

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
"САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М. А. БОНЧ-БРУЕВИЧА"

Факультет инфокоммуникационных сетей и систем  
Кафедра программной инженерии и вычислительной техники

**Курсовая работа**

по дисциплине «ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ  
ПРОГРАММИРОВАНИЕ»

Вариант №13

Выполнил:

студент 2 курса

группы ИКПИ-12

Музычук Артём Сергеевич

Принял:

Рожков Максим Александрович

---

## Оглавление

1. Постановка задачи .....	3
1.1 Требования к функциям базы данных .....	3
1.2 Требования к интерфейсу программы .....	3
2. Выбор и обоснование типа разрабатываемого контейнера.....	4
3. Разработка структуры программы .....	5
4. Разработка собственных классов.....	6
4.1 MainWindow – класс главного окна.....	6
4.2 Adding– класс для добавления или редактирования записей .....	6
4.3 CSV_View– класс для отображения содержимого в csv-формате.....	6
5. Разработка интерфейса .....	7
5.1. Основное окно программы .....	7
5.2 Диалоговое окно для добавления/редактирования записей .....	8
5.3 Диалоговое окно для отображения данных в csv-формате .....	8
6. Инструкция пользователю.....	9
6.1 Добавление записи в базу данных .....	9
6.2. Редактирование записи .....	10
6.3. Удаление записи.....	11
6.4. Очистка таблицы.....	12
6.5. Создание новой базы данных .....	13
6.6. Открыть, сохранить, сохранить как .....	14
6.7. Объединение баз данных .....	16
6.8. CSV-отображение данных .....	17
6.9. Поиск.....	17
6.10 Отображение информации о программе и закрытие программы. ....	18
7. Код программы.....	19
8. Выводы.....	41
9. Список литературы .....	42

## 1. Постановка задачи

Создать базу данных, работающую на основе контейнера, который следует разработать в процессе выполнения курсовой работы. В качестве прототипа контейнера можно использовать контейнеры `vector` или `list` из библиотеки STL. Для работы с контейнером следует предусмотреть итератор. Остальные требования к контейнеру определяются студентом самостоятельно.

Разрабатываемая база данных должна быть создана использованием C++ и библиотеки Qt. Среда программирования Qt Creator.

В качестве элементов контейнеров должны использоваться объекты классов, приведенные в таблице 1.

Таблица 1.

Вариант	Класс
13	Класс, характеризующий игрока футбольной команды

### 1.1 Требования к функциям базы данных

База данных должна обеспечивать выполнение следующих операций:

- создание базы данных,
- объединение баз данных,
- добавления и удаления записей,
- редактирования записей,
- просмотра базы данных,
- поиск данных (найти данные, удовлетворяющие заданному критерию),
- запись на диск,
- чтения с диска.

### 1.2 Требования к интерфейсу программы

Интерфейсная часть программы должна содержать следующие компоненты:

- окно «ABOUT»,
- главное меню,
- всплывающее меню,
- строку состояния,
- панель инструментов,
- подсказки по командам меню и панели инструментов.

## **2. Выбор и обоснование типа разрабатываемого контейнера**

В качестве прототипа разрабатываемого контейнера был выбран `vector`. Выбор данного контейнера обусловлен личным опытом разработки, наличием хорошей литературы и удобством использования.

Разрабатываемый контейнер должен обладать следующими функциями:

- Создание пустого контейнера,
- Создание контейнера размера `n`,
- Конструкторы копирования,
- Оператор копирования,
- Конструктор перемещения,
- Оператор перемещения,
- Получение элемента по индексу,
- Вставка элементов,
- Удаление элементов,
- Очистка контейнера
- Получение длины массива,
- Проверка на наличие элементов в массиве,
- Наличие класса итератора и итераторов `begin` и `end`.

### 3.Разработка структуры программы

Программа должна состоять из графического интерфейса и кода – инструмента взаимодействия с графическим интерфейсом.

#### Интерфейс:

*MainWindow* (основное окно программы) включает в себя:

- **QMenuBar** – главное меню программы;
- **QLineEdit** – строка для поиска по базе данных;
- **QTableWidget** – таблица из пяти колонок, представляющая собой базу данных;
- **QToolBar** – панель инструментов с кнопками;
- **QStatusBar** – строка статуса.
- Вертикальный компоновщик (**QVBoxLayout**).

*QDialog* (диалоговое окно для добавления/редактирования элементов) включает в себя:

- **QLabel** (x5) – поля для вывода текстовой информации (Команда, лига, позиция, фио, номер);
- **QLineEdit** (x5) – поля для ввода текста
- **QVBoxLayout** - вертикальный компоновщик;
- **QGridLayout** - компоновщик с сеткой.

*QDialog* (диалоговое окна просмотра содержимого базы данных в csv-формате) включает в себя:

- **QPushButton** – кнопка для закрытия окна;
- **QTextEdit** – многострочное поле для отображения текста.

#### Код:

- Точка входа (**main.cpp**);
- Главное окно программы (**mainwindow.cpp, mainwindow.h, mainwindow.ui**);
- Диалоговое окно редактирования (**adding.cpp, adding.h, adding.ui**);
- Отображение содержимого БД (**csv\_view.cpp, csv\_view.h, csv\_view.ui**);

## 4. Разработка собственных классов

### 4.1 MainWindow – класс главного окна

*Методы класса:*

<b>void</b>	<b>on_CreateButton_clicked</b>	открытие нового файла
<b>void</b>	<b>on_ExitButton_clicked</b>	выход из программы
<b>void</b>	<b>on_AddButton_clicked</b>	добавление записи
<b>void</b>	<b>on_EditButton_clicked</b>	изменение записи
<b>void</b>	<b>on_AboutButton_clicked</b>	информация о программе и авторе
<b>void</b>	<b>on_DeleteButton_clicked</b>	удаление записи
<b>void</b>	<b>on_ClearButton_clicked</b>	очистка таблицы
<b>void</b>	<b>on_InputButton_clicked</b>	чтение БД из файла
<b>void</b>	<b>on_Merge_clicked</b>	объединение БД
<b>void</b>	<b>on_SaveButton_clicked</b>	сохранение
<b>void</b>	<b>on_SaveAsButton_clicked</b>	сохранить как
<b>void</b>	<b>on_SearchBar_textChanged(const QString &amp;arg1)</b>	поиск в searchBar
<b>void</b>	<b>on_CSVButton_clicked</b>	отображение данных через «;» (csv-формат)

Таблица 1. Методы главного окна

### 4.2 Adding– класс для добавления или редактирования записей

*Методы класса:*

<b>void</b>	<b>on_buttonBox_accepted</b>	Принятие добавления или изменения
<b>void</b>	<b>setData(QList&lt;QTableWidgetItem*&gt; list)</b>	Изменение/добавление данных в таблице

Таблица 2. Методы класса добавления и редактирования

### 4.3 CSV\_View– класс для отображения содержимого в csv-формате

*Методы класса:*

<b>explicit</b>	<b>CSV_View(QWidget *parent = nullptr)</b>	Конструктор
<b>void</b>	<b>on_closeButton_clicked</b>	Заккрытие окна
<b>void</b>	<b>set_text(QString row)</b>	Вывод элементов таблицы в поле через “;”

Таблица 3. Методы класса CSV

# 5.Разработка интерфейса

## 5.1. Основное окно программы

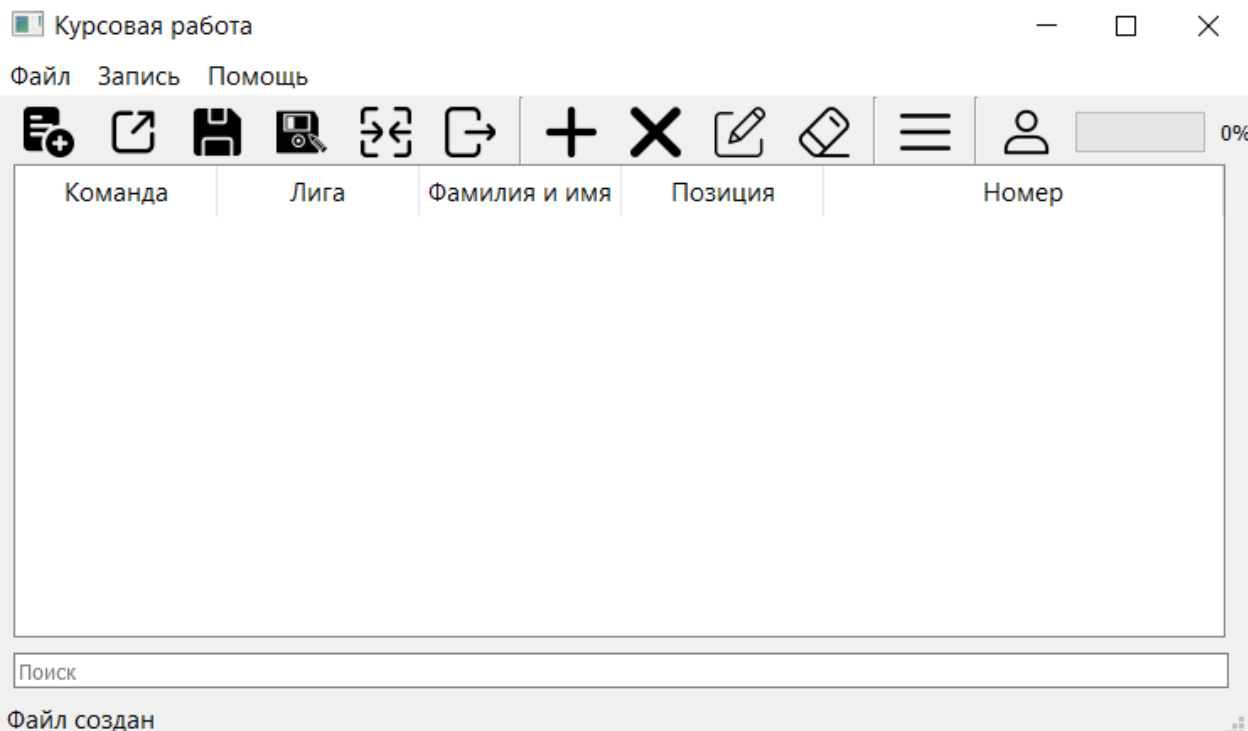


Рисунок 1. Основное окно программы (меню, панель инструментов с иконками, таблица, строка поиска, строка состояния, иконка программы)

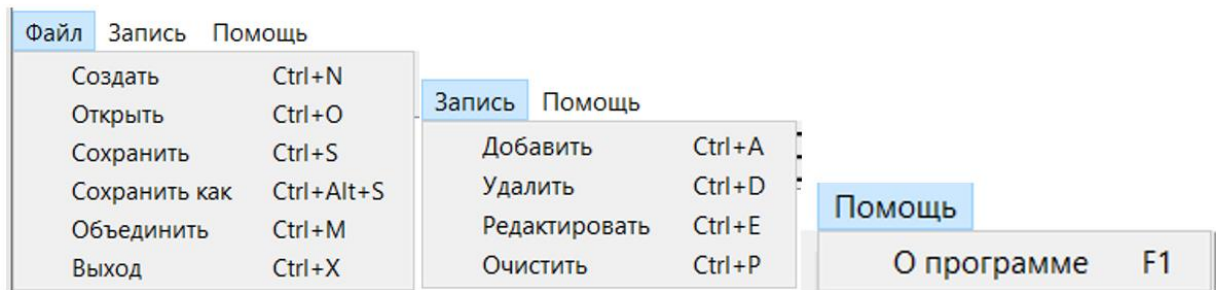


Рисунок 2. Выпадающее меню с подсказками в виде горячих клавиш.

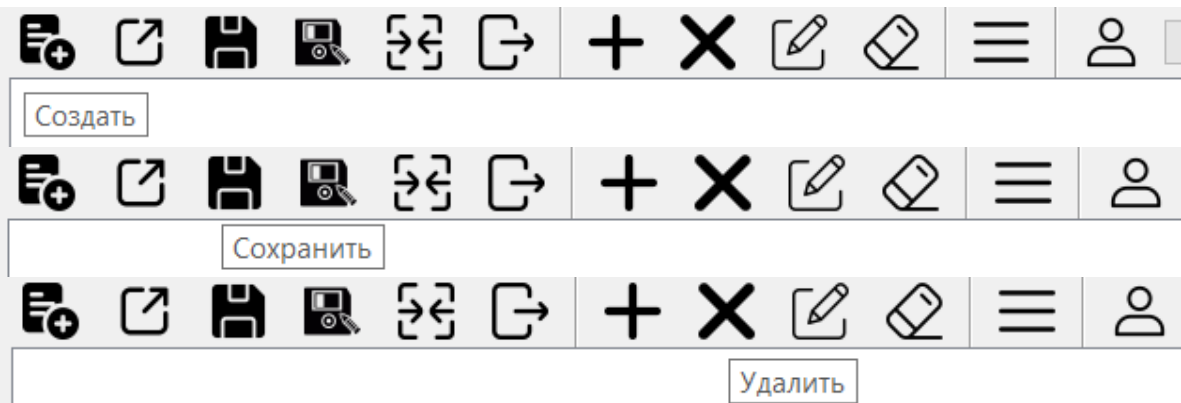


Рисунок 3. Панель инструментов с интуитивно понятными иконками и подсказками при наведении.

## 5.2 Диалоговое окно для добавления/редактирования записей

The image shows two side-by-side dialog boxes, both titled "Dialog". Each dialog has five input fields and two buttons at the bottom: "OK" and "Cancel".

Field	Left Dialog (Empty)	Right Dialog (Filled)
Команда		ЦСКА
Лига		РПЛ
Фамилия и имя		Музычук Артём
Позиция		вратарь
Номер	int	1

Рисунок 4. Пустое и заполненное диалоговые окна для добавления и редактирования записей (используется функция *simplified*, удаляющая ведущие и завершающие пробелы).

## 5.3 Диалоговое окно для отображения данных в csv-формате

The image shows two side-by-side dialog boxes, both titled "Dialog". Each dialog has a large text area and a "Закреть" (Close) button at the bottom.


Field	Left Dialog (Empty)	Right Dialog (Filled)
Text Area		ЦСКА;РПЛ;Музычук Артём;нападающий;12; Зенит;рпл2;Роналду Алина;нападающий;23; Реал;ЧМ;Месси Мбапе;вратарь;17; Анжи;Корсаковская;ЖанКлод Вандам Терминатор;смотрящий;1;

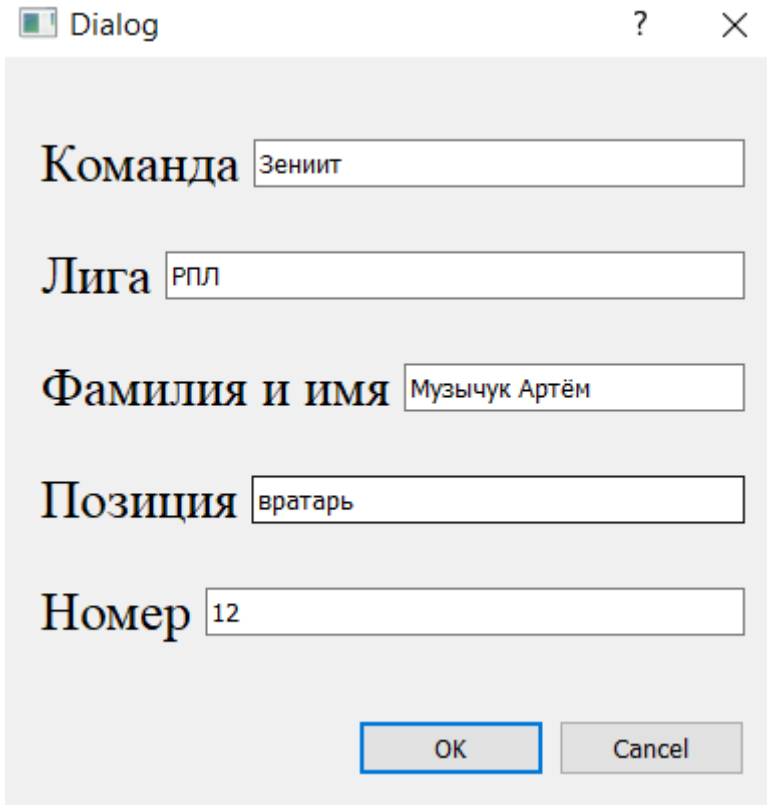
Рисунок 5. Пустое и заполненное окно с данными таблицы через «;», только для чтения.



## 6. Инструкция пользователю

### 6.1 Добавление записи в базу данных

Добавление записи осуществляется с помощью меню, сочетания **ctrl+a** или кнопки  .



Dialog

Команда

Лига

Фамилия и имя

Позиция

Номер

OK Cancel

Рисунок 6. Добавление записей через диалоговое окно.

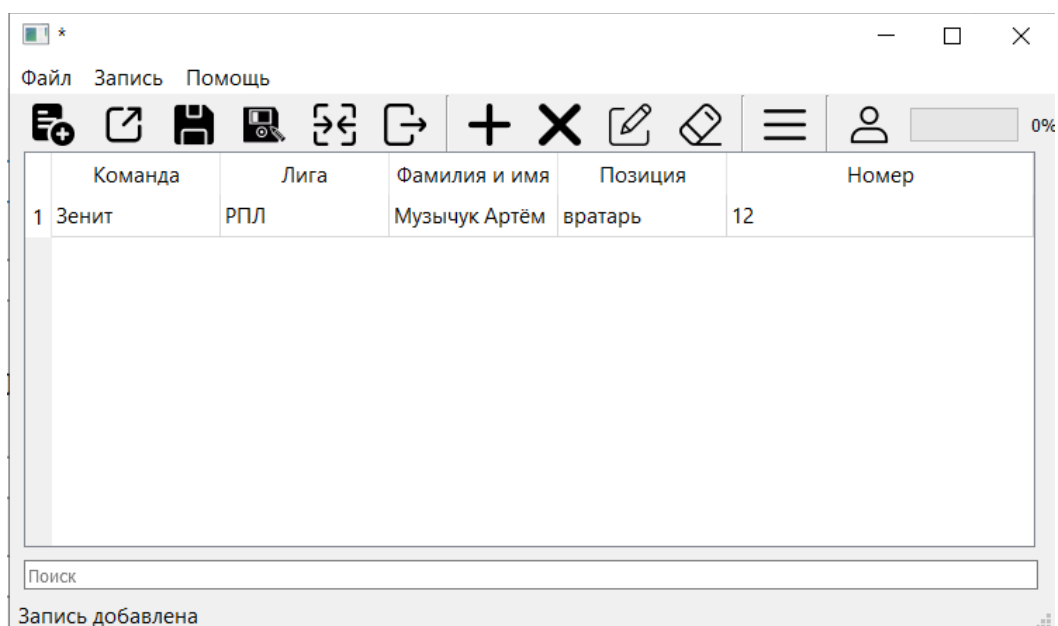



Рисунок 7. Результат добавления записей, отображение информации в статусной строке.

6.2. Редактирование записи

Редактирование записи осуществляется с помощью меню, двойного клика по строке, сочетания **ctrl+e** или кнопки .

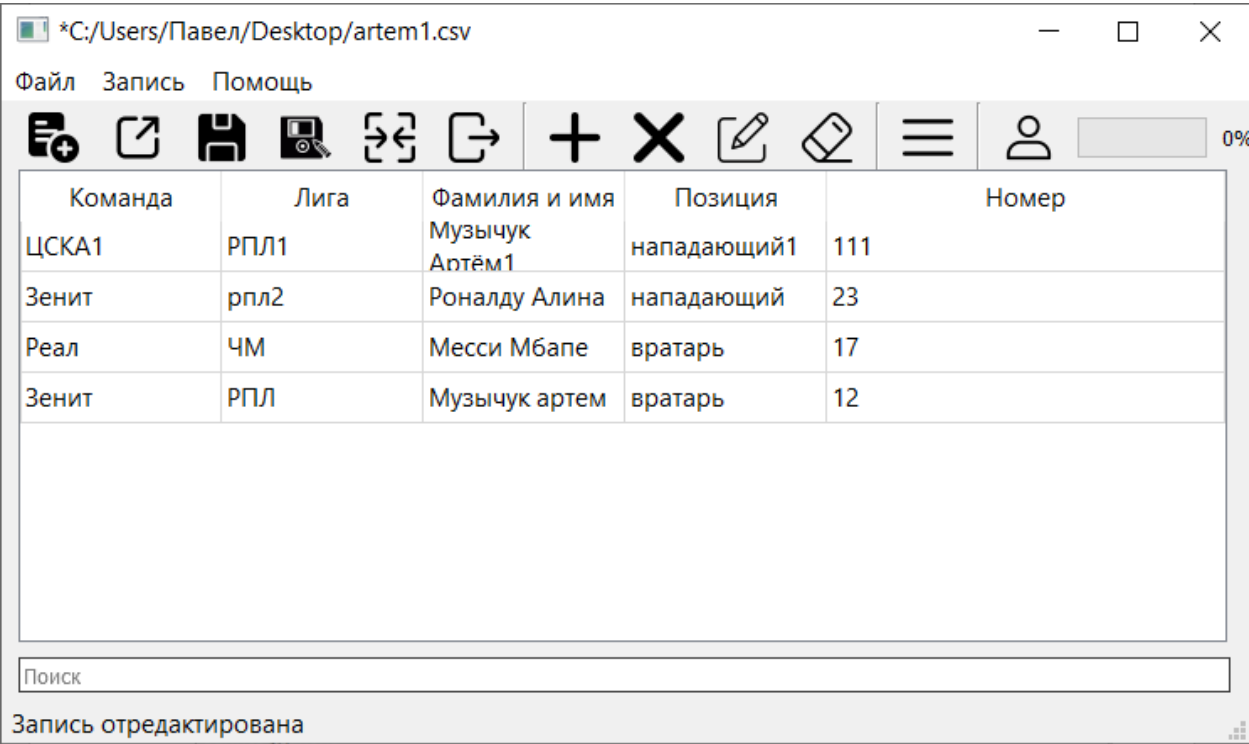
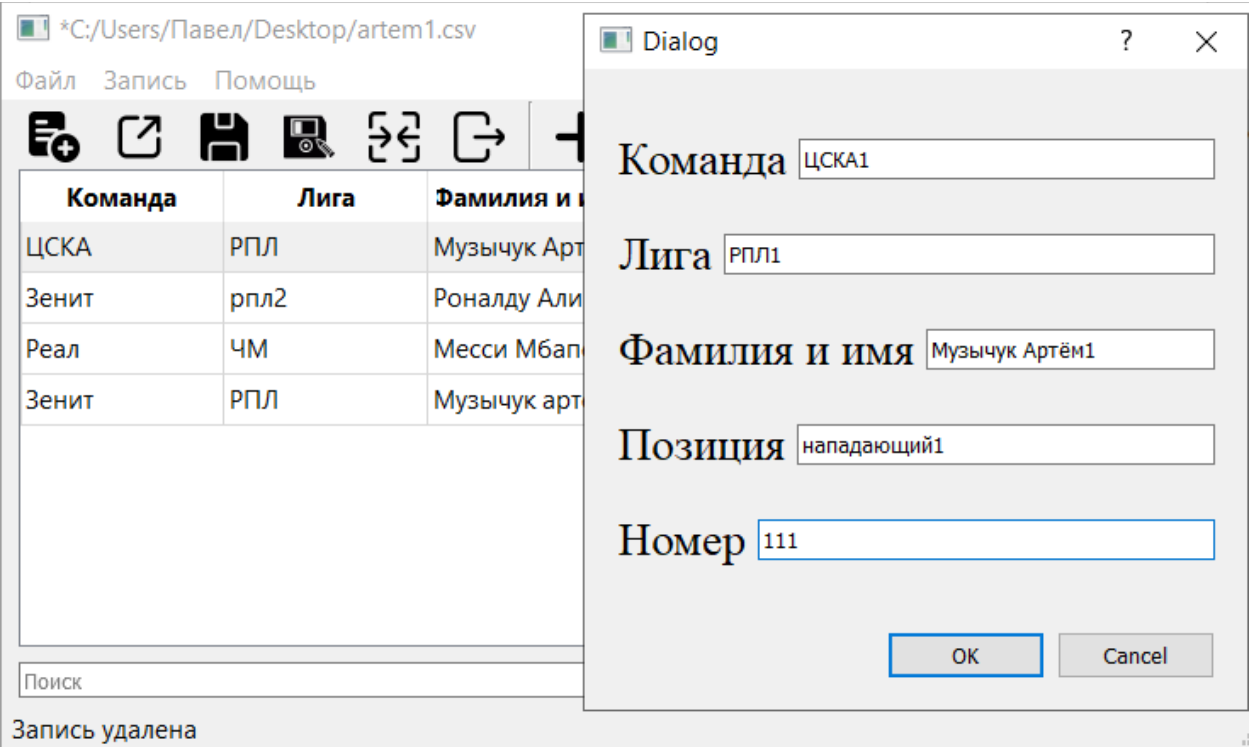


Рисунок 8. Редактирование номинала «СВЧ усилителя», сохранение в таблице, информация в статусной строке.

### 6.3. Удаление записи

Удаление записи осуществляется с помощью меню, сочетания **ctrl+d** или кнопки **X**.

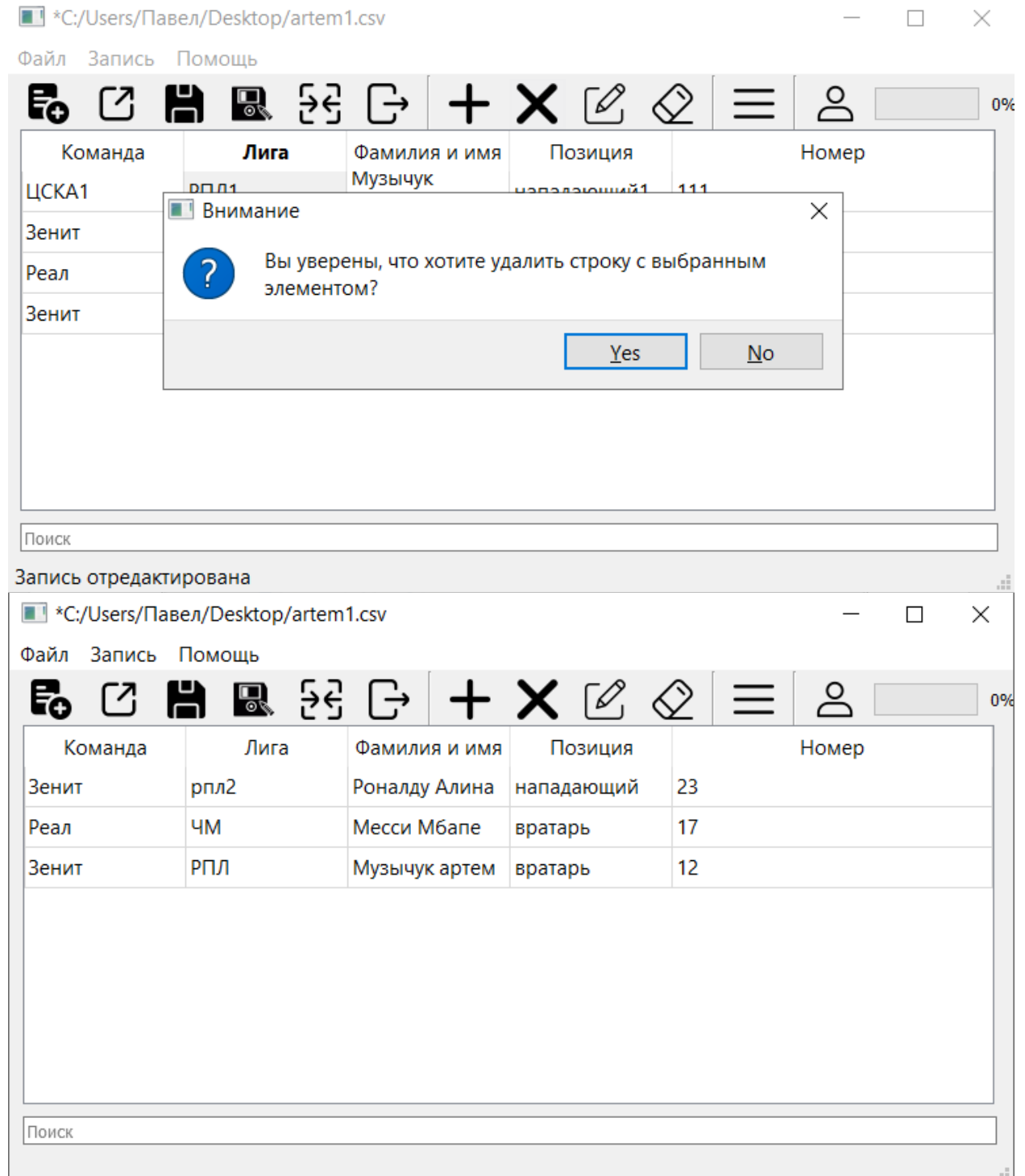


Рисунок 8. Удаление записи с подтверждением, вывод информации в статусную строку.

## 6.4. Очистка таблицы

Удаление всех осуществляется с помощью меню, сочетания **ctrl+p** или кнопки .

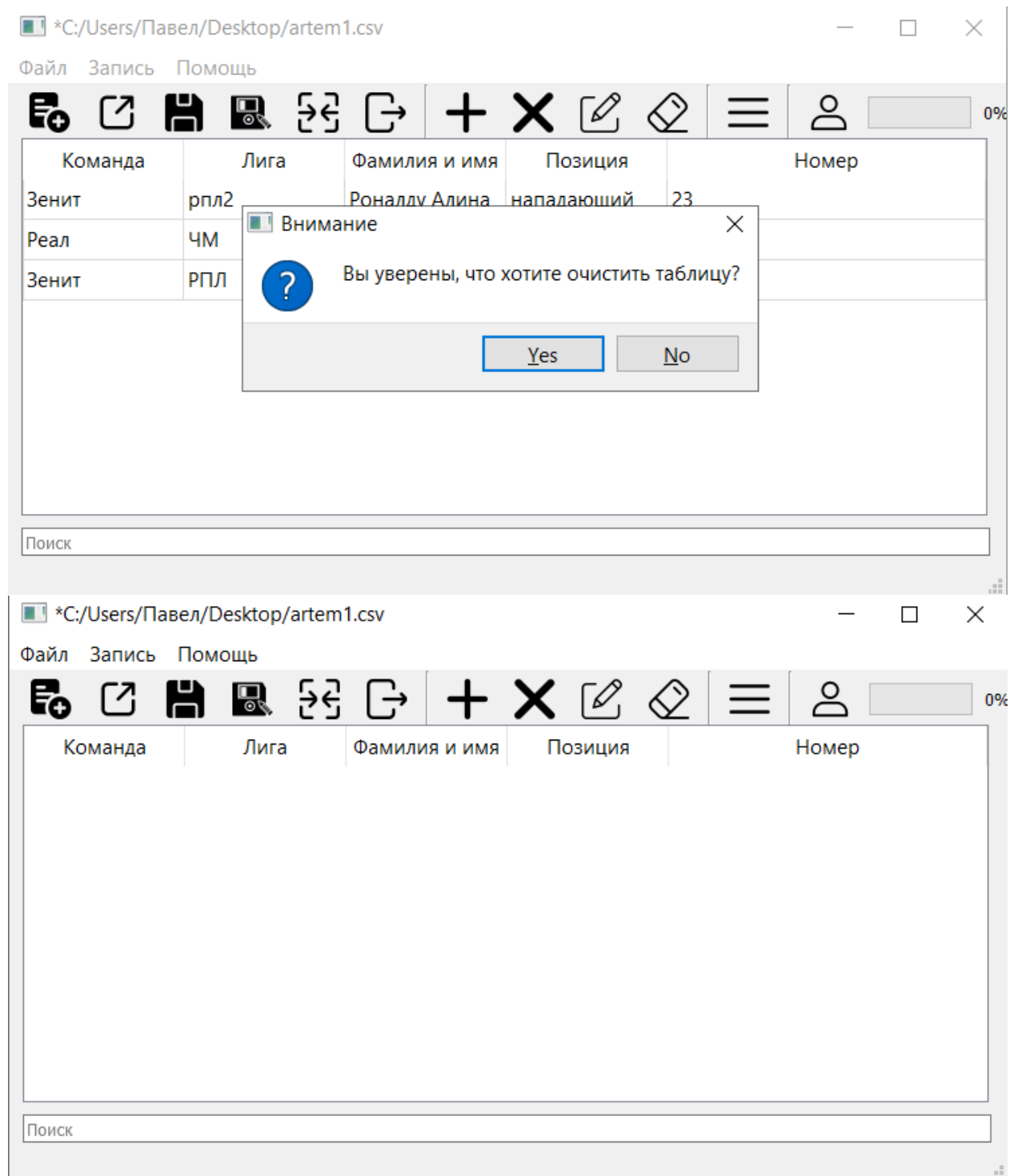

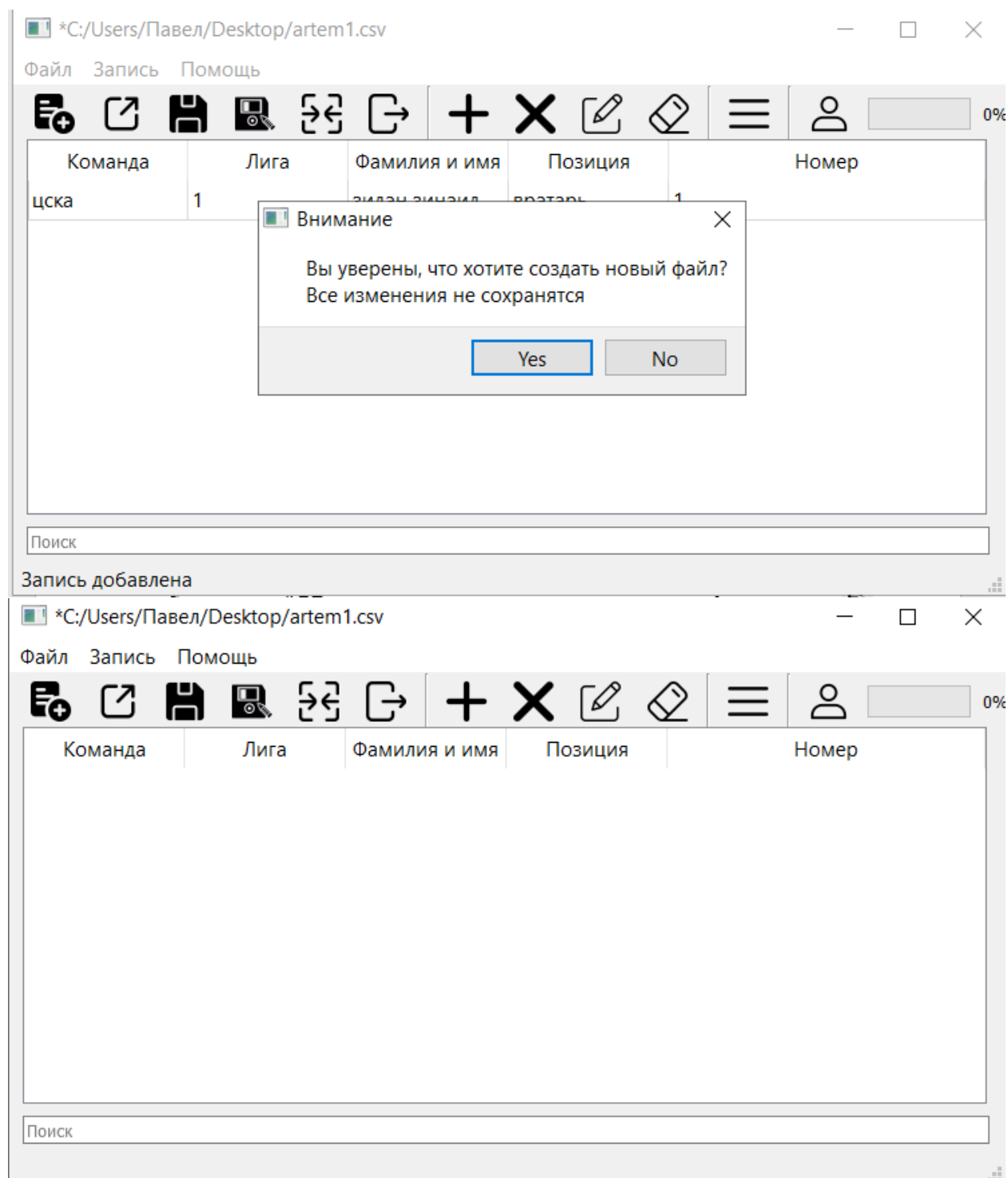





Рисунок 9. Очистка таблицы, информация в статусной строке.

## 6.5. Создание новой базы данных

Создание новой БД осуществляется с помощью меню, сочетания **ctrl+n** или кнопки .



## 6.6. Открыть, сохранить, сохранить как

Открытие уже существующей базы данных осуществляется с помощью меню, сочетания **ctrl+o** или кнопки , сохранение – сочетание **ctrl+s** или кнопка , сохранить как – сочетание **ctrl+alt+s** или кнопка .

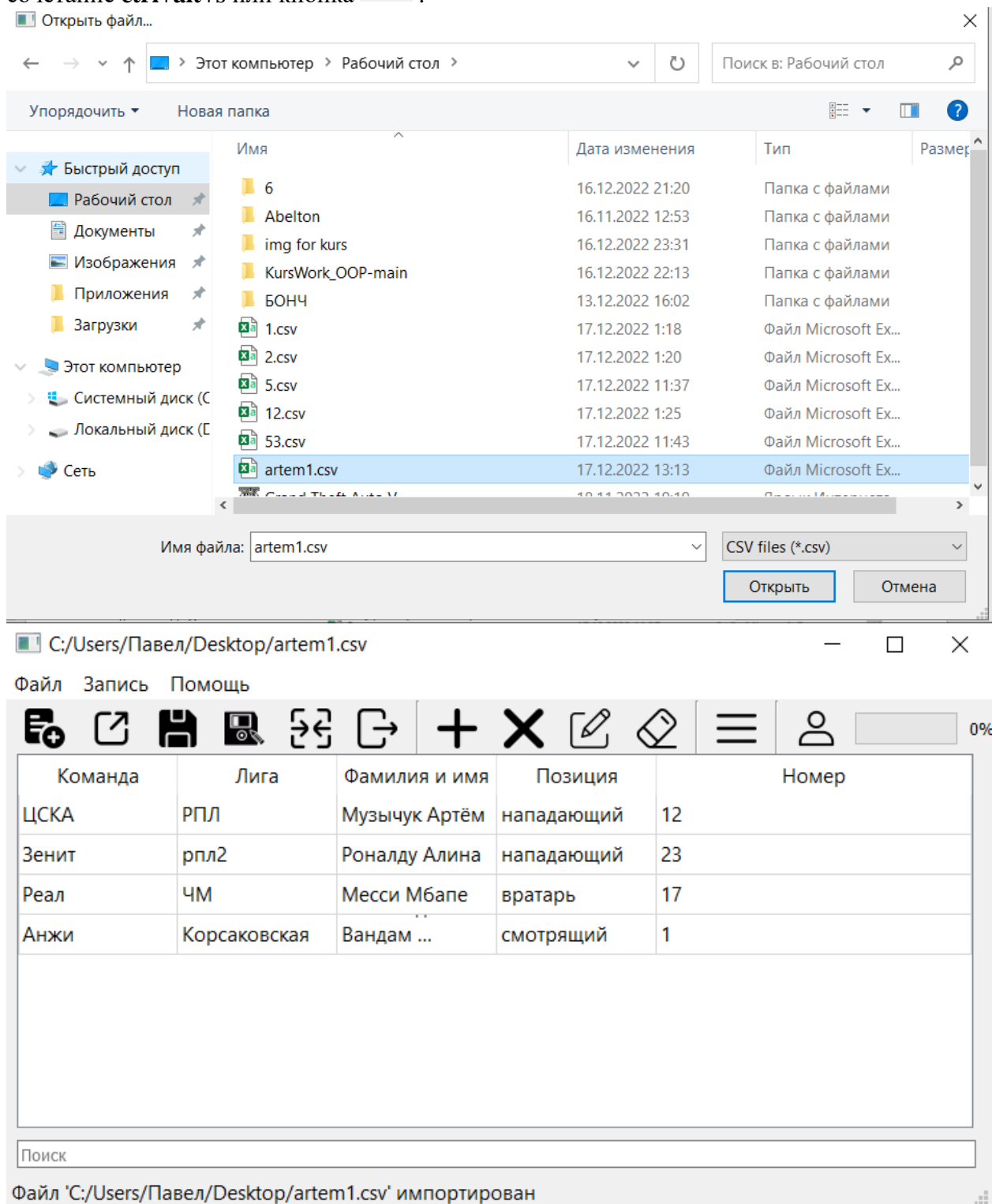


Рисунок 11. Открытие уже существующей базы данных, информация в статусной строке.

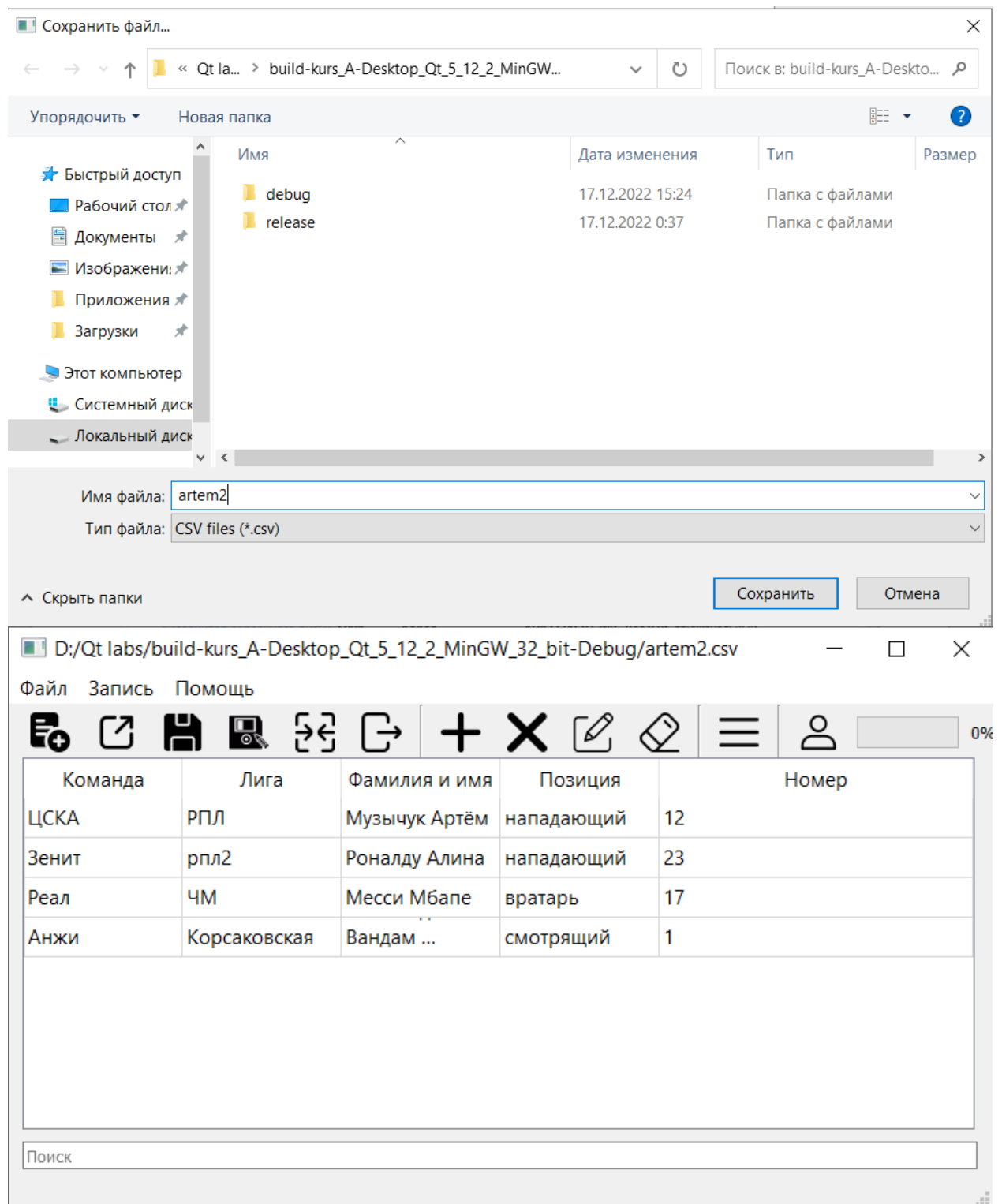


Рисунок 12. Сохранение базы данных, информация в статусной строке.

6.7. Объединение баз данных

Объединение БД осуществляется с помощью меню, сочетания **ctrl+m** или кнопки  .

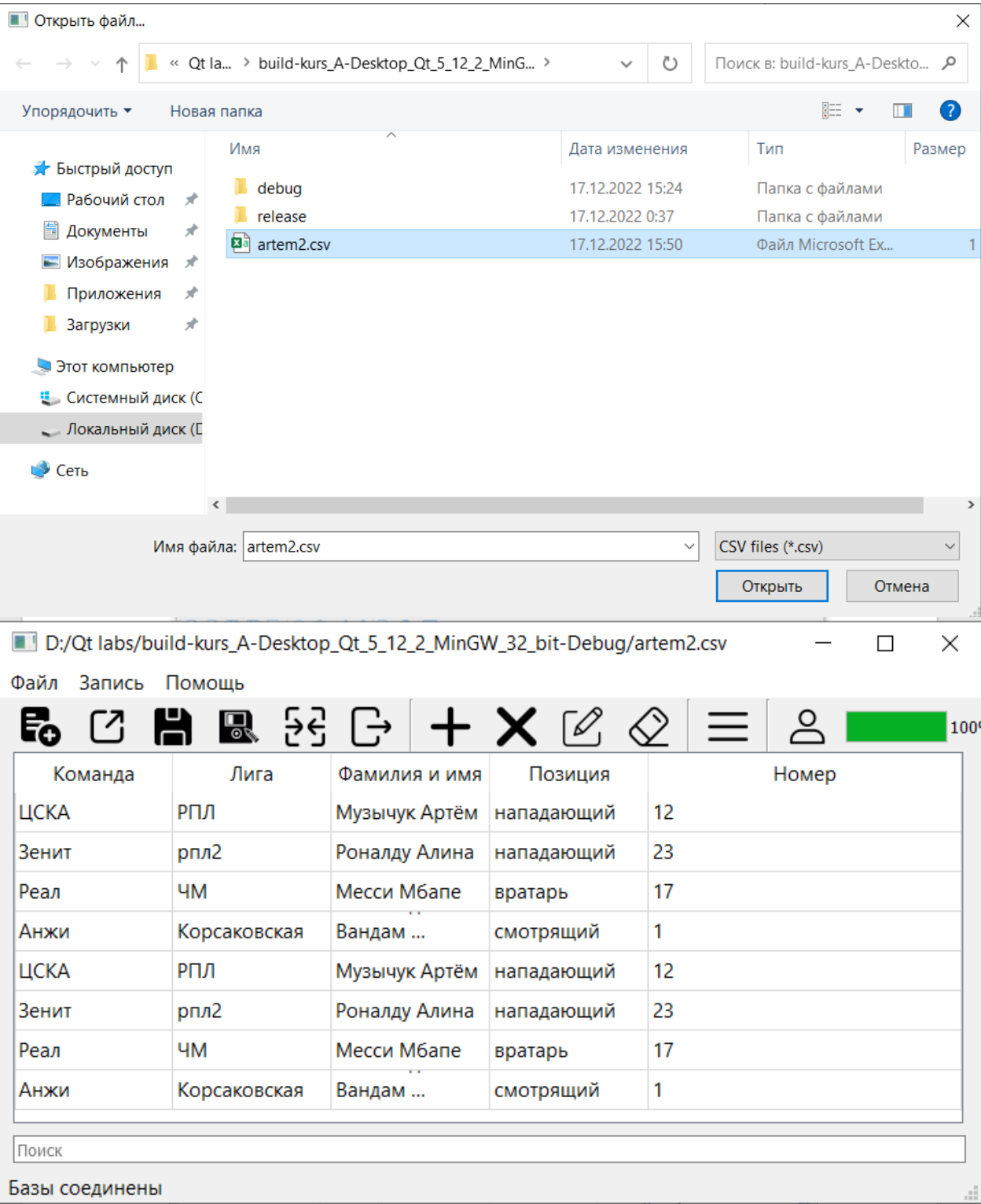


Рисунок 13. Объединение «база\_данных\_2.csv» с «база\_данных.csv», информация в статусной строке.



## 6.8. CSV-отображение данных

Просмотр данных в формате CSV осуществляется с помощью меню, сочетания **ctrl+`** или

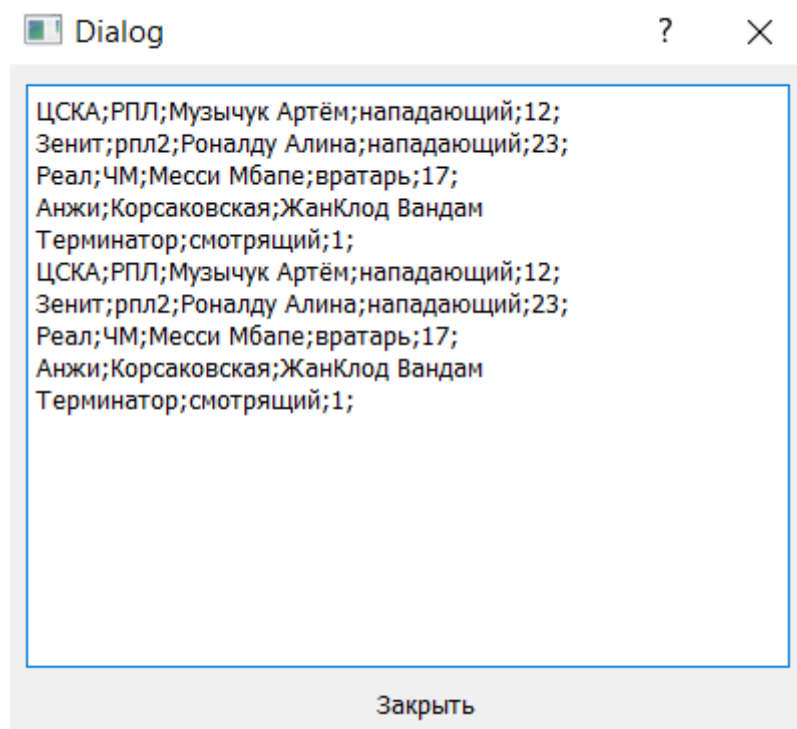


Рисунок 14. CSV-отображение.

## 6.9. Поиск

В строке поиска, находящееся над основным окном. Введите данные, которые хотите найти. В таблице автоматически выделяются совпадения, а внизу окна выведется какое количество ячеек совпадает с введенными данными:

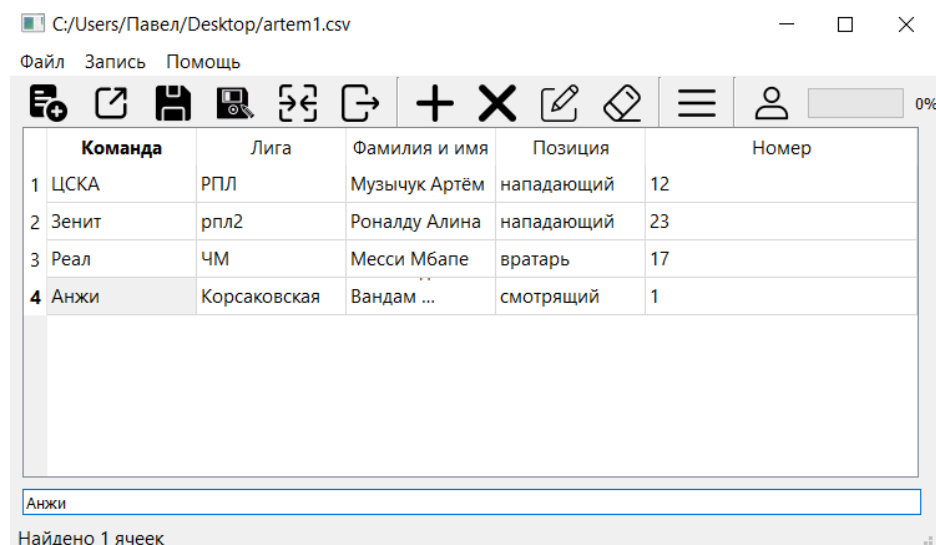



Рисунок 15. Поиск по базе данных, отображение информации в статусной строке

## 6.10 Отображение информации о программе и закрытие программы.

Просмотр информации о программе и об авторе осуществляется с помощью меню, панели инструментов, клавиши **A** или кнопки 

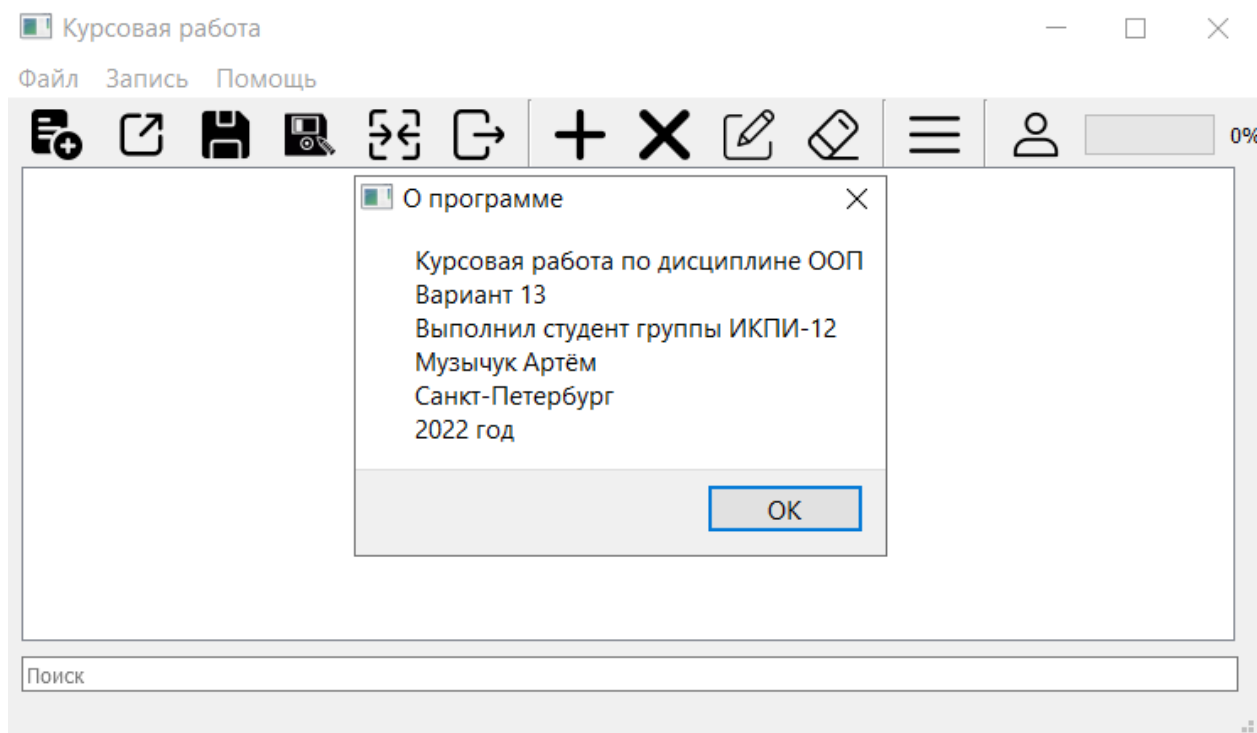



Рисунок 16. Просмотр информации о программе и авторе.

Заккрыть программу можно с помощью кнопки с правом верхнем углу, меню, панели инструментов, сочетания **ctrl+q** или кнопки .

## 7. Код программы

kurs_A.pro
<pre> QT      += core gui  greaterThan(QT_MAJOR_VERSION, 4): QT += widgets  CONFIG += c++11  # You can make your code fail to compile if it uses deprecated APIs. # In order to do so, uncomment the following line. #DEFINES += QT_DISABLE_DEPRECATED_BEFORE=0x060000   # disables all the APIs deprecated before Qt 6.0.0  SOURCES += \     adding.cpp \     csv_view.cpp \     main.cpp \     mainwindow.cpp  HEADERS += \     adding.h \     csv_view.h \     mainwindow.h  FORMS += \     adding.ui \     csv_view.ui \     mainwindow.ui  # Default rules for deployment. qnx: target.path = /tmp/\${TARGET}/bin else: unix:!android: target.path = /opt/\${TARGET}/bin !isEmpty(target.path): INSTALLS += target  RESOURCES += \     resource.qrc </pre>

adding.h
<pre> #ifndef ADDING_H #define ADDING_H  #include &lt;QDialog&gt; #include &lt;QTableWidgetItem&gt;  namespace Ui { class adding; }  class adding : public QDialog {     Q_OBJECT  public:     explicit adding(QWidget *parent = nullptr);      ~adding();     void setData(QList&lt;QTableWidgetItem *&gt; list/*QString creator, QString name, QString color, QString delivery_time, QString count*/); </pre>

```
private slots:
    void on_buttonBox_accepted();

private:
    Ui::adding *ui;
};

#endif // ADDING_H
```

### csv\_view.h

```
#ifndef CSV_VIEW_H
#define CSV_VIEW_H

#include <QDialog>

namespace Ui {
class CSV_View;
}

class CSV_View : public QDialog
{
    Q_OBJECT

public:
    explicit CSV_View(QWidget *parent = nullptr);
    ~CSV_View();
    void set_text(QString row);

private slots:
    void on_closeButton_clicked();

private:
    Ui::CSV_View *ui;
};

#endif // CSV_VIEW_H
```

### mainwindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <adding.h>
#include <csv_view.h>
QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT

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

```

void adding(QString creator, QString name, QString color, QString delivery_time, int count, int row = -1);
private slots:

void on_CreateButton_clicked();

void on_AddButton_clicked();
void on_AboutButton_clicked();

void on_EditButton_clicked();

void on_DeleteButton_clicked();

void on_ClearButton_clicked();

void on_InputButton_clicked();

void on_Merge_clicked();

void on_SaveButton_clicked();

void on_SaveAsButton_clicked();

void on_ExitButton_clicked();

void on_SearchBar_textChanged(const QString &arg1);

void on_CSVButton_clicked();

void on_ExitButtonMenu_triggered();

void on_AboutButtonMenu_triggered();

void on_AddButtonMenu_triggered();

void on_DeleteButtonMenu_triggered();

void on_EditButtonMenu_triggered();

void on_ClearButtonMenu_triggered();

void on_CreateButtonMenu_triggered();

void on_OpenButtonMenu_triggered();

void on_SaveButtonMenu_triggered();

void on_SaveAsButtonMenu_triggered();

void on_MergeButtonMenu_triggered();

private:
    Ui::MainWindow *ui;
    adding *add;
    CSV_View *csv;
    QString file_ = "";

};
#endif // MAINWINDOW_H

```

**adding.cpp**

```

#include "adding.h"
#include "ui_adding.h"
#include "mainwindow.h"

#include <QTableWidgetItem>

adding::adding(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::adding)
{
    ui->setupUi(this);
}

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

void adding::on_buttonBox_accepted()
{
    QString name = (ui->name->text());
    QString Creator(ui->creator->text());
    int Count(ui->count->text().toInt());
    QString DeliveryTime(ui->delivery_time->text());
    QString Color(ui->color->text());
    MainWindow *main_window = (MainWindow *) parentWidget();
    main_window->adding(Creator, name, Color, DeliveryTime, Count);
}

void adding::setData(QList<QTableWidgetItem *> list){
    ui->creator->setText(list.at(0)->text());
    ui->name->setText(list.at(1)->text());
    ui->color->setText(list.at(2)->text());
    ui->delivery_time->setText(list.at(3)->text());
    ui->count->setText(list.at(4)->text());
}

```

**csv\_view.cpp**

```

#include "csv_view.h"
#include "ui_csv_view.h"

CSV_View::CSV_View(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::CSV_View)
{
    ui->setupUi(this);
}

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

```

```
void CSV_View::set_text(QString row){
    ui->textEdit->setText(row);
}
void CSV_View::on_closeButton_clicked()
{
    close();
}
```

### 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 "QFileInfo"
#include "QFileDialog"
#include "QTextStream"
#include <QMessageBox>
#include <QTableWidget>
#include <adding.h>
MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setUpUi(this);
    ui->CreateButton->setToolTip("Создать");
    ui->InputButton->setToolTip("Открыть");
    ui->SaveButton->setToolTip("Сохранить");
    ui->SaveAsButton->setToolTip("Сохранить как");
    ui->Merge->setToolTip("Объединить");
    ui->ExitButton->setToolTip("Выйти");
    ui->AddButton->setToolTip("Добавить");
    ui->DeleteButton->setToolTip("Удалить");
    ui->EditButton->setToolTip("Редактировать");
    ui->ClearButton->setToolTip("Очистить");
    ui->CSVButton->setToolTip("Формат");
    ui->AboutButton->setToolTip("Разработчик");
}

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

void MainWindow::on_CreateButton_clicked()
```

```

{
    QMessageBox question;
    question.setWindowTitle("Внимание");
    question.setText("Вы уверены, что хотите создать новый файл?\nВсе изменения не сохранятся");
    question.addButton(QMessageBox::Yes);
    question.addButton(QMessageBox::No);
    if(question.exec() == QMessageBox::Yes){
        ui->tableWidget->setRowCount(0);
        ui->tableWidget->setColumnCount(5);
        ui->tableWidget->setHorizontalHeaderLabels(QStringList() << "Команда" << "Лига" << "Фамилия и
имя" << "Позиция" << "Номер");
        file_ = "";
        ui->statusbar->showMessage("Файл создан");
    }
}

void MainWindow::on_AddButton_clicked()
{
    add = new adding(this);
    if(add->exec() == QDialog::Accepted){
        setWindowTitle("*" + file_);
    }
}

void MainWindow::adding(QString creator, QString name, QString color, QString delivery_time, int count, int
row){
    int Count = count;
    auto list = ui->tableWidget->selectedItems();
    if (list.size() > 0){row = list.at(0)->row();}
    QTableWidgetItem *itm = new QTableWidgetItem(creator);
    QTableWidgetItem *itm1 = new QTableWidgetItem(name);
    QTableWidgetItem *itm2 = new QTableWidgetItem(color);
    QTableWidgetItem *itm3 = new QTableWidgetItem(delivery_time);
    QTableWidgetItem *itm4 = new QTableWidgetItem(QString::number(Count));
    if(row == -1){
        ui->tableWidget->setRowCount(ui->tableWidget->rowCount() + 1);
        ui->tableWidget->setItem(ui->tableWidget->rowCount()-1, 0, itm);
        ui->tableWidget->setItem(ui->tableWidget->rowCount()-1, 1, itm1);
        ui->tableWidget->setItem(ui->tableWidget->rowCount()-1, 2, itm2);
        ui->tableWidget->setItem(ui->tableWidget->rowCount()-1, 3, itm3);
        ui->tableWidget->setItem(ui->tableWidget->rowCount()-1, 4, itm4);
        ui->statusbar->showMessage("Запись добавлена");
    } else{
        ui->tableWidget->setItem(row, 0, itm);
        ui->tableWidget->setItem(row, 1, itm1);
        ui->tableWidget->setItem(row, 2, itm2);
        ui->tableWidget->setItem(row, 3, itm3);
        ui->tableWidget->setItem(row, 4, itm4);}
}

void MainWindow::on_AboutButton_clicked()
{
    QMessageBox::about(this,
        "О программе",
        "Курсовая работа по дисциплине ООП\n"
        "Вариант 13\n"
        "Выполнил студент группы ИКПИ-12\n"
        "Музычук Артём\n"

```



```

        "Санкт-Петербург\n"
        "2022 год");
    }

void MainWindow::on_EditButton_clicked()
{
    auto list = ui->tableWidget->selectedItems();
    if (list.size() > 0) {
        add = new adding(this);
        ui->tableWidget->selectRow(list.at(0)->row());
        auto list = ui->tableWidget->selectedItems();
        add->setData(list);
        add->exec();
        ui->tableWidget->clearSelection();
        ui->statusbar->showMessage("Запись отредактирована");
        setWindowTitle("*" + file_);
    }
}

void MainWindow::on_DeleteButton_clicked()
{
    QMessageBox msgBox;
    msgBox.setWindowTitle("Внимание");
    msgBox.setText("Вы уверены, что хотите удалить строку с выбранным элементом?");
    msgBox.setIcon(QMessageBox::Question);
    msgBox.setStandardButtons(QMessageBox::Yes | QMessageBox::No);
    msgBox.setDefaultButton(QMessageBox::Yes);
    if (msgBox.exec() == QMessageBox::Yes) {
        auto list = ui->tableWidget->selectedItems();
        if (list.size() > 0){
            for(int i = 0; i<list.length(); i++){
                ui->tableWidget->removeRow(list.at(i)->row());
            }
        }
        ui->statusbar->showMessage("Запись удалена");
    }
}

void MainWindow::on_ClearButton_clicked()
{
    if(ui->tableWidget->rowCount()>0){
        QMessageBox msgBox;
        msgBox.setWindowTitle("Внимание");
        msgBox.setText("Вы уверены, что хотите очистить таблицу?");
        msgBox.setIcon(QMessageBox::Question);
        msgBox.setStandardButtons(QMessageBox::Yes | QMessageBox::No);
        msgBox.setDefaultButton(QMessageBox::Yes);
        if (msgBox.exec() == QMessageBox::Yes) {
            ui->tableWidget->setRowCount(0);
        }
        ui->statusbar->showMessage("Таблица очищена");
        setWindowTitle("*" + file_);
    }
}

void MainWindow::on_InputButton_clicked()
{
    QString file_name = QFileDialog::getOpenFileName(this, "Открыть файл...", ".", "CSV files (*.csv);;All files (*.*)");
}

```

```

QFile file(file_name);
file_ = file_name;
if (file.open(QIODevice::ReadOnly)) {
    ui->tableWidget->setRowCount(0);
    ui->tableWidget->setColumnCount(5);
    // for(int i = 0; i<5; i++){
    // ui->tableWidget->setColumnWidth(i, 140);}
    ui->tableWidget->setHorizontalHeaderLabels(QStringList() << "Команда"<<"Лига"<<"Фамилия и
имя"<<"Позиция"<<"Номер");
    QTextStream in(&file);
    while (!in.atEnd()) {
        QString line = in.readLine();
        QStringList fields = line.split(";");
        adding(fields.at(0).simplified(), fields.at(1).simplified(), fields.at(2).simplified(),
fields.at(3).simplified(), fields.at(4).simplified().toInt());
    }
    file.close();
    setWindowTitle(file_name);
    ui->statusbar->showMessage("Файл '" + file_name + "' импортирован");
} else ui->statusbar->showMessage("Файл '" + file_name + "' не удалось открыть на чтение (" +
file.errorString() + ")");
}

void MainWindow::on_Merge_clicked()
{
    QString file_name = QFileDialog::getOpenFileName(this, "Открыть файл...", ".", "CSV files (*.csv);;All files
(*.*);;");
    QFile file(file_name);
    if (file.open(QIODevice::ReadOnly)) {
        if(ui->tableWidget->rowCount() == 0){
            ui->tableWidget->setColumnCount(5);
            ui->tableWidget->setHorizontalHeaderLabels(QStringList() << "Команда"<<"Лига"<<"Фамилия и
имя"<<"Позиция"<<"Номер");
        }
        QTextStream in(&file);
        while (!in.atEnd()) {
            QString line = in.readLine();
            QStringList fields = line.split(";");
            adding(fields.at(0).simplified(), fields.at(1).simplified(), fields.at(2).simplified(), fields.at(3).simplified(),
fields.at(4).simplified().toInt());

        }
        file.close();
        ui->statusbar->showMessage("Базы соединены");
        ui->progressBar->setValue(100);
    }
}

void MainWindow::on_SaveButton_clicked()
{
    if (file_ == "") {
        MainWindow::on_SaveAsButton_clicked();
        return;
    }
    QFile file(file_);
    if (file.open(QIODevice::WriteOnly)) {
        QTextStream out(&file);
        int len = ui->tableWidget->rowCount();

```

```

        for (int i = 0; i < len; ++i) {
            QString creator = ui->tableWidget->item(i, 0)->text();
            QString name = ui->tableWidget->item(i, 1)->text();
            QString color = ui->tableWidget->item(i, 2)->text();
            QString delivery_time = ui->tableWidget->item(i, 3)->text();
            QString count = ui->tableWidget->item(i, 4)->text();
            out << creator << ";" << name << ";" << color << ";" << delivery_time << ";" << count << ";" << "\n";
        }
        ui->statusbar->showMessage("Сохранено");
        file.close();
    }
    setWindowTitle(file_);
}

void MainWindow::on_SaveAsButton_clicked()
{
    QString file_name = QFileDialog::getSaveFileName(this, "Сохранить файл...", ".", "CSV files (*.csv);;All files (*.*);;");
    if (file_name != "") {
        file_ = file_name;
        MainWindow::on_SaveButton_clicked();
        setWindowTitle(file_);
        ui->statusbar->showMessage("Сохранено");
    }
}

void MainWindow::on_ExitButton_clicked()
{
    if (ui->tableWidget->rowCount() > 0 || file_ != "") {
        QMessageBox msgBox;
        msgBox.setWindowTitle("Внимание");
        msgBox.setText("Вы уверены, что хотите выйти?\nВсе сделанные изменения не будут сохранены.");
        msgBox.setIcon(QMessageBox::Question);
        msgBox.setStandardButtons(QMessageBox::Yes | QMessageBox::No);
        msgBox.setDefaultButton(QMessageBox::Yes);
        if (msgBox.exec() == QMessageBox::No) {
            return;
        }
    }
    exit(0);
}

void MainWindow::on_SearchBar_textChanged(const QString &arg1)
{
    ui->tableWidget->setCurrentCell(-1, -1);
    if (arg1 == "") {
        return;
    }
    ui->tableWidget->setSelectionMode(QAbstractItemView::SelectionMode::MultiSelection);
    auto find_items = ui->tableWidget->findItems(arg1, Qt::MatchContains);
    int len = find_items.size();
    for (int i = 0; i < len; ++i) {
        auto item = find_items.at(i);
        ui->tableWidget->setItemSelected(item, true);
    }
    ui->tableWidget->setSelectionMode(QAbstractItemView::SelectionMode::SingleSelection);
    ui->statusbar->showMessage("Найдено " + QString::number(len) + " ячеек");
}

```

```
void MainWindow::on_CSVButton_clicked()
{
    int len = ui->tableWidget->rowCount();
    csv = new CSV_View();
    QString row = "";
    for (int i = 0; i < len; ++i) {
        QString creator = ui->tableWidget->item(i, 0)->text();
        QString name = ui->tableWidget->item(i, 1)->text();
        QString color = ui->tableWidget->item(i, 2)->text();
        QString delivery_time = ui->tableWidget->item(i, 3)->text();
        QString count = ui->tableWidget->item(i, 4)->text();
        row += creator + ";" + name + ";" + color + ";" + delivery_time + ";" + count + ";\n";
    }
    csv->set_text(row);
    csv->exec();
}

void MainWindow::on_ExitButtonMenu_triggered()
{
    on_ExitButton_clicked();
}

void MainWindow::on_AboutButtonMenu_triggered()
{
    on_AboutButton_clicked();
}

void MainWindow::on_AddButtonMenu_triggered()
{
    on_AddButton_clicked();
}

void MainWindow::on_DeleteButtonMenu_triggered()
{
    on_DeleteButton_clicked();
}

void MainWindow::on_EditButtonMenu_triggered()
{
    on_EditButton_clicked();
}

void MainWindow::on_ClearButtonMenu_triggered()
{
    on_ClearButton_clicked();
}

void MainWindow::on_CreateButtonMenu_triggered()
{
    on_CreateButton_clicked();
}

void MainWindow::on_OpenButtonMenu_triggered()
```

```

{
    on_InputButton_clicked();
}

void MainWindow::on_SaveButtonMenu_triggered()
{
    on_SaveButton_clicked();
}

void MainWindow::on_SaveAsButtonMenu_triggered()
{
    on_SaveAsButton_clicked();
}

void MainWindow::on_MergeButtonMenu_triggered()
{
    on_Merge_clicked();
}

```

### adding.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>adding</class>
<widget class="QDialog" name="adding">
<property name="geometry">
<rect>
<x>0</x>
<y>0</y>
<width>390</width>
<height>377</height>
</rect>
</property>
<property name="windowTitle">
<string>Dialog</string>
</property>
<widget class="QWidget" name="layoutWidget">
<property name="geometry">
<rect>
<x>20</x>
<y>20</y>
<width>351</width>
<height>291</height>
</rect>
</property>
<layout class="QVBoxLayout" name="verticalLayout">
<item>
<layout class="QHBoxLayout" name="horizontalLayout">
<item>
<widget class="QLabel" name="label">
<property name="font">
<font>
<family>Times New Roman</family>
<pointsize>16</pointsize>
</font>
</property>
<property name="text">
<string>Команда</string>

```

```

    </property>
  </widget>
</item>
<item>
  <widget class="QLineEdit" name="creator"/>
</item>
</layout>
</item>
<item>
  <layout class="QHBoxLayout" name="horizontalLayout_2">
    <item>
      <widget class="QLabel" name="label_2">
        <property name="font">
          <font>
            <family>Times New Roman</family>
            <pointsize>16</pointsize>
          </font>
        </property>
        <property name="text">
          <string>Лига</string>
        </property>
      </widget>
    </item>
    <item>
      <widget class="QLineEdit" name="name"/>
    </item>
  </layout>
</item>
<item>
  <layout class="QHBoxLayout" name="horizontalLayout_3">
    <item>
      <widget class="QLabel" name="label_3">
        <property name="font">
          <font>
            <family>Times New Roman</family>
            <pointsize>16</pointsize>
          </font>
        </property>
        <property name="text">
          <string>Фамилия и имя</string>
        </property>
      </widget>
    </item>
    <item>
      <widget class="QLineEdit" name="color"/>
    </item>
  </layout>
</item>
<item>
  <layout class="QHBoxLayout" name="horizontalLayout_5">
    <item>
      <widget class="QLabel" name="label_5">
        <property name="font">
          <font>
            <family>Times New Roman</family>
            <pointsize>16</pointsize>
          </font>
        </property>
        <property name="text">
          <string>Позиция</string>
        </property>
      </widget>
    </item>
  </layout>
</item>

```

```

</item>
<item>
  <widget class="QLineEdit" name="delivery_time"/>
</item>
</layout>
</item>
<item>
  <layout class="QHBoxLayout" name="horizontalLayout_4">
    <item>
      <widget class="QLabel" name="label_4">
        <property name="font">
          <font>
            <family>Times New Roman</family>
            <pointsize>16</pointsize>
          </font>
        </property>
        <property name="text">
          <string>Home</string>
        </property>
      </widget>
    </item>
    <item>
      <widget class="QLineEdit" name="count">
        <property name="inputMask">
          <string/>
        </property>
        <property name="text">
          <string/>
        </property>
        <property name="placeholderText">
          <string>int</string>
        </property>
      </widget>
    </item>
  </layout>
</item>
</layout>
</widget>
<widget class="QWidget" name="layoutWidget">
  <property name="geometry">
    <rect>
      <x>20</x>
      <y>330</y>
      <width>351</width>
      <height>30</height>
    </rect>
  </property>
  <layout class="QHBoxLayout" name="horizontalLayout_6">
    <item>
      <widget class="QDialogButtonBox" name="buttonBox">
        <property name="orientation">
          <enum>Qt::Horizontal</enum>
        </property>
        <property name="standardButtons">
          <set>QDialogButtonBox::Cancel|QDialogButtonBox::Ok</set>
        </property>
      </widget>
    </item>
  </layout>
</widget>
</resources/>

```

```

<connections>
<connection>
  <sender>buttonBox</sender>
  <signal>accepted()</signal>
  <receiver>adding</receiver>
  <slot>accept()</slot>
<hints>
  <hint type="sourcelabel">
    <x>248</x>
    <y>254</y>
  </hint>
  <hint type="destinationlabel">
    <x>157</x>
    <y>274</y>
  </hint>
</hints>
</connection>
<connection>
  <sender>buttonBox</sender>
  <signal>rejected()</signal>
  <receiver>adding</receiver>
  <slot>reject()</slot>
<hints>
  <hint type="sourcelabel">
    <x>316</x>
    <y>260</y>
  </hint>
  <hint type="destinationlabel">
    <x>286</x>
    <y>274</y>
  </hint>
</hints>
</connection>
</connections>
</ui>

```

#### csv\_view.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>CSV_View</class>
  <widget class="QDialog" name="CSV_View">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>400</width>
        <height>337</height>
      </rect>
    </property>
    <property name="windowTitle">
      <string>Dialog</string>
    </property>
    <widget class="QTextEdit" name="textEdit">
      <property name="geometry">
        <rect>
          <x>10</x>
          <y>10</y>
          <width>381</width>
          <height>291</height>
        </rect>
      </property>

```



```

<property name="readOnly">
  <bool>true</bool>
</property>
</widget>
<widget class="QPushButton" name="closeButton">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>298</y>
      <width>419</width>
      <height>41</height>
    </rect>
  </property>
  <property name="text">
    <string>Заккрыть</string>
  </property>
  <property name="flat">
    <bool>true</bool>
  </property>
</widget>
</widget>
<resources/>
<connections/>
</ui>

```

#### mainwindow.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>MainWindow</class>
  <widget class="QMainWindow" name="MainWindow">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>743</width>
        <height>407</height>
      </rect>
    </property>
    <property name="windowTitle">
      <string>Курсовая работа</string>
    </property>
    <property name="toolTip">
      <string/>
    </property>
    <property name="unifiedTitleAndToolBarOnMac">
      <bool>>false</bool>
    </property>
    <widget class="QWidget" name="centralwidget">
      <widget class="QLineEdit" name="SearchBar">
        <property name="geometry">
          <rect>
            <x>10</x>
            <y>330</y>
            <width>721</width>
            <height>21</height>
          </rect>
        </property>
        <property name="whatsThis">
          <string/>

```

```

</property>
<property name="autoFillBackground">
  <bool>false</bool>
</property>
<property name="placeholderText">
  <string>Поиск</string>
</property>
</widget>
<widget class="QWidget" name="layoutWidget">
  <property name="geometry">
    <rect>
      <x>10</x>
      <y>0</y>
      <width>746</width>
      <height>43</height>
    </rect>
  </property>
  <layout class="QHBoxLayout" name="horizontalLayout_2">
    <item>
      <layout class="QHBoxLayout" name="horizontalLayout">
        <item>
          <widget class="QPushButton" name="CreateButton">
            <property name="toolTip">
              <string/>
            </property>
            <property name="statusTip">
              <string/>
            </property>
            <property name="text">
              <string/>
            </property>
            <property name="icon">
              <iconset resource="resource.qrc">
                <normaloff>:/img/add.png</normaloff>:/img/add.png</iconset>
              </property>
            <property name="iconSize">
              <size>
                <width>30</width>
                <height>30</height>
              </size>
            </property>
            <property name="autoDefault">
              <bool>false</bool>
            </property>
            <property name="default">
              <bool>false</bool>
            </property>
            <property name="flat">
              <bool>true</bool>
            </property>
          </widget>
        </item>
        <item>
          <widget class="QPushButton" name="InputButton">
            <property name="text">
              <string/>
            </property>
            <property name="icon">
              <iconset resource="resource.qrc">
                <normaloff>:/img/share.png</normaloff>:/img/share.png</iconset>
              </property>
            <property name="iconSize">

```

```

    <size>
    <width>30</width>
    <height>30</height>
    </size>
  </property>
  <property name="flat">
    <bool>true</bool>
  </property>
</widget>
</item>
<item>
  <widget class="QPushButton" name="SaveButton">
    <property name="text">
      <string/>
    </property>
    <property name="icon">
      <iconset resource="resource.qrc">
        <normaloff>:/img/diskette.png</normaloff>:/img/diskette.png</iconset>
      </property>
    <property name="iconSize">
      <size>
        <width>30</width>
        <height>30</height>
      </size>
    </property>
    <property name="flat">
      <bool>true</bool>
    </property>
  </widget>
</item>
<item>
  <widget class="QPushButton" name="SaveAsButton">
    <property name="text">
      <string/>
    </property>
    <property name="icon">
      <iconset resource="resource.qrc">
        <normaloff>:/img/save-as.png</normaloff>:/img/save-as.png</iconset>
      </property>
    <property name="iconSize">
      <size>
        <width>30</width>
        <height>30</height>
      </size>
    </property>
    <property name="flat">
      <bool>true</bool>
    </property>
  </widget>
</item>
<item>
  <widget class="QPushButton" name="Merge">
    <property name="text">
      <string/>
    </property>
    <property name="icon">
      <iconset resource="resource.qrc">
        <normaloff>:/img/merge.png</normaloff>:/img/merge.png</iconset>
      </property>
    <property name="iconSize">
      <size>
        <width>30</width>

```

```

    <height>30</height>
  </size>
</property>
<property name="flat">
  <bool>true</bool>
</property>
</widget>
</item>
<item>
  <widget class="QPushButton" name="ExitButton">
    <property name="text">
      <string/>
    </property>
    <property name="icon">
      <iconset resource="resource.qrc">
        <normaloff>:/img/exit.png</normaloff>:/img/exit.png</iconset>
      </property>
    <property name="iconSize">
      <size>
        <width>30</width>
        <height>30</height>
      </size>
    </property>
    <property name="flat">
      <bool>true</bool>
    </property>
  </widget>
</item>
<item>
  <widget class="Line" name="line">
    <property name="orientation">
      <enum>Qt::Vertical</enum>
    </property>
  </widget>
</item>
<item>
  <widget class="QPushButton" name="AddButton">
    <property name="text">
      <string/>
    </property>
    <property name="icon">
      <iconset resource="resource.qrc">
        <normaloff>:/img/free-icon-plus-7263361.png</normaloff>:/img/free-icon-plus-7263361.png</iconset>
      </property>
    <property name="iconSize">
      <size>
        <width>30</width>
        <height>30</height>
      </size>
    </property>
    <property name="flat">
      <bool>true</bool>
    </property>
  </widget>
</item>
<item>
  <widget class="QPushButton" name="DeleteButton">
    <property name="text">
      <string/>
    </property>
    <property name="icon">
      <iconset resource="resource.qrc">

```

```

    <normaloff>:/img/close.png</normaloff>:/img/close.png</iconset>
  </property>
  <property name="iconSize">
    <size>
      <width>30</width>
      <height>30</height>
    </size>
  </property>
  <property name="flat">
    <bool>true</bool>
  </property>
</widget>
</item>
<item>
  <widget class="QPushButton" name="EditButton">
    <property name="text">
      <string/>
    </property>
    <property name="icon">
      <iconset resource="resource.qrc">
        <normaloff>:/img/edit2.png</normaloff>:/img/edit2.png</iconset>
      </iconset>
    </property>
    <property name="iconSize">
      <size>
        <width>30</width>
        <height>30</height>
      </size>
    </property>
    <property name="flat">
      <bool>true</bool>
    </property>
  </widget>
</item>
<item>
  <widget class="QPushButton" name="ClearButton">
    <property name="text">
      <string/>
    </property>
    <property name="icon">
      <iconset resource="resource.qrc">
        <normaloff>:/img/eraser.png</normaloff>:/img/eraser.png</iconset>
      </iconset>
    </property>
    <property name="iconSize">
      <size>
        <width>30</width>
        <height>30</height>
      </size>
    </property>
    <property name="flat">
      <bool>true</bool>
    </property>
  </widget>
</item>
<item>
  <widget class="Line" name="line_2">
    <property name="orientation">
      <enum>Qt::Vertical</enum>
    </property>
  </widget>
</item>
<item>
  <widget class="QPushButton" name="CSVButton">

```

```

<property name="text">
<string/>
</property>
<property name="icon">
<iconset resource="resource.qrc">
<normaloff>:/img/menu.png</normaloff>:/img/menu.png</iconset>
</property>
<property name="iconSize">
<size>
<width>30</width>
<height>30</height>
</size>
</property>
<property name="flat">
<bool>true</bool>
</property>
</widget>
</item>
<item>
<widget class="Line" name="line_3">
<property name="orientation">
<enum>Qt::Vertical</enum>
</property>
</widget>
</item>
<item>
<widget class="QPushButton" name="AboutButton">
<property name="text">
<string/>
</property>
<property name="icon">
<iconset resource="resource.qrc">
<normaloff>:/img/user2.png</normaloff>:/img/user2.png</iconset>
</property>
<property name="iconSize">
<size>
<width>30</width>
<height>30</height>
</size>
</property>
<property name="flat">
<bool>true</bool>
</property>
</widget>
</item>
</layout>
</item>
<item>
<widget class="QProgressBar" name="progressBar">
<property name="value">
<number>0</number>
</property>
</widget>
</item>
</layout>
</widget>
<widget class="QTableWidget" name="tableWidget">
<property name="geometry">
<rect>
<x>10</x>
<y>40</y>
<width>719</width>

```

```

    <height>281</height>
  </rect>
</property>
<property name="tabletTracking">
  <bool>false</bool>
</property>
<property name="autoFillBackground">
  <bool>false</bool>
</property>
<attribute name="horizontalHeaderCascadingSectionResizes">
  <bool>true</bool>
</attribute>
<attribute name="horizontalHeaderDefaultSectionSize">
  <number>120</number>
</attribute>
<attribute name="horizontalHeaderMinimumSectionSize">
  <number>35</number>
</attribute>
<attribute name="horizontalHeaderStretchLastSection">
  <bool>true</bool>
</attribute>
<attribute name="verticalHeaderVisible">
  <bool>true</bool>
</attribute>
<attribute name="verticalHeaderCascadingSectionResizes">
  <bool>false</bool>
</attribute>
<attribute name="verticalHeaderDefaultSectionSize">
  <number>30</number>
</attribute>
<attribute name="verticalHeaderMinimumSectionSize">
  <number>19</number>
</attribute>
<attribute name="verticalHeaderStretchLastSection">
  <bool>false</bool>
</attribute>
</widget>
</widget>
<widget class="QMenuBar" name="menubar">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>0</y>
      <width>743</width>
      <height>26</height>
    </rect>
  </property>
  <widget class="QMenu" name="menu">
    <property name="title">
      <string>Файл</string>
    </property>
    <addaction name="CreateButtonMenu"/>
    <addaction name="OpenButtonMenu"/>
    <addaction name="SaveButtonMenu"/>
    <addaction name="SaveAsButtonMenu"/>
    <addaction name="MergeButtonMenu"/>
    <addaction name="ExitButtonMenu"/>
  </widget>
  <widget class="QMenu" name="menu_2">
    <property name="title">
      <string>Запись</string>
    </property>

```

```

<addaction name="AddButtonMenu"/>
<addaction name="DeleteButtonMenu"/>
<addaction name="EditButtonMenu"/>
<addaction name="ClearButtonMenu"/>
</widget>
<widget class="QMenu" name="menu_3">
  <property name="title">
    <string>Помощь</string>
  </property>
  <addaction name="AboutButtonMenu"/>
</widget>
<addaction name="menu"/>
<addaction name="menu_2"/>
<addaction name="menu_3"/>
</widget>
<widget class="QStatusBar" name="statusbar"/>
<action name="AboutButtonMenu">
  <property name="text">
    <string>О программе</string>
  </property>
  <property name="shortcut">
    <string>F1</string>
  </property>
</action>
<action name="AddButtonMenu">
  <property name="text">
    <string>Добавить</string>
  </property>
  <property name="shortcut">
    <string>Ctrl+A</string>
  </property>
</action>
<action name="DeleteButtonMenu">
  <property name="text">
    <string>Удалить</string>
  </property>
  <property name="shortcut">
    <string>Ctrl+D</string>
  </property>
</action>
<action name="EditButtonMenu">
  <property name="text">
    <string>Редактировать</string>
  </property>
  <property name="shortcut">
    <string>Ctrl+E</string>
  </property>
</action>
<action name="ClearButtonMenu">
  <property name="text">
    <string>Очистить</string>
  </property>
  <property name="shortcut">
    <string>Ctrl+P</string>
  </property>
</action>
<action name="CreateButtonMenu">
  <property name="text">
    <string>Создать</string>
  </property>
  <property name="shortcut">
    <string>Ctrl+N</string>
  </property>
</action>

```



```
</property>
</action>
<action name="OpenButtonMenu">
  <property name="text">
    <string>Открыть</string>
  </property>
  <property name="shortcut">
    <string>Ctrl+O</string>
  </property>
</action>
<action name="SaveButtonMenu">
  <property name="text">
    <string>Сохранить</string>
  </property>
  <property name="shortcut">
    <string>Ctrl+S</string>
  </property>
</action>
<action name="SaveAsButtonMenu">
  <property name="text">
    <string>Сохранить как</string>
  </property>
  <property name="shortcut">
    <string>Ctrl+Alt+S</string>
  </property>
</action>
<action name="MergeButtonMenu">
  <property name="text">
    <string>Объединить</string>
  </property>
  <property name="shortcut">
    <string>Ctrl+M</string>
  </property>
</action>
<action name="ExitButtonMenu">
  <property name="text">
    <string>Выход</string>
  </property>
  <property name="shortcut">
    <string>Ctrl+X</string>
  </property>
</action>
</widget>
<resources>
  <include location="resource.qrc"/>
</resources>
<connections/>
</ui>
```

## **8. Выводы**

В ходе курсовой работы была разработана база данных, обеспечивающая выполнение всех требуемых операции: создание базы данных, чтение из файла, сохранение базы данных, редактирование/удаление записей, объединение баз данных, поиск записей, просмотр информации о программе и так далее.

Был разработан интуитивно-понятный интерфейс, с которым разберется даже неопытный пользователь компьютера. Предусмотрено меню, панель инструментов с иконами и множество горячих клавиш.

В ходе курсовой работы были изучены основы библиотеки QT и возможности среды разработки QT Creator.

## **9. Список литературы**

1. Справочные материалы Qt Creator. (Использовались при построении графического интерфейса и на протяжении всей разработки программы)
2. Коробов С. А. Лекции по дисциплине Объектно–ориентированное программирование. (Использовались при раз работке класс базы данных)