**2 Проектирование программного продукта**

**2.1 Формирование структуры проекта**

Разработка алгоритма функционирования программного продукта проводилась нисходящим методом, который заключается в том, что разрабатываемый алгоритм разделяется на дочерние алгоритмы, а те, в свою очередь, разделяются на элементарные алгоритмы. Структуру проекта можно представить в виде двух блоков.

В первом блоке выполняется запуск программы, в ходе которого происходит запуск формы с игрой и появляется возможность играть в «Sudoku».

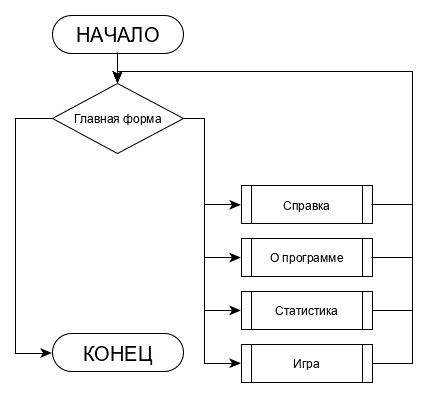
Во втором блоке появляется возможность просмотра статистики (таблицы рекордов).

Данный программный продукт, «Sudoku», состоит из трёх составляющих частей:

* трех файлов, содержащих внутри себя статистику о количестве прохождений «Sudoku» по трём уровням сложности: лёгкий, средний, сложный,
* приложения, разработанного в среде программирования «QT Creator»,

## **2.2 Описание схемы программы**

Схема программы представлена на рисунке 2.1.



**Рисунок 2.1 – Схема работы игрового приложения «Судоку»**

После запуска программы появляется главная форма, на которой можно: перейти к справке, посмотреть информацию о программе, перейти к просмотру статистики и начать игру (рисунок 2.2).

|  |
| --- |
| Выход  О приложении  Как играть?  Статистика  Начать игру  Название |

**Рисунок 2.2 – Макет главного меню**

После перехода по кнопке «О приложении» открывается новое окно с информацией о приложении (рисунок 2.3).

Информация о приложении

**Рисунок 2.3 – Макет окна «О приложении»**

После перехода по кнопке «Как играть?» открывается новое окно с информацией об игре (рисунок 2.4).

|  |
| --- |
| Правила игры судоку |

**Рисунок 2.4 – Макет окна «Как играть?»**

После перехода по кнопке «Статистика» открывается окно со статистикой, разделенной по сложностям (рисунок 2.5).

|  |
| --- |
| Лучшее сыгранное время   1. 00:00:00 2. 00:00:00 3. 00:00:00   Сложная  Средняя  Лёгкая |

**Рисунок 2.5 – Макет окна «Статистика»**

После перехода по кнопке «Начать игру» открывается окно с выбором сложности для начала игры (рисунок 2.6).

|  |
| --- |
| Главное меню  Сложная  Средняя  Лёгкая  Выбор сложности игры |

**Рисунок 2.6 – Макет выбора сложности игры**

После перехода по кнопкам «Лёгкая», «Средняя» или «Сложная» открывается окно с игровым полем (рисунок 2.7).

|  |
| --- |
| 00:00:00  Проверка на правильность  Выход  Вернуться назад  подсказка  Игровое поле |

**Рисунок 2.7 – Макет игрового поля**

После заполнения игрового поля и нажатия кнопки «Проверка на правильность» происходит вызов нового окна с сообщением о завершении (в случае правильного заполнения) или сообщением об ошибке в решении (в случае неправильного заполнения) (рисунки 2.8-2.9).

|  |
| --- |
| Время за которое было решено судоку  Выбранная сложность  Поздравляю! Вы смогли завершить решение судоку!  Сложность:  Время: |

**Рисунок 2.8 – Макет сообщения о завершении решения**

|  |
| --- |
| Судоку решено неверно!  Ок |

**Рисунок 2.9 – Макет сообщения об ошибке в решении**

**2.3 Среда разработки «Qt Creator»**

Для разработки интерфейса программного продукта использовалась среда программирования «Qt Creator». Qt Creator - это кроссплатформенная среда разработки, ориентированная на нужды Qt разработчиков.

Среда программирования «Qt Creator» объединяет в себе высокопроизводительный компилятор языка «С++», средства наглядного создания программ.

Основное назначение среды программирования «Qt Creator» –использование ее как средство для быстрого создания широкого класса «Windows»-приложений. Она учитывает многие новейшие достижения в программировании и практике создания программных приложений.

Qt Creator фокусируется на предоставлении возможностей, которые помогут новичкам быстро оcвоить Qt и начать с ним работать, а также повысить продуктивность опытных Qt разработчиков.

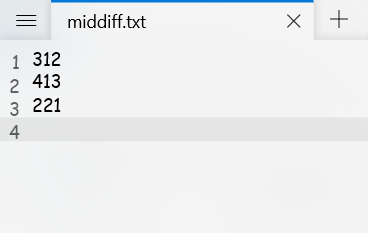
* Редактор кода с поддержкой C++, QML и ECMAscript
* Средства эффективной навигации по коду
* Подсветка синтаксиса и автодополнение кода
* Статическая проверка кода и подсказка стиля при вводе
* Поддержка рефакторинга
* Контекстно-зависимая помощь
* Сворачивание кода
* Режимы выделения и автоподстановки скобок[1]

Qt используется для разработки графических пользовательских интерфейсов (GUI) и многоплатформенных приложений, которые работают на всех основных настольных платформах и большинстве мобильных или встроенных платформ. Большинство программ GUI, созданных с помощью Qt, имеют естественный интерфейс, и в этом случае Qt классифицируется как инструментарий виджетов . Также можно разрабатывать программы без графического интерфейса пользователя, такие как инструменты командной строки и консоли для серверов. Примером такой программы без графического интерфейса, использующей Qt, является веб-фреймворк Cutelyst .

Qt поддерживает различные компиляторы, включая компилятор GCC C ++, набор Visual Studio , PHP через расширение для PHP5, и имеет обширную поддержку интернационализации . Qt также предоставляет Qt Quick , который включает декларативный язык сценариев под названием QML, который позволяет использовать JavaScript для обеспечения логики. Благодаря Qt Quick стала возможной быстрая разработка приложений для мобильных устройств, в то время как логика по-прежнему может быть написана с использованием собственного кода для достижения максимально возможной производительности. Другие функции включают в себя SQL доступа к базе данных, XML синтаксического анализа, JSON синтаксического анализа, нити управления и поддержку сети[2].

**2.4 Хранение данных**

Для хранения данных в курсовом проекте используются три текстовых файла, которые могут быть открыты с помощью программы «Блокнот» (пример использования представлен на рисунке 2.10).



**Рисунок 2.10 – Пример хранения данных в блокноте**

Текстовый файл – компьютерный файл, содержащий текстовые данные.

Преимущества:

* универсальность,
* устойчивость,
* формат текстового файла можно легко изменить.