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](#_Toc122213286)

[1.1 Требования к функциям базы данных 3](#_Toc122213287)

[1.2 Требования к интерфейсу программы 3](#_Toc122213288)

[2. Выбор и обоснование типа разрабатываемого контейнера 4](#_Toc122213289)

[3. Разработка структуры программы 5](#_Toc122213290)

[4. Разработка собственных классов 6](#_Toc122213291)

[4.1 MainWindow – класс главного окна 6](#_Toc122213292)

[4.2 Adding– класс для добавления или редактирования записей 6](#_Toc122213293)

[4.3 CSV\_View– класс для отображения содержимого в csv-формате 6](#_Toc122213294)

[5. Разработка интерфейса 7](#_Toc122213295)

[5.1. Основное окно программы 7](#_Toc122213296)

[5.2 Диалоговое окно для добавления/редактирования записей 8](#_Toc122213297)

[5.3 Диалоговое окно для отображения данных в csv-формате 8](#_Toc122213298)

[6. Инструкция пользователю 9](#_Toc122213299)

[6.1 Добавление записи в базу данных 9](#_Toc122213300)

[6.2. Редактирование записи 10](#_Toc122213301)

[6.3. Удаление записи 11](#_Toc122213302)

[6.4. Очистка таблицы 12](#_Toc122213303)

[6.5. Создание новой базы данных 13](#_Toc122213304)

[6.6. Открыть, сохранить, сохранить как 14](#_Toc122213305)

[6.7. Объединение баз данных 16](#_Toc122213306)

[6.8. CSV-отображение данных 17](#_Toc122213307)

[6.9. Поиск 17](#_Toc122213308)

[6.10 Отображение информации о программе и закрытие программы. 18](#_Toc122213309)

[7. Код программы 19](#_Toc122213310)

[8. Выводы 41](#_Toc122213311)

[9. Список литературы 42](#_Toc122213312)

# **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 программировании Qt Creator.

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

*Таблица 1.*

|  |  |
| --- | --- |
| **Вариант** | **Класс** |
| 13 | Класс, характеризующий игрока футбольной команды |

## 

## **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.

# **Выбор и обоснование типа разрабатываемого контейнера**

В курсовой работе необходимо использовать контейнер для хранения объектов.

В качестве прототипа разрабатываемого контейнера выбран vector. Выбор данного контейнера обусловлен относительной простотой разработки и организации, а также возможностью произвольного доступа к своим элемента.

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**

## **MainWindow – класс главного окна**

***Методы класса:***

|  |  |  |
| --- | --- | --- |
| **void** | **on\_CreateButton\_clicked** | открытие нового файла |
| ***void*** | **on\_ExitButton\_clicked** | выход из программы |
| ***void*** | **on\_AddButton\_clicked** | добавление записи |
| ***void*** | **on\_EditButton\_clicked** | изменение записи |
| ***void*** | **on\_AboutButton\_clicked** | информация о программе и авторе |
| ***void*** | **on\_DeleteButton\_clicked** | удаление записи |
| ***void*** | **on\_ClearButton\_clicked** | очистка таблицы |
| ***void*** | **on\_InputButton\_clicked** | чтение БД из файла |
| ***void*** | **on\_Merge\_clicked** | объединение БД |
| ***void*** | **on\_SaveButton\_clicked** | сохранение |
| ***void*** | **on\_SaveAsButton\_clicked** | сохранить как |
| ***void*** | **on\_SearchBar\_textChanged(const QString &arg1)** | поиск в searchBar |
| ***void*** | **on\_CSVButton\_clicked** | отображение данных через «;» (csv-формат) |

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

***Методы класса:***

|  |  |  |
| --- | --- | --- |
| ***void*** | **on\_buttonBox\_accepted** | *Принятие добавления или изменения* |
| ***void*** | ***setData(QList<QTableWidgetItem \*> list)*** | *Изменение/добавление данных в таблице* |

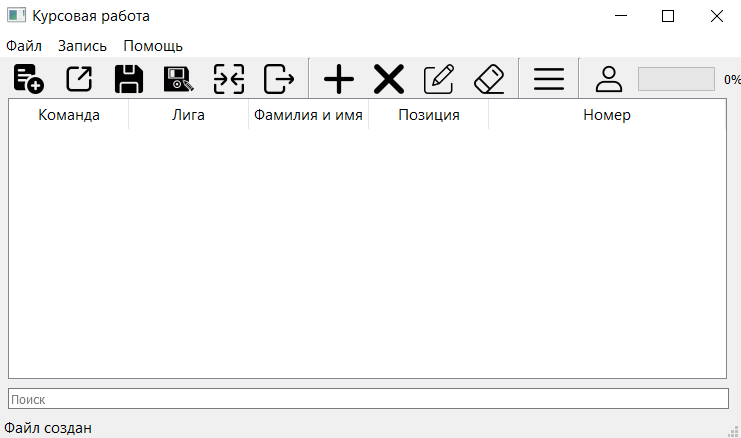
## **CSV\_View– класс для отображения содержимого в csv-формате**

***Методы класса:***

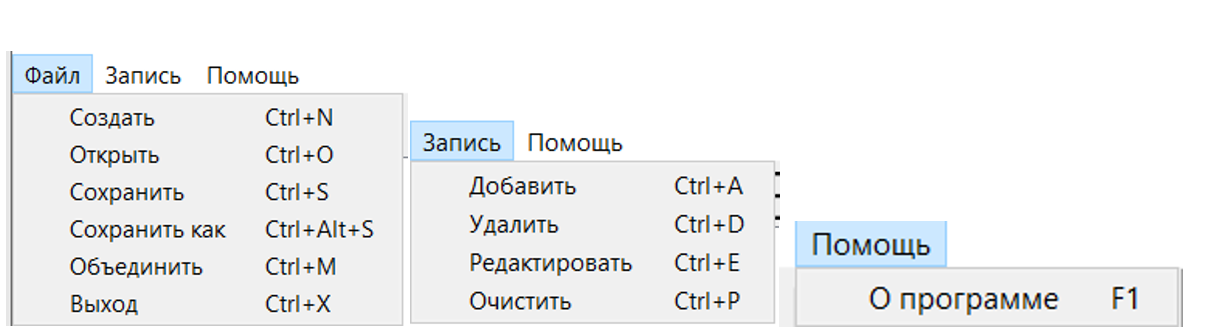
|  |  |  |
| --- | --- | --- |
| **explicit** | **CSV\_View(QWidget \*parent = nullptr)** | *Конструктор* |
| ***void*** | **on\_closeButton\_clicked** | *Закрытие окна* |
| ***void*** | **set\_text(QString row)** | *Вывод элементов таблицы в поле через “;”* |

# **5.Разработка интерфейса**

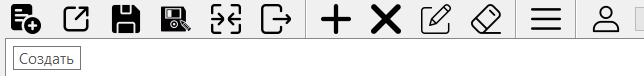
5.1. Основное окно программы

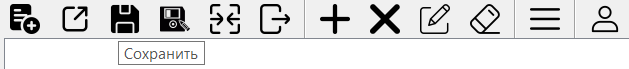


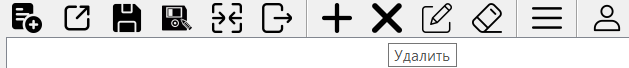
*Рисунок 1. Основное окно программы (меню, панель инструментов с иконками, таблица,  
строка поиска, строка состояния, иконка программы)*



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

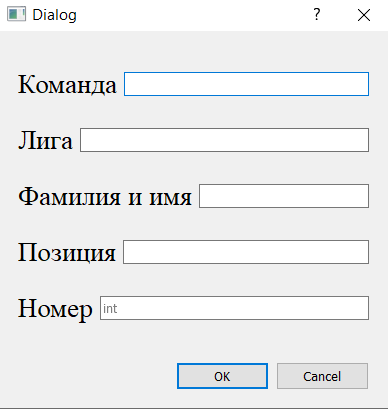
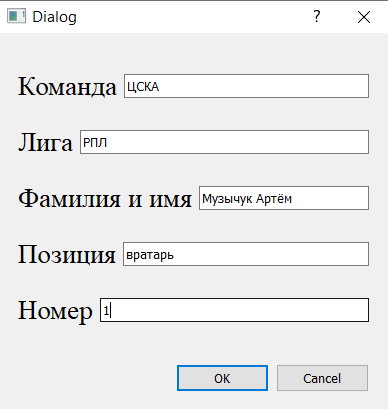
**

**

**

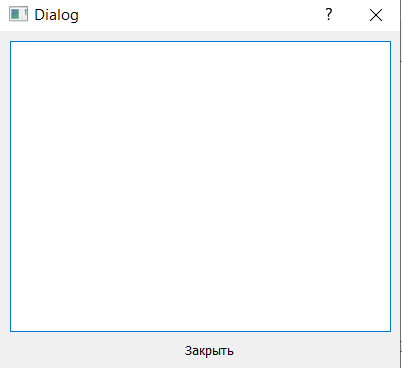
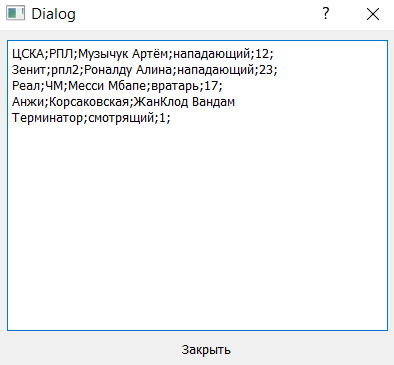
*Рисунок 3. Панель инструментов с интуитивно понятными иконами  
и подсказками при наведении.*

5.2 Диалоговое окно для добавления/редактирования записей

******

*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

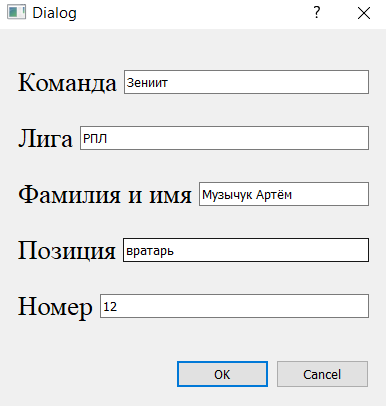
 

*Рисунок 5. Пустое и заполненное окно с данными таблицы через «;», только для чтения.*

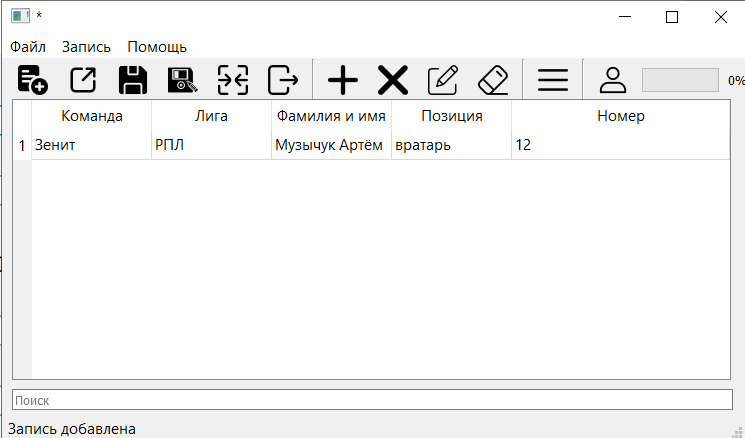
# **6.Инструкция пользователю**

6.1 Добавление записи в базу данных

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



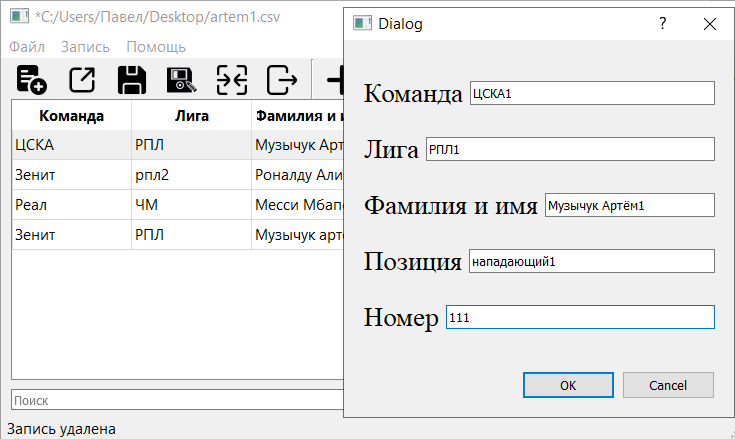
*Рисунок 6. Добавление записей через диалоговое окно.*

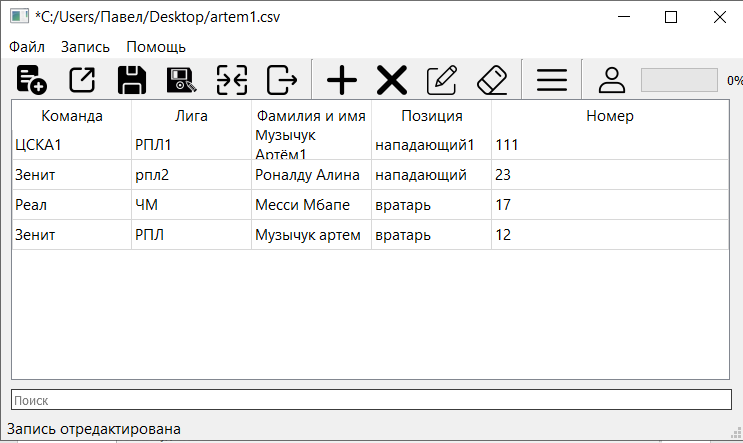


*Рисунок 7. Результат добавления записей, отображение информации в статусной строке.*

6.2. Редактирование записи

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

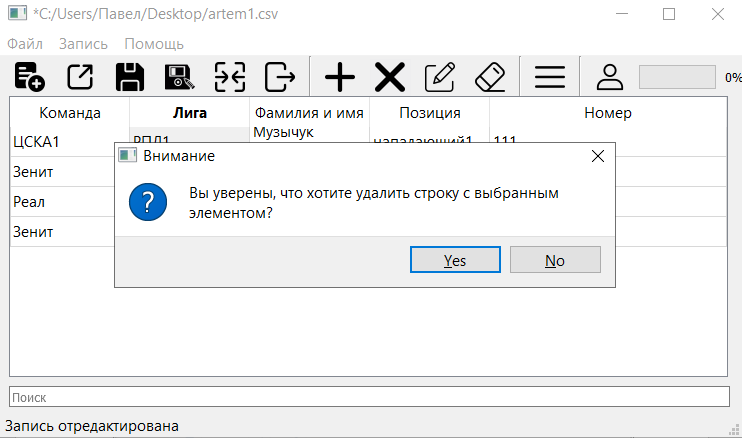


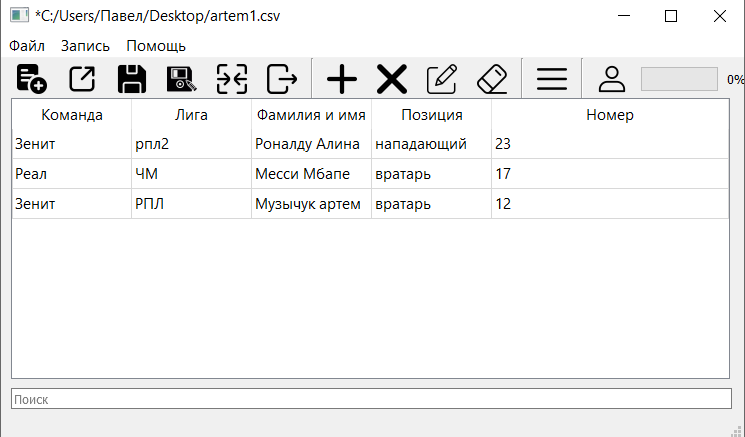


*Рисунок 8. Редактирование номинала «СВЧ усилителя», сохранение в таблице, информация в статусной строке.*

6.3. Удаление записи

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

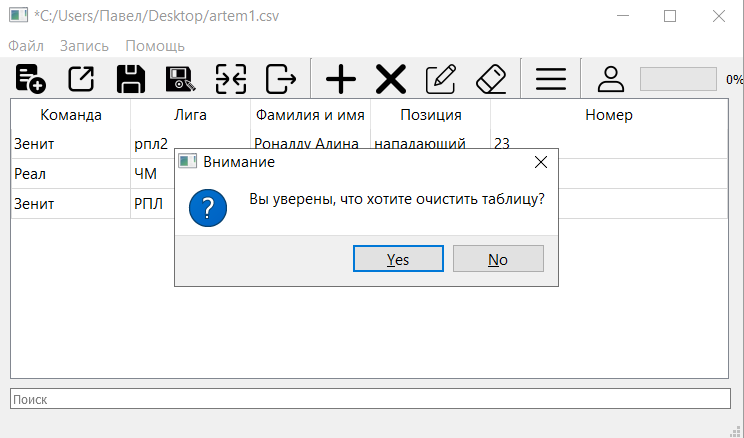


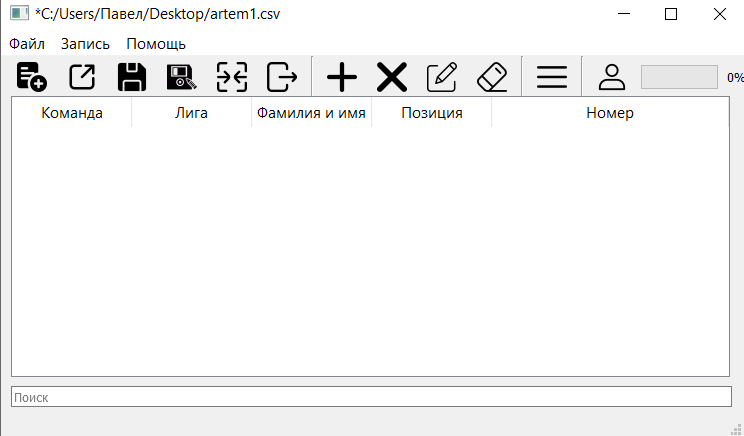


*Рисунок 8. Удаление записи с подтверждением, вывод информации в статусную строку*.

6.4. Очистка таблицы

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

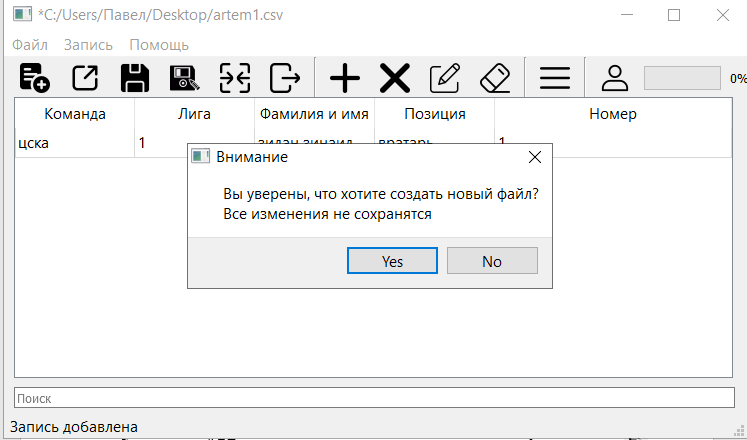


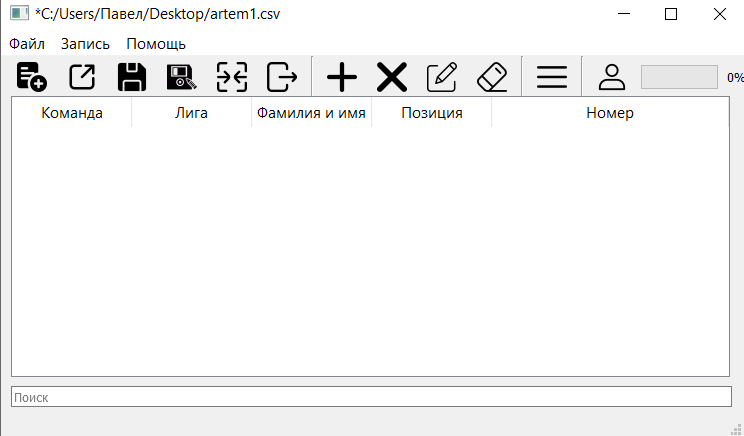


*Рисунок 9. Очистка таблицы, информация в статусной строке.*

6.5. Создание новой базы данных

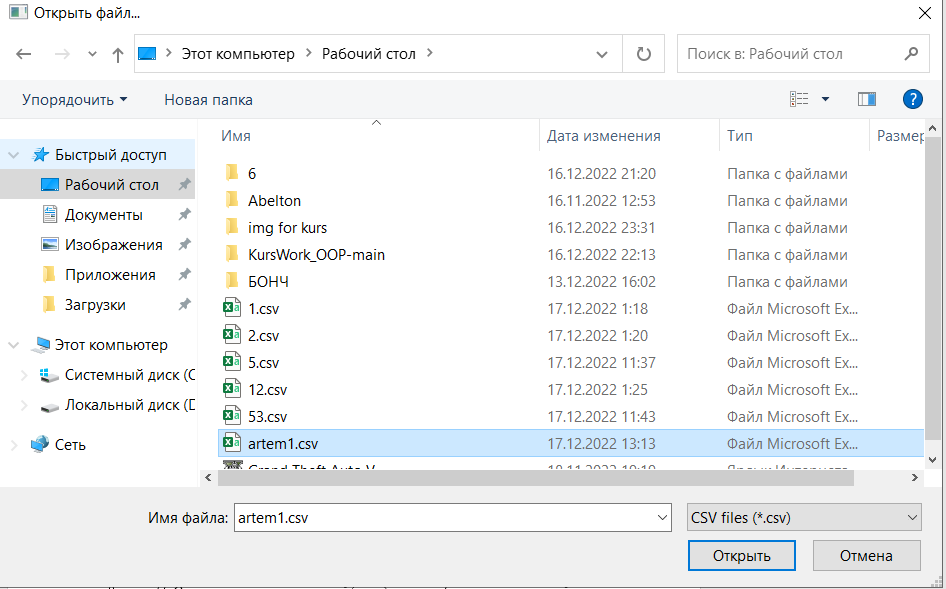
Создание новой БД осуществляется с помощью меню, сочетания **ctrl+n** или кнопки .

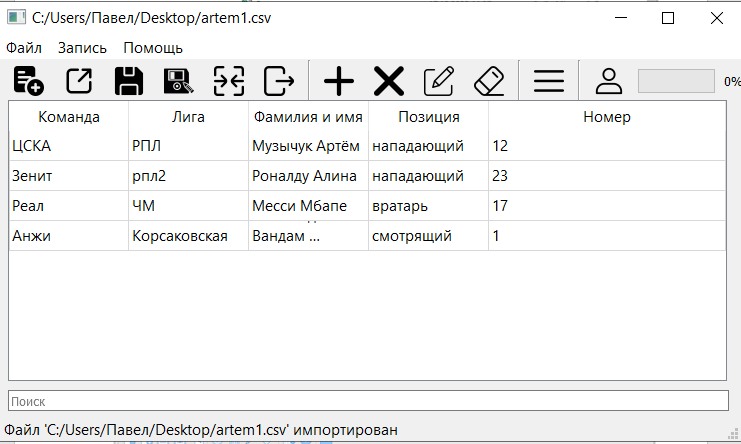




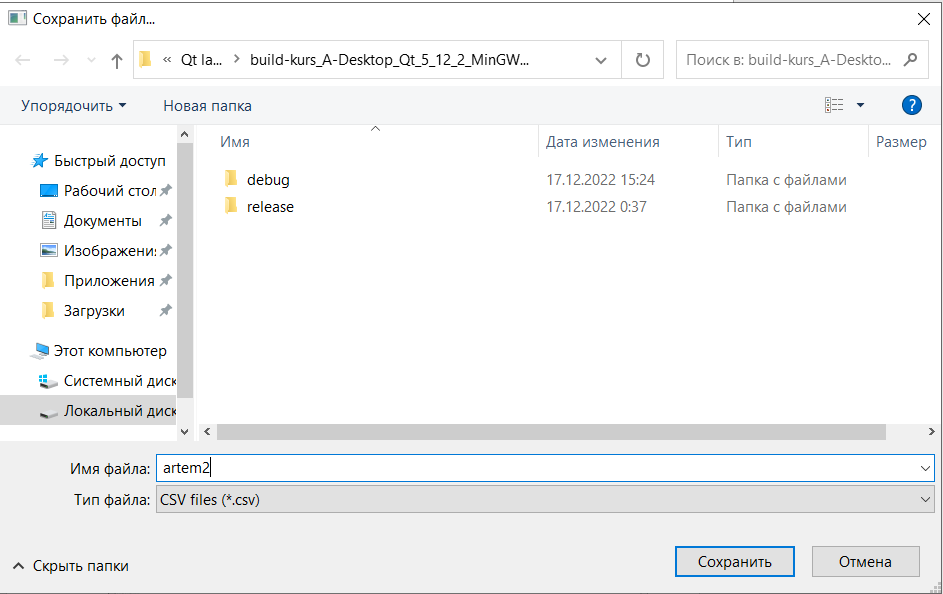
6.6. Открыть, сохранить, сохранить как

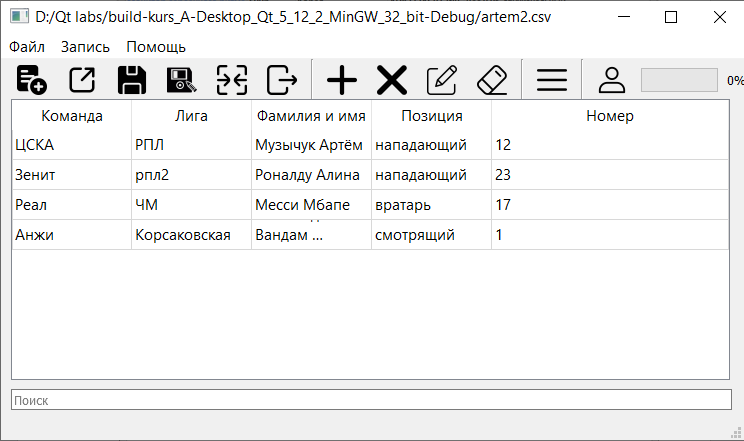
Открытие уже существующей базы данных осуществляется с помощью меню, сочетания **ctrl+o** или кнопки , сохранение – сочетание **ctrl+s** или кнопка  , сохранить как – сочетание **ctrl+alt+s** или кнопка  .





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

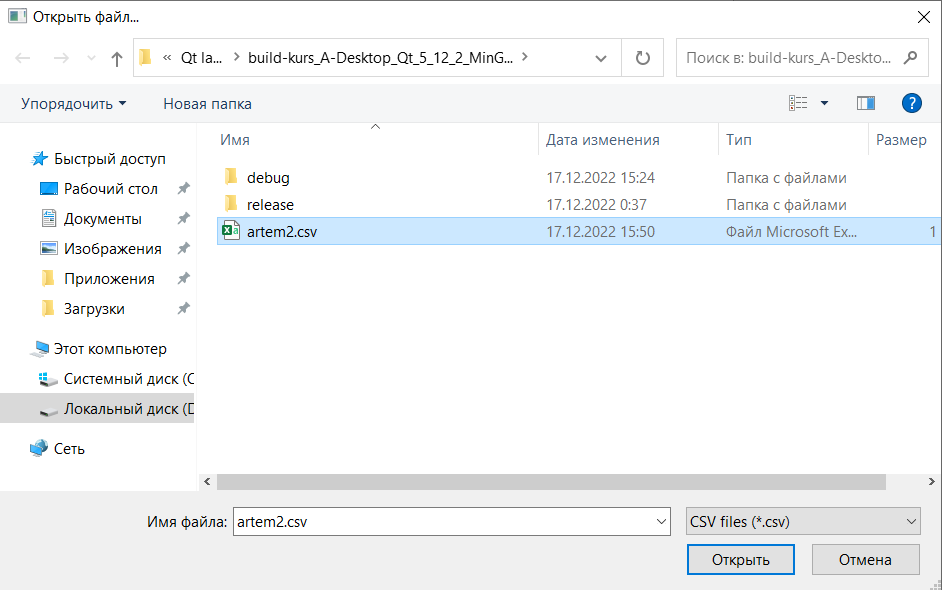
**

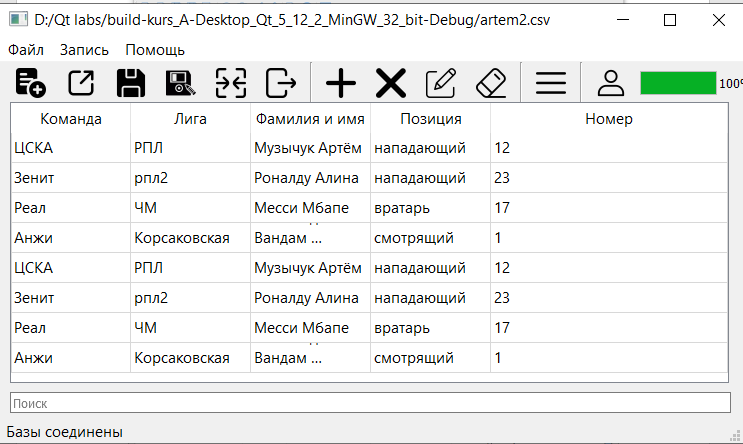
**

*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  .

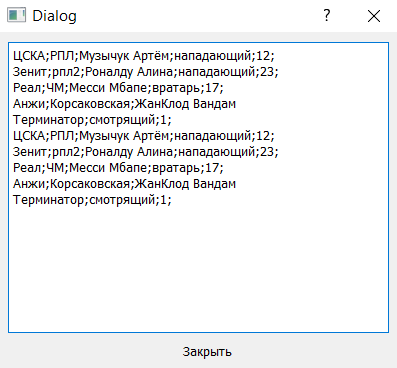




*Рисунок 13. Объединение «база\_данных\_2.csv» с «база\_данных.csv», информация в статусной строке.*

6.8. CSV-отображение данных

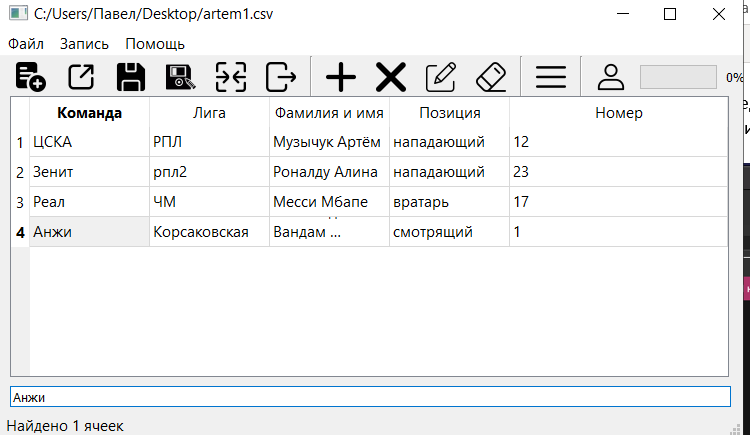
Просмотр данных в формате CSV осуществляется с помощью меню, сочетания **ctrl+`** или 



*Рисунок 14. CSV-отображение.*

6.9. Поиск

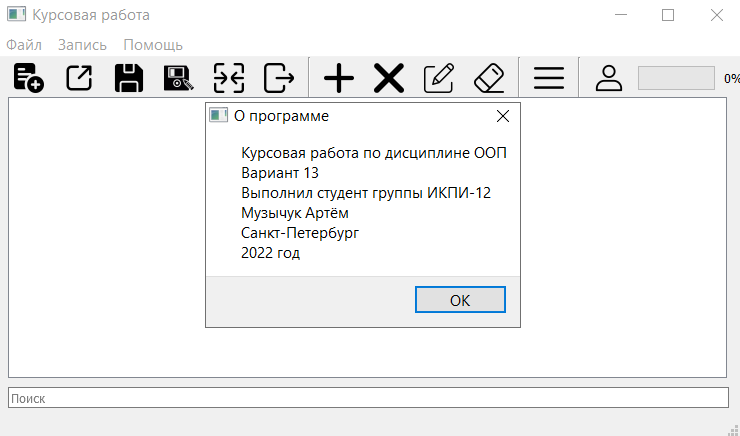
В строке поиска, находящееся над основным окном. Введите данные, которые хотите найти. В таблице автоматически выделятся совпадения, а внизу окна выведется какое количество ячеек совпадает с введенными данными:



*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.*

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

# **7. Код программы**

|  |
| --- |
| **kurs\_A.pro** |
| 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 |

|  |
| --- |
| **adding.h** |
| #ifndef ADDING\_H  #define ADDING\_H  #include <QDialog>  #include <QTableWidgetItem>  namespace Ui {  class adding;  }  class adding : public QDialog  {  Q\_OBJECT  public:  explicit adding(QWidget \*parent = nullptr);  ~adding();  void setData(QList<QTableWidgetItem \*> list/\*QString creator, QString name, QString color, QString delivery\_time, QString count\*/);  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 addding(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->addding(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("Объеденить");  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::addding(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(";");  addding(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(";");  addding(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 проделанные 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>  <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>Номер</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>  </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>Закрыть</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>  </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>  </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>  <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)