МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (ФГБОУ ВО «ВГТУ», ВГТУ)

Факультет информационных технологий и компьютерной безопасности

(факультет)

Кафедра компьютерных интеллектуальных технологий проектирования

КУРСОВОЙ ПРОЕКТ

по дисциплине «Среды визуального программирования»

Тема «Картотека пациентов»

Расчетно-пояснительная записка

Разработал студент М. Е. Федоров

гр. бИВТ-223 Подпись, дата Инициалы, фамилия

Руководитель В. В. Сокольников Подпись, дата Инициалы, фамилия

Нормоконтролер

В. В. Сокольников Подпись, дата Инициалы, фамилия

Защищена Оценка

дата

Воронеж 2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (ФГБОУ ВО «ВГТУ», ВГТУ)

Кафедра компьютерных интеллектуальных технологий проектирования

Задание на курсовой проект

по дисциплине «Среды визуального программирования»

Тема работы «Картотека пациентов»

Вариант работы 16

Студент группы бИВТ-223, Федоров Михаил Евгеньевич

Фамилия, имя, отчество

Технические условия: AMD Ryzen 5 3600, RTX 2060, 16 ГБ ОЗУ, Microsoft Word 2016

Содержание и объем работы (графические работы, расчеты и прочее): 30 страниц, 21 рисунок, 0 таблиц, 6 приложений

Сроки выполнения этапов Рассмотрение теоретических сведений (сентябрь-декабрь 2023); оформление пояснительной записки (декабрь 2023);

Срок защиты курсового проекта сентябрь-декабрь 2023

Руководитель В.В. Сокольников

Подпись, дата Инициалы, фамилия

Задание принял студент М.Е. Федоров

Подпись, дата Инициалы, фамилия

# Замечания руководителя

ОГЛАВЛЕНИЕ

[Задание на курсовой проект 2](#_Toc153971766)

[Замечания руководителя 3](#_Toc153971767)

[Введение 5](#_Toc153971768)

[1 Теоретические сведения 6](#_Toc153971769)

[2 Описание инструментов разработки 7](#_Toc153971770)

[2.1 Переход между окнами 7](#_Toc153971771)

[2.2 Закрытие окон 7](#_Toc153971772)

[2.3 Отображение информации о программе 8](#_Toc153971773)

[2.4 Переход к окну управления 8](#_Toc153971774)

[2.5 Чтение файла Excel 8](#_Toc153971775)

[2.6 Открытие файла Excel 9](#_Toc153971776)

[2.7 Запись данных в таблицу 9](#_Toc153971777)

[2.8 Конвертация типов данных 10](#_Toc153971778)

[2.9 Проверка входной строки 11](#_Toc153971779)

[2.10 Класс mainwindow.h 12](#_Toc153971780)

[2.11 Класс management.h 12](#_Toc153971781)

[2.12 Класс patientLinkedList.h 12](#_Toc153971782)

[2.13 Класс addPatient.h 13](#_Toc153971783)

[2.14 Класс delPatient.h 13](#_Toc153971784)

[2.15 Классы fio.h, height.h, weight.h, yearOfBirth.h 13](#_Toc153971785)

[3 Проект программного средства 14](#_Toc153971786)

[4 Сценарий тестирования программного средства 18](#_Toc153971787)

[5 Руководство пользователя 28](#_Toc153971788)

[Заключение 29](#_Toc153971789)

[Список литературы 30](#_Toc153971790)

# Введение

Qt - кроссплатформенный фреймворк, который обычно используется для создания приложений с графическим интерфейсом. Стоит отметить, что набор инструментов позволяет разрабатывать и многие другие утилиты. Фреймворк запускается на трёх наиболее часто используемых операционных системах (Windows, Linux, MacOS) также хорошо, как и на мобильных ОС (Symbian, Nokia Belle, Meego Harmattan, MeeGo) и на встраиваемых устройствах. Qt также адаптирован для разработки программного обеспечения на Android и iOS.

Выполнение работы происходит в Qt Creator – это интегрированная среда разработки для языка программирования C++. Она очень хорошо подходит для программирования приложений Qt. Среда предоставляет браузер документов и "конструктор", который упрощает создание окон. Всё это дополнено продуманным пользовательским интерфейсом и скоростью сборки файлов в единое приложения.

В рамках курсового проекта было получено задание разработать программное обеспечение в сфере медицины. Цель работы заключается в создании приложения картотеки пациентов, позволяющего управлять списком и хранить его данные в виде файла.

# Теоретические сведения

Для разработки программного обеспечения «Картотека пациентов» требуется определить способ хранения данных и их последующее управление.

Хранить поля с данными пациентов было решено в Excel-документе. Управление осуществляется при помощи использования двусвязного списка, разработка которого написана в приложении 1.

Программную реализацию можно разделить на несколько этапов:

* запуск программы, во время которого считывается файл с данными пациентов и организовывается двусвязный список;
* работа пользователя, во время которой осуществляется управление данными (добавление, удаление) при помощи двусвязного списка;
* открытие Excel-файла, во время которого считывается измененный двусвязный список и перезаписывается файл с данными;
* закрытие программы, во время которого файл с данными перезаписывается.

# Описание инструментов разработки

В процессе выполнения курсовой работы было использовано множество инструментов, встроенных в интегрированную среду разработки «QT Creator». Остановимся на каждом из них.

## Переход между окнами

Программа состоит из форм, переход между которыми осуществляется после нажатия определенных кнопок. При нажатии на кнопку вызывается условный метод «on\_yourButton\_clicked()» класса «yourClass.h», которому принадлежит форма с данной кнопкой. Стоит отметить, что модификатор доступа данного метода имеет метку «private» и доступ к управлению кнопкой другим классам ограничен. Будем считать, что данная кнопка необходима для открытия другой формы. В процессе выполнения вызывается встроенный метод «this->hide()», скрывающий окно с кнопкой и отправляется условный сигнал «mySignal», отмеченный модификатором доступа «private» класса формы с кнопкой. После отправления сигнала должен вызываться метод формы, которую мы хотим отобразить, модификатор доступа должен иметь тип «public», поскольку метод вызывается со стороны класса первоначального окна. В вызванном методе вызывается встроенный метод «this->show()», который отображает новую форму.

Теперь необходимо соединить сигнал и слот при помощи встроенного метода «connect». В качестве аргументов необходимо передать объект класса, отправляющего сигнал, объект класса, реализовывающего определенный метод (слот), название сигнала и слота. В конечном счёте команда выглядит по типу «connect(signalObject, &signalClass::mySignal, slotObject, &slotClass::mySlot)». Данный метод написан в приложении 2.

## Закрытие окон

При закрытии программы (значок крестика в правом верхнем углу) запускается встроенный метод «closeEvent()» класса «QCloseEvent», наследника «QEvent». Он переопределен в окнах «меню», «управление» программы следующим образом: отображается окно с кнопками запроса «Да/Нет» и при соглашении программа завершает работу, предварительно сохранив данные списка в Excel-файл вызовом определенного сигнала. При нажатии на кнопку «нет» пользователь продолжает работу программы (вызов функции игнорируется, event->ignore()).

## Отображение информации о программе

В меню программы на вкладке «Программа» (левый верхний угол) доступны действия для отображения информации для пользователя и разработчика. При нажатии на действие считывается файл с соответствующей информацией и записывается в скрытое текстовое поле. После завершения чтения отображается текстовое поле при помощи встроенного метода, описанного в пункте 2.1.

## Переход к окну управления

При нажатии на действие «Вход» отображаются элементы, позволяющие вводить импровизированный логин и пароль пользователя. При нажатии на кнопку «Войти» соответственно проверяется совпадение текста элементов «login» и «password» со строками «Student» и «S123456s». При несовпадении текста одного из элементов в полосу «statusBar» выводится строка, информирующая пользователя о неверно введенных данных – «Ошибка авторизации!!!». При верной авторизации пользователю открывается окно-управление списком – метод описан в приложении 3.

## Чтение файла Excel

Получение данных из ячеек Excel документа осуществляется при помощи COM (Components object model) объектов, организованных в библиотеке «QAxObject», для подключение которой в файле с расширением «.pro» необходимо добавить строку «QT += axcontainer». Чтение осуществляется согласно следующему принципу: создается объект для управления Excel, выбирается необходимый файл, указывается необходимый лист Excel, устанавливаются флаги отображения файла (запрос на подтверждение действий, отображение документа). Считывается рабочий диапазон (ячейки, в которых лежат данные), при помощи которого вычисляется количество строк и столбцов Excel таблицы. Оперируя указанными выше данными создаются два цикла, которые позволяют внести данные с ячеек в двусвязный список. При завершении циклов мы закрываем файл и завершаем выполнение управления Excel листом. Работа с внешними файлами затрагивается в приложении 4.

## Открытие файла Excel

Открытие файла Excel происходит подобно предыдущему пункту, но вместо циклов с перебором ячеек необходимо перебрать все элементы двусвязного списка и записать их в Excel-файл, предварительно очистив все строки документа. Таким образом мы сохраняем данные в файл и после предоставляем его к просмотру пользователя.

## Запись данных в таблицу

Для того, чтобы отобразить данные пациентов в виде таблице, необходимо воспользоваться библиотеками «QTableWidget» и «QTableItem». В начале работы метода начинается настройка таблицы, в частности это настройка границ (ui->patientsTable->setFixedSize(915, 600), ширины ячеек (ui->patientsTable->setColumnWidth(i, 150) и создание оглавления столбцов (ui->patientsTable->setHorizontalHeaderLabels(horzHeaders), где horzHeaders – объект класса QStringList, хранящий в себе строки-оглавления каждого столбца, описание – приложение 5.

После инициализации таблицы программа приступает к перебору элементов списка, в каждой итерации цикла в таблицу добавляется строка, создается 6 новых элементов класса QTableWidgetItem (они хранят определенные поля списка) и значения устанавливаются в определенный столбец созданной таблицы (ui->patientsTable->setItem(ui->patientsTable->rowCount()-1, 5, tmpTbl)).

## Конвертация типов данных

Во время работы с данными (создание списка, запись в таблицу, запись в файл) происходит проблема несовместимости типов данных. Опишем данную проблему более подробно.

Во время чтения файла Excel программа оперирует данными типа «QVariant», которые использует и распознает в процессе своей работы Excel-документ. Для чтения этих данных и последующего создания двусвязного списка требуется конвертация данных «QVariant» в «QString», поскольку конструктор двусвязного списка принимает в качестве аргументов такие типы данных, как «QString», «int», «float». Для конвертации «QVariant» в «QString» необходимо воспользоваться встроенным методом класса «QVariant»: value.toString(), аналогично для целочисленных значений и чисел с плавающей точкой были использованы методы toInt(), toFloat(), но как быть с датой? Нам необходима строка в формате «дд.мм.гггг» – согласно установленными нами условиями. Необходимо определенные ячейки, содержащие значения с датой конвертировать в промежуточную переменную «date» типа «QDate» при помощи встроенного метода value.toDate(), «вытянуть» из данной переменной значения дня, месяца, года при помощи встроенных методов библиотеки «QDate» .date(), .month(), .year() и «собрать» необходимую нам строку: strDate = QString::number(date.day()) + "." + QString::number(date.month()) + "." + QString::number(date.year()) и только потом передавать её конструктору создаваемого списка. Работа с типами данных упоминается в приложении 6.

Во время записи в файл Excel нам необходимо конвертировать наши данные в тип «QVariant». Ошибок при передачи типа данных «QString» не возникает, поскольку функция установки значения принимает данный тип. Для примера конвертируем строковый тип данных в тип «QVariant». Для этого необходимо прописать команду qvariant\_cast<QVariant>(yourString). Для записи в Excel документ роста и веса пациента необходимо тип «int» и «float» конвертировать в строку. Для int: QString::number(visibleElement->getHeight()), для float: QString::number(visibleElement->getWeight(), 'f', 1)) – округление до десятых.

## Проверка входной строки

В процессе добавления и удаления пациентов из двусвязного списка вытекает требование подвергать входные данные проверкам. Наличие ошибок во входных данных помогает библиотека «QRegExp» - регулярные выражения. Для проверки ФИО пациента необходимо проверить, есть ли в строке буквы и спецсимволы ([\\d\\W](file:///\\d\\W)), при положительном исходе функция yourRegExp.indexIn(yourStr) вернет начальную позицию вхождения регулярного выражения в строку, в отрицательном исходе будет возвращено значение -1. Для даты минимальная защита в виде регулярного выражения выглядит так: "[0-9]{1,2}\\.[01]{1}[0-9]{1}\\.((20)|(19)){1}[0-9]{2}" – дд.мм.гггг (где год рождения от 1900 до 2099).; для веса "[0-9]{1,3}\\.{0,1}[0-9]{0,2}$", для роста "[0-9]{2,3}". Также ограничения для роста варьируются от 45 до 250, для веса от 2 до 350.

Аналогичная проверка в виде регулярных выражений присутствует и при удалении пациентов.

В случае неверного ввода данных пользователь получает уведомление с пояснением, в каком поле ввода данных присутствует опечатка. Данный метод работы осуществляется при помощи библиотеки «QMessageBox». Вызывается окно, информирующее пользователя об ошибке и её типе. Например, при неверном написании имени пациента (наличие цифр, спецсимволов) вызывается следующий метод: QMessageBox::information(this, "Ошибка ввода", "Ошибка в имени пациента").

## Класс mainwindow.h

Данный класс формы выступает в роли меню программы. Отсюда есть доступ к информации для пользователя, разработчика, вход для управления данными списка, отображение Excel-документа, выход из программы. Данный класс – подготовка программы к работе с данными: считывается Excel-файл, данные переносятся в двусвязный список, объявляются сигналы и слоты для перехода между окнами.

## Класс management.h

Класс формы представляет собой окно с возможностью управления данными списка и возвращения к главному окну (меню программы). В классе определяются сигналы и слоты для перехода к окнам отображения таблицы пациентов, добавлению пациента, его удалению и отображения Excel-документа.

## Класс patientLinkedList.h

Данный класс комбинирует в себе реализацию двусвязного списка, отрисовку окна с таблицей пациентов и работу по сохранению данных в Excel лист. Из методов двусвязного списка представлены следующие: push\_back(добавление элемента в конец списка), push\_front(добавление элемента в начало списка), pop\_back (удаление последнего элемента), pop\_front (удаление первого элемента), printList (занесение листа в таблицу), showList (показать Excel лист), cntPatient (метод возвращает количество пациентов + 1), showTableSlot (вызывает метод printList() и отображает форму, хранящую в себе таблицу пациентов), testList (отображает в qDebug() фамилии пациентов).

## Класс addPatient.h

Класс отвечает за отображение формы с полями ввода данных нового пациента, проверяет корректность данных при помощи регулярных выражений. При верно введенных значениях добавляет в конец списка нового пациента.

## Класс delPatient.h

В классе реализована форма с выбором критерия удаления пациента: ФИО, рост, вес, дата рождения. При нажатии на определенную кнопку осуществляется переход к следующему окну, где пользователю предлагается ввести данные для удаления пациента. Ввод данных контролируется регулярным выражением. Если несколько пациентов имеют одинаковый вес, то они будут выписаны из списка. Аналогичная схема удаления происходит и с другими параметрами. Удаление осуществляется следующим образом: list->delByFio(inSurName, inName, inSecondName), где вместо предложенных переменных используется обращение к тексту определенных label-элементов виджета, например (ui->surname->text()).

## Классы fio.h, height.h, weight.h, yearOfBirth.h

Данные формы классов открываются при вызове с окна класса формы «delPatient.h». В них реализованы проверки регулярным выражением, отображение предупреждений при наличии ошибок, обращение к методу удаления из списка по введенному значению.

# Проект программного средства

Проект пользовательского интерфейса представлен в виде схемы ниже (Рисунок 1). Данное представление программы позволяет визуально понять переход между формами и действия, выполняемые в них.

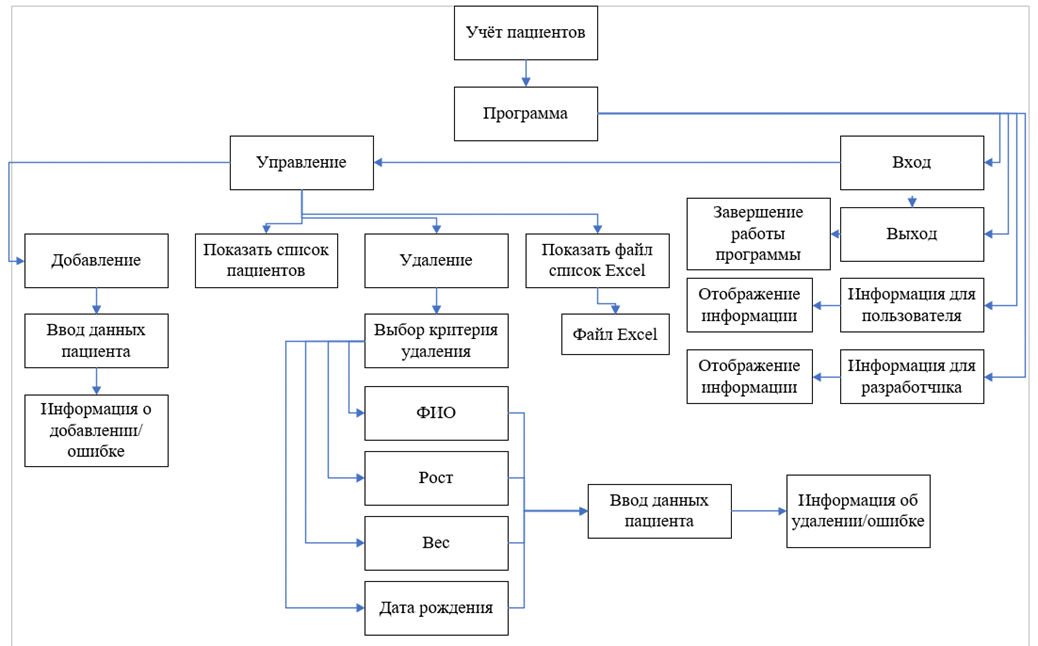


Рисунок 1 – Схема пользовательского интерфейса программы.

Изобразим схемы алгоритмов двусвязного списка, в частности методов push\_back (Рисунок 2), push\_front (Рисунок 3), pop\_back (Рисунок 4), pop\_front (Рисунок 5), printList (Рисунок 6).

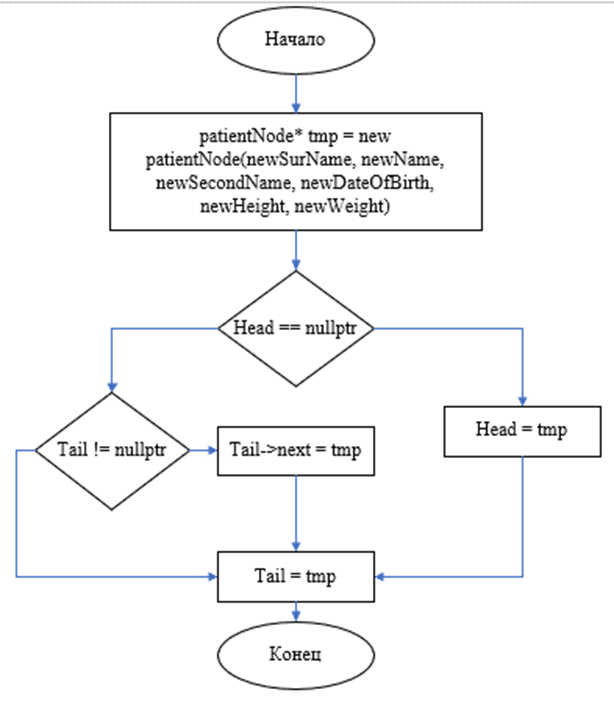


Рисунок 2 – Метод push\_back двусвязного списка.

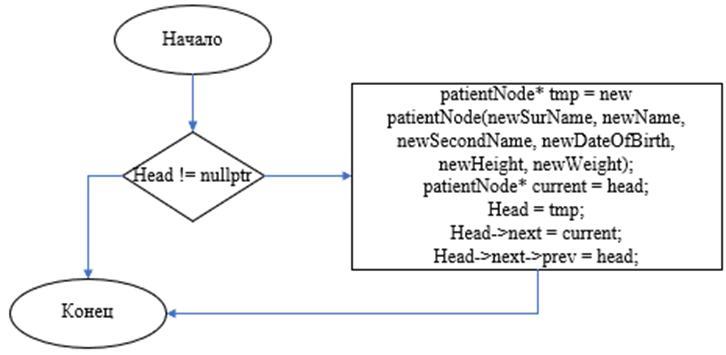


Рисунок 3 – Метод push\_front двусвязного списка.

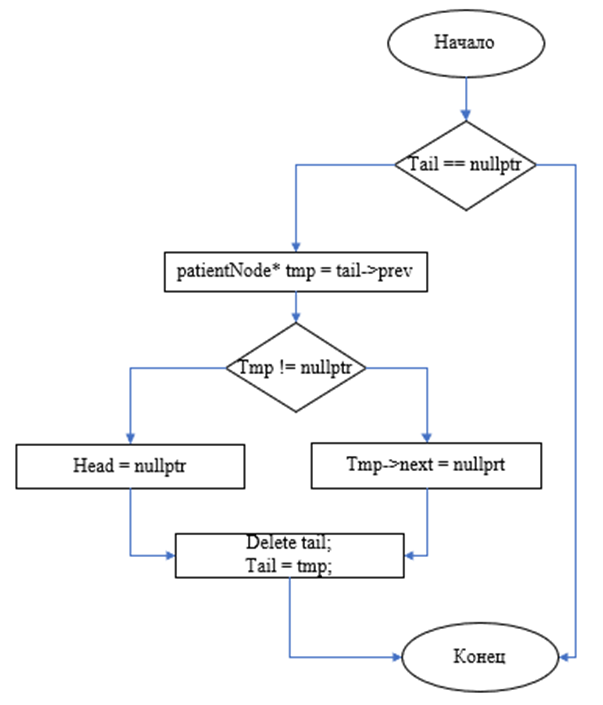


Рисунок 4 – Метод pop\_back двусвязного списка.

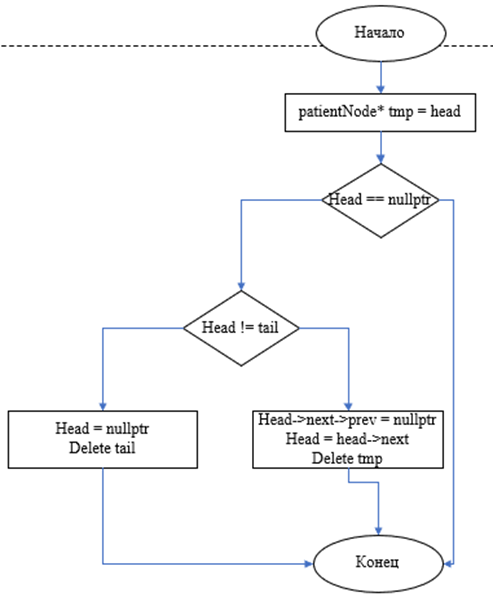


Рисунок 5 – Метод pop\_front двусвязного списка.

В двусвязном списке большую роль играют методы работы удаления элементов, которым программа динамически выделяет память. Представим алгоритм одного из таких методов ниже (Рисунок 6). Диаграмма классов – рисунок 7

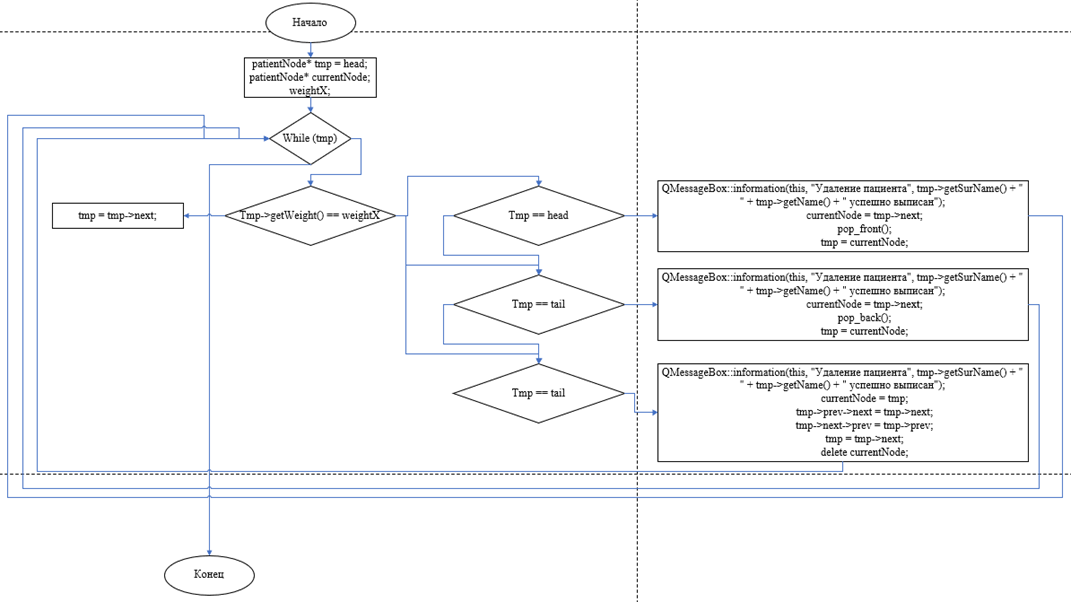


Рисунок 6 – Удаление элемента delByWeight.

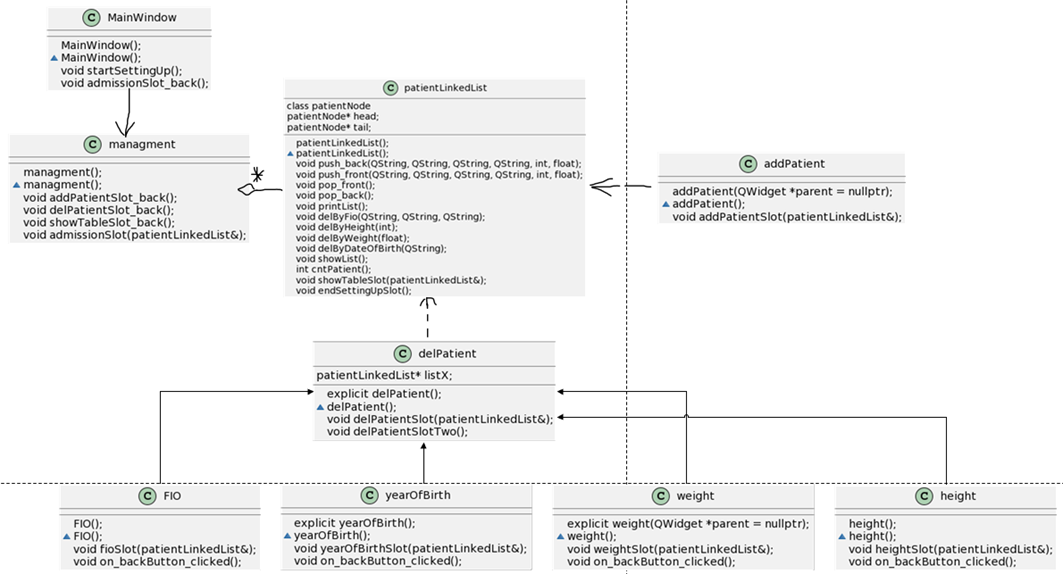


Рисунок 7 – Диаграмма классов.

# Сценарий тестирования программного средства

Запустим программу. Появляется главное окно с заголовком «Учёт пациентов». Слева сверху доступно меню для ознакомления с программой (Рисунок 8, 9).



Рисунок 8 – Запуск программы.



Рисунок 9 – нажатие на вкладку «Программа».

Нажмем на действие «Войти». На главном окне появится окно авторизации пользователя (Рисунок 10).

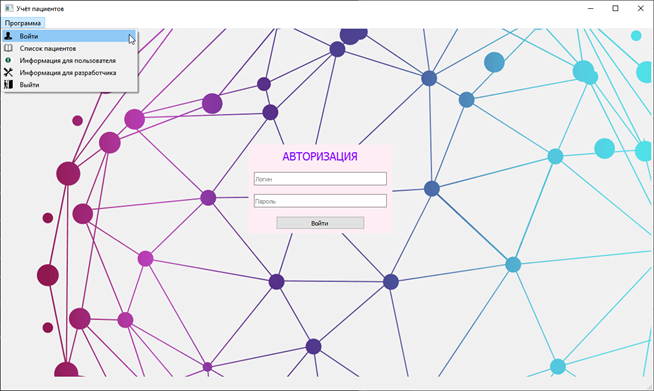


Рисунок 10 – Авторизация пользователя.

Запустим действие «Список пациентов», откроется Excel-документ с данными о пациентах (Рисунок 11).

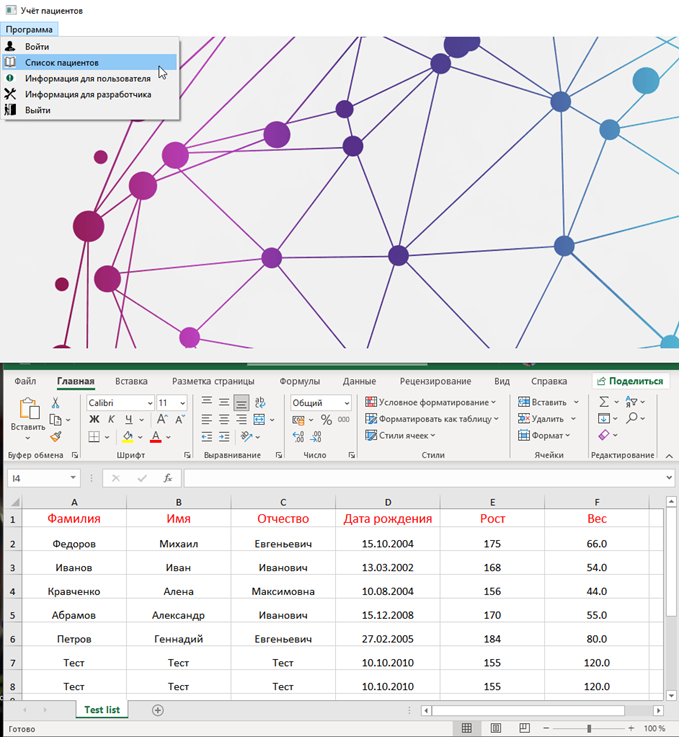


Рисунок 11 – Excel документ картотеки пациентов.

Нажмем на кнопку «Информация для пользователя». На главном окне появится поле с данными из текстового файла, содержащем описание работы с программой (Рисунок 12).

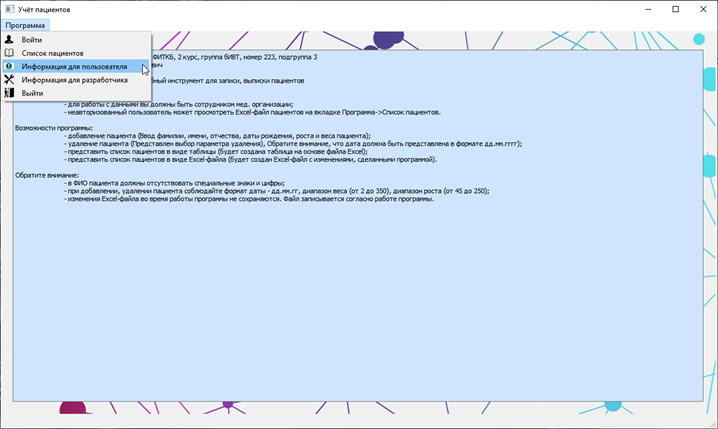


Рисунок 12 – Информация для пользователя.

Действие «Информация для разработчика» также откроет файл с соответствующей информацией (Рисунок 13).

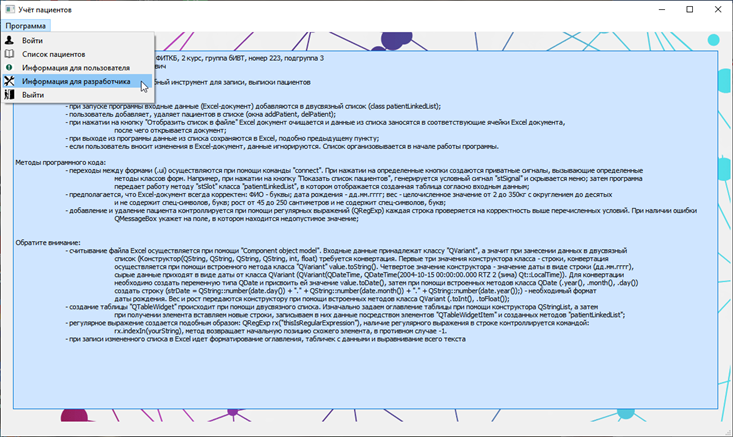


Рисунок 13 – Информация для разработчика.

Попробуем авторизоваться в окне «Авторизация». Попробуем ввести неверные данные, на главном окне в «statusBar» появится информация об ошибке входа (Рисунок 14).

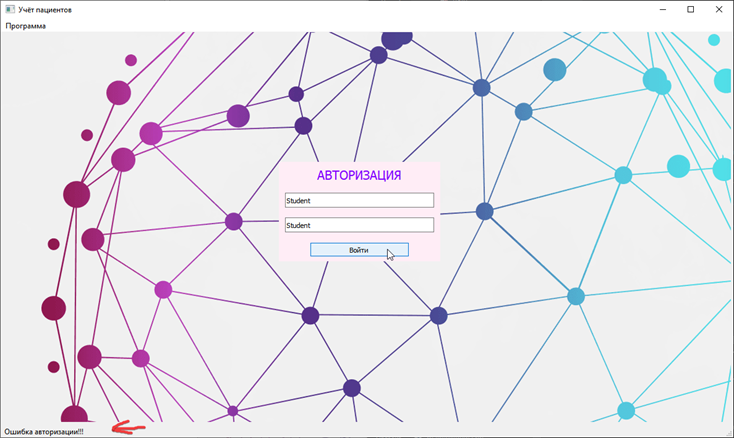


Рисунок 14 – Неправильный ввод данных в поле авторизации.

При вводе корректного логина и пароля будет осуществлен переход к окну управления списком пациентов (Рисунок 15, 16).

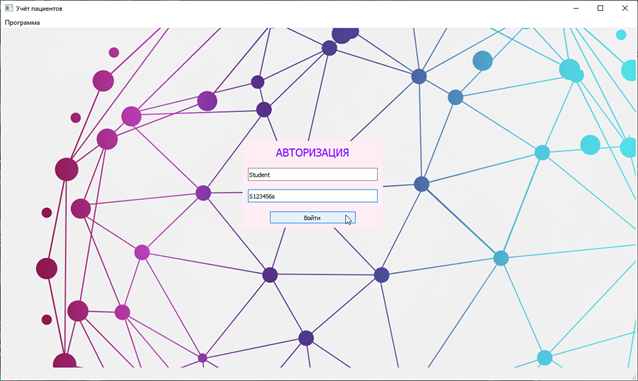


Рисунок 15 – Корректный ввод данных.

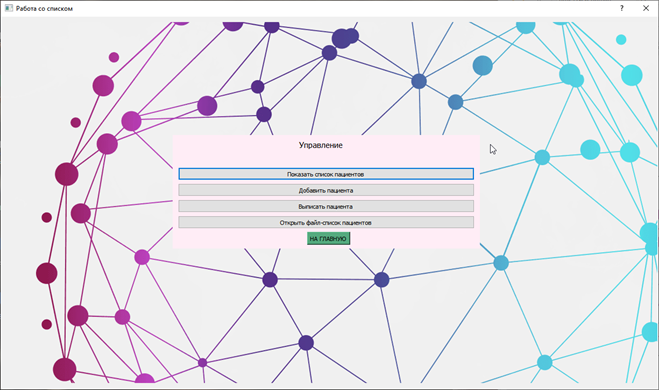


Рисунок 16 – Переход к окну «Работа со списком».

Нажмем на кнопку «Показать список пациентов». Будет осуществлен переход к новому окну, в котором создается таблица согласно данным двусвязного списка (Рисунок 17).

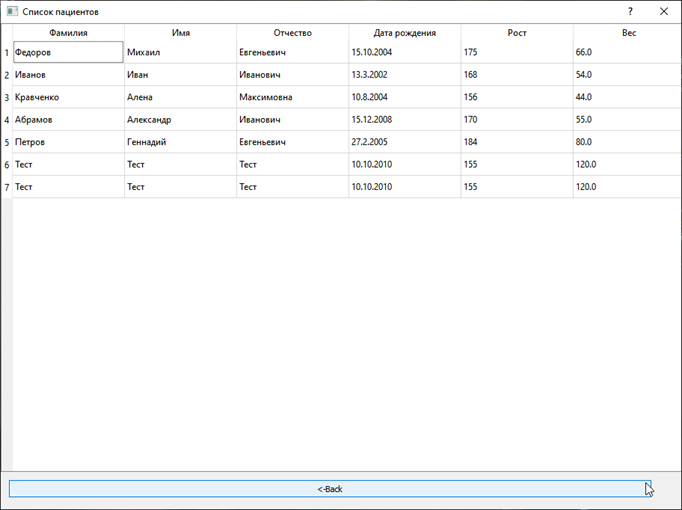


Рисунок 17 – Таблица пациентов.

Вернемся к окну «Управление» и выберем пункт «Добавить пациента». Будет осуществлен переход к новому окну, где пользователю предлагается ввести данные нового пациента в соответствующие поля. При неверном вводе будет выведена информация, в каком месте пользователь допустил ошибку (Рисунок 18). При правильном вводе информации пользователь будет уведомлен о добавлении пациента: «Фамилия» + «Имя» (Рисунок 19, 20).

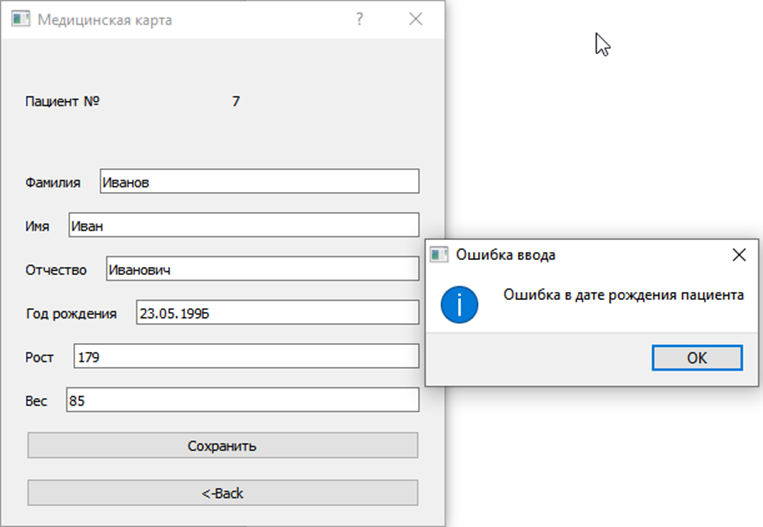


Рисунок 18 – Неправильный ввод даты рождения.

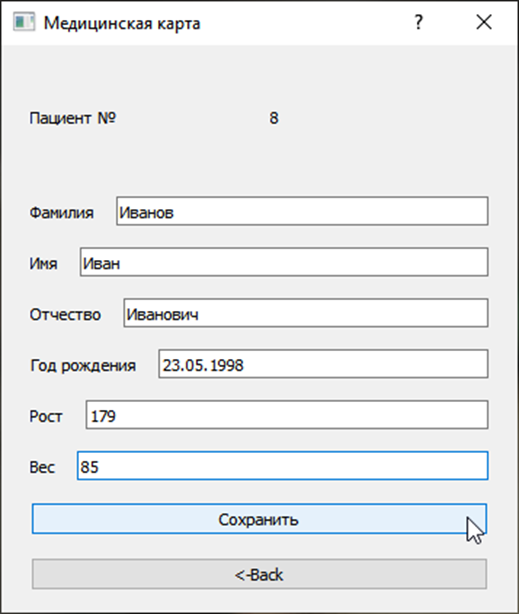


Рисунок 19 – Правильный ввод данных пациента

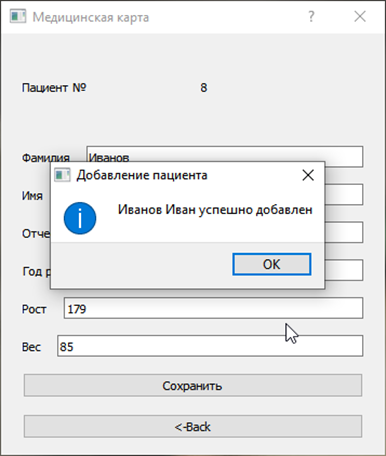


Рисунок 20 – Правильный ввод данных пациента

Проверим таблицу и Excel-файл после добавления пациента. Откроем таблицу пациентов (Рисунок 21).

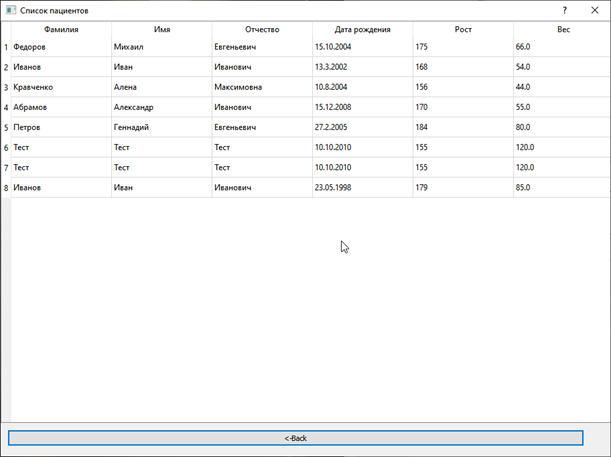


Рисунок 21 – Таблица после добавления пациента.

Теперь отобразим Excel-документ (Рисунок 22). Перед открытием файла данные «выгружаются» из двусвязного списка и перезаписывают каждую ячейку, форматируют их.

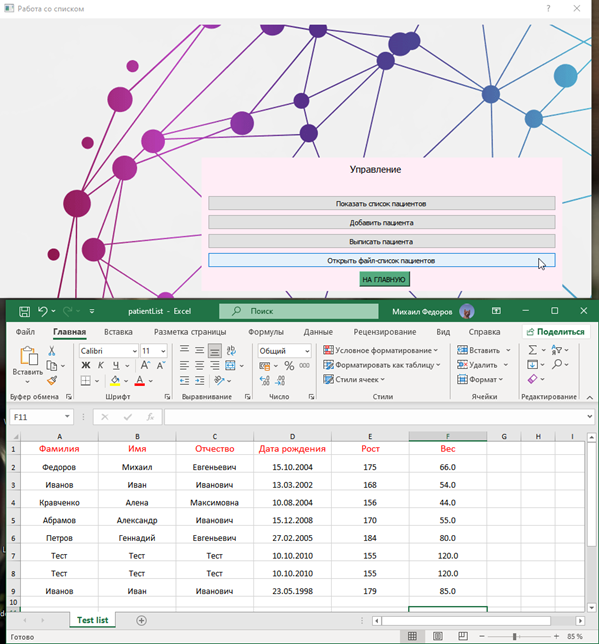


Рисунок 22 – Excel после добавления пациента

Попробуем удалить пациента согласно критерию поиска «Рост» (Рисунок 23, 24, 25). Для данного метода работы в таблицу были добавлены 2 пациента «Тест», имеющих одинаковый рост. Результат представлен в виде таблицы и файла Excel (Рисунок 26, 27).

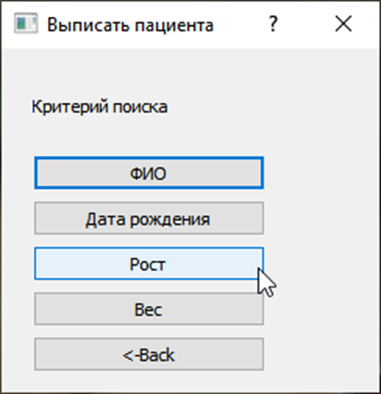


Рисунок 23 – Выбор критерия удаления.

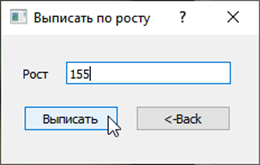


Рисунок 24 – Ввод данных.

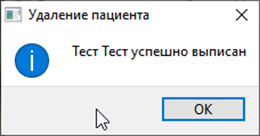


Рисунок 25 – Результат удаления пациента.

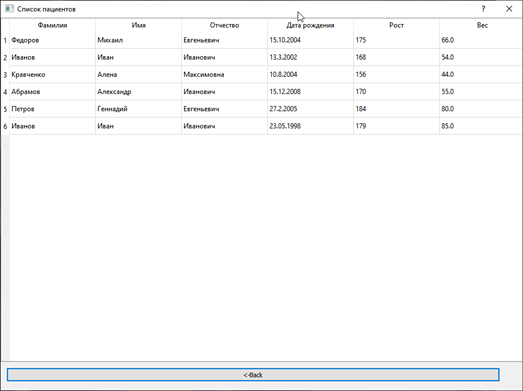


Рисунок 26 – Таблица после удаления пациента.

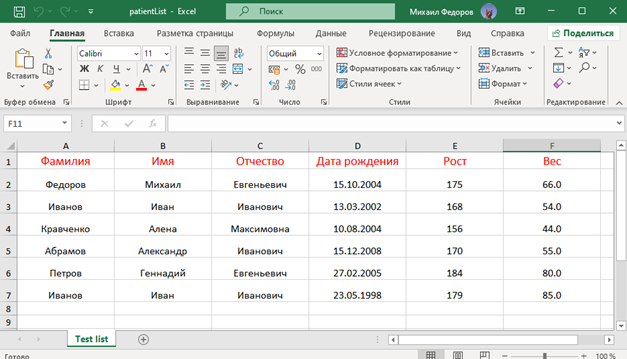


Рисунок 27 – Excel файл после удаления пациента.

Завершим работу программы. Для этого на окне «Учет пациентов» или «Работа со списком» нажмем крестик в правом верхнем углу (Рисунок 20). Будет предложено выйти или остаться в программе. При выходе данные сохраняются в Excel файл. Об этом будет написано в консоли программы (Рисунок 21).

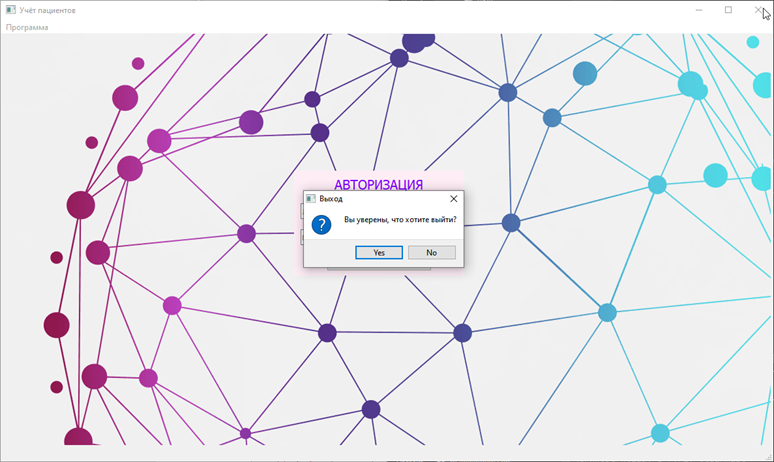


Рисунок 20 – Завершение работы программы.

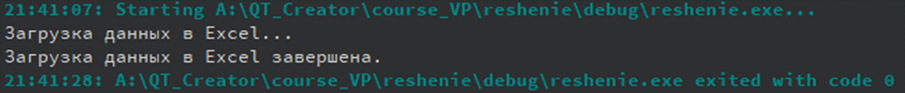


Рисунок 21 – Консоль при завершении работы.

# Руководство пользователя

Данная программа представляет собой удобный инструмент для записи, выписки пациентов

Вход:

* для работы с данными вы должны быть сотрудником мед. организации;
* неавторизованный пользователь может просмотреть Excel-файл пациентов на вкладке Программа->Список пациентов.

Возможности программы:

* добавление пациента (Ввод фамилии, имени, отчества, даты рождения, роста и веса пациента);
* удаление пациента (Представлен выбор параметра удаления), Обратите внимание, что дата должна быть представлена в формате дд.мм.гггг;
* представить список пациентов в виде таблицы (будет создана таблица на основе файла Excel);
* представить список пациентов в виде Excel-файла (будет создан Excel-файл с изменениями, сделанными программой).

Обратите внимание:

* в ФИО пациента должны отсутствовать специальные знаки и цифры;
* при добавлении, удалении пациента соблюдайте формат даты - дд.мм.гг, диапазон веса (от 2 до 350), диапазон роста (от 45 до 250);
* изменения Excel-файла во время работы программы не сохраняются. Файл записывается согласно работе программы.

# Заключение

В рамках курсового проекта были изучены инструменты разработки интегрированной среды программирования Qt Creator. Работа опирается на работу таких встроенных библиотек, как QTableWidget, QTableItem, QFile, QAxObject, QDebug, QtGui, QCloseEvent, QMessageBox, QRegExp. В процессе разработки программного обеспечения исследовано взаимодействие Qt с текстовыми файлами и excel документами, создание таблицы и улучшены навыки работы с разделением и реализацией файлов, динамическим выделением памяти.

Ссылка на проект github: <https://github.com/finesko1/SVP>

# Список литературы

1. Шлее М. Профессиональное программирование на С++. / Шлее М.— СПб.: БХВ-Петербург, 2018. — 1072 с,ил.
2. Мейер Б. Почувствуй класс.: учебное издание. / Б. Мейер; Перевод под ред. В.А. Биллига. — М.: Национальный Открытый Университет «ИНТУИТ»: БИНОМ. Лаборатория знаний, 2011. —775 с.
3. Иванова Г.С, Ничушкина Т.Н., Пугачев Е.К. И21 Объектно-ориентированное программирование: Учеб. для вузов/ Под ред. Г.С. Ивановой. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2001. - 320 с, ил.
4. Лафоре Р. Л29 Объектно-ориентированное программирование в С++ / Р. Лафоре - 4-е изд. – СПб.: Санкт-Петербург,2011 – 928с.: ил.
5. Колесов, Ю. Б. Моделирование систем. Объектно-ориентированный подход / Ю.Б. Колесов, Ю.Б. Сениченков. - М.: БХВ-Петербург, 2006. - 192 c.
6. Саммерфилд М. Qt. Профессиональное программирование. Разработка кроссплатформенных приложений на С++. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 560 с.