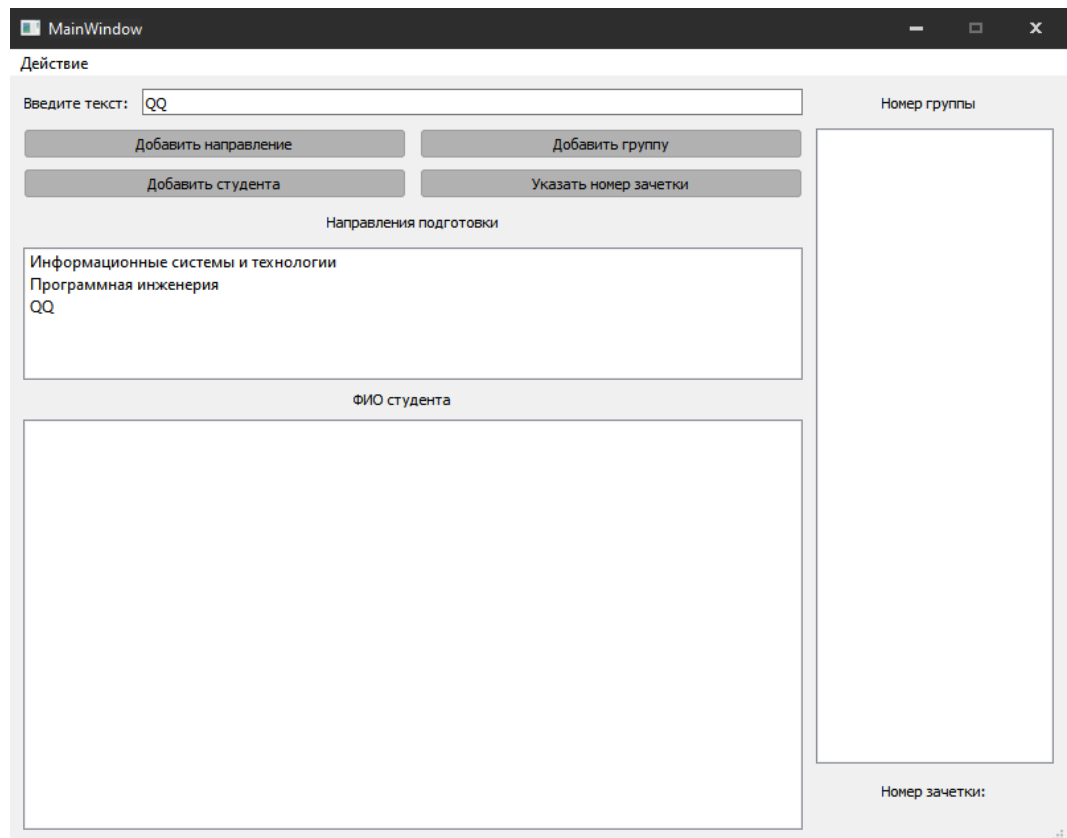


Рисунок 6 – добавили новое направление.

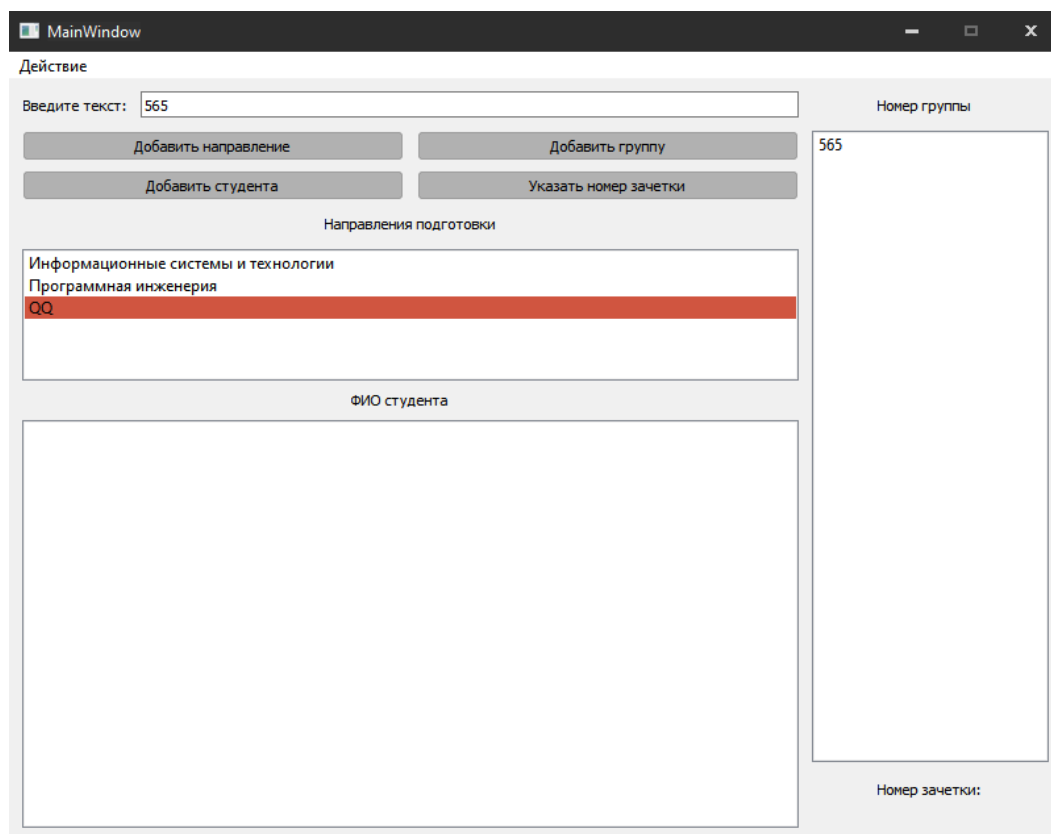


Рисунок 7 – добавили новую группу в направление “QQ”.

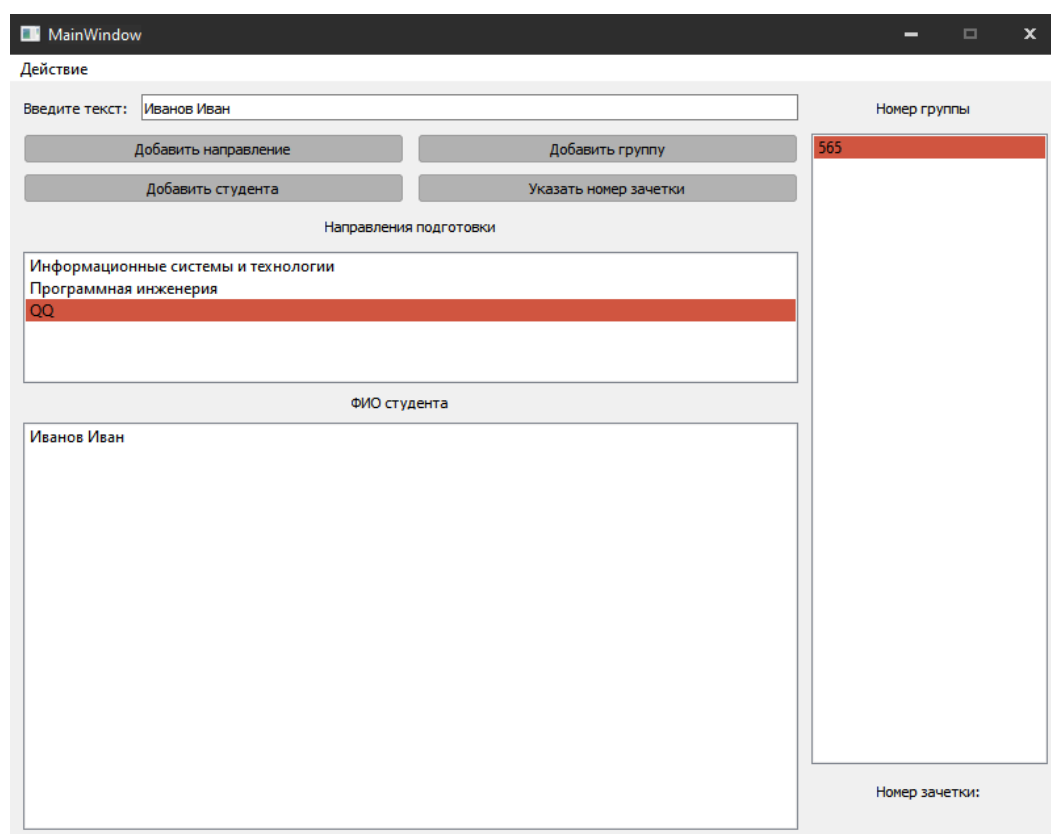


Рисунок 8 – добавили нового ученика в группу “565”.

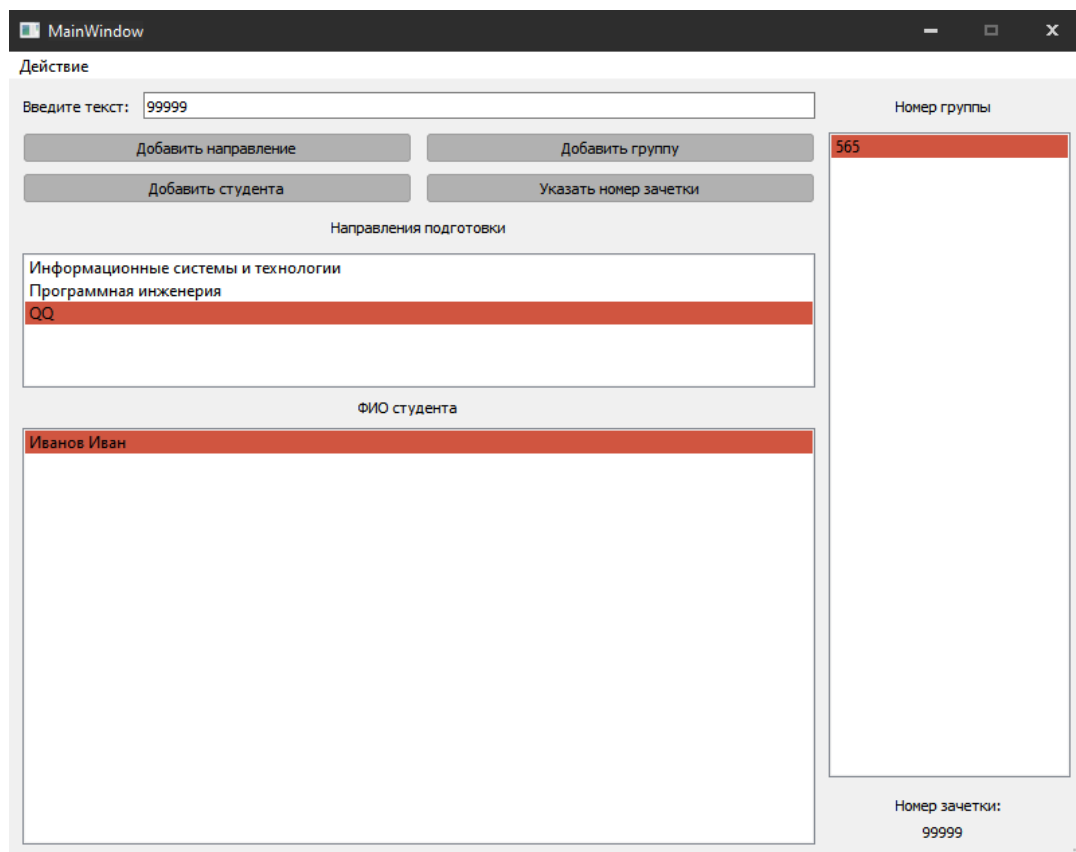


Рисунок 9 – указали номер зачетки ученику “Иванов Иван”.

По двойному щелчку по направлению, группе или студенту можно их удалить. Чтобы изменения сохранились, нужно нажать кнопку “Сохранить файл” в меню “Действие”.

ВЫВОД

Разобрались как работать в QT с QListWidget, QJsonDocument, QJsonObject и QJsonArray.

Написали программу соответствующую варианту задания.

Разобрались с синтаксисом файла Json.