**Санкт-Петербургский государственный университет**

**Факультет технической кибернетики**

**Кафедра компьютерных систем и программных технологий**

**Курсовая работа**

**Тема работы: «Разработка приложения с графическим интерфейсом на языке C++ с использованием библиотеки QT»**

**Разработал: Киселёв Антон, группа 1081/3**

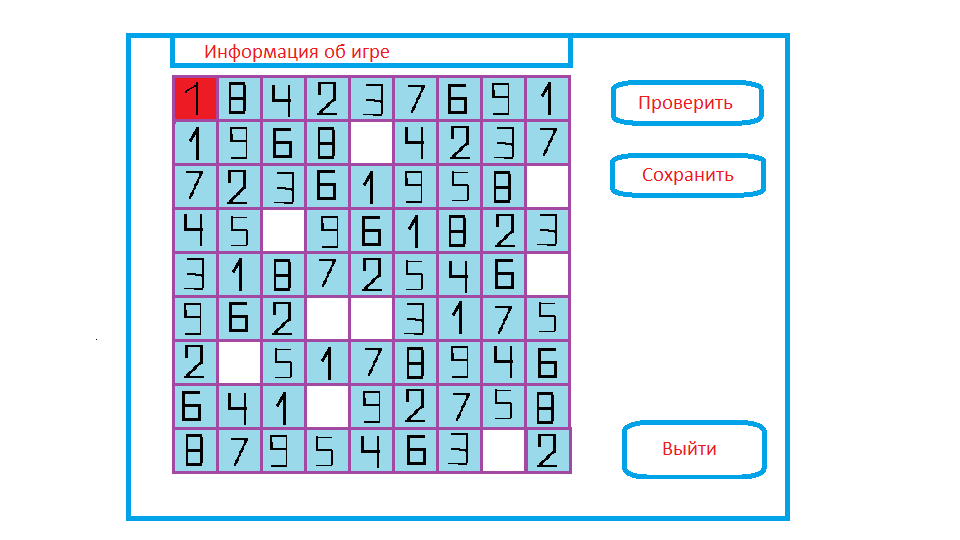
**Преподаватель: доцент Пышкин Е. В.**

**Санкт-Петербург 2012г**

1. **Образ и границы проекта.**

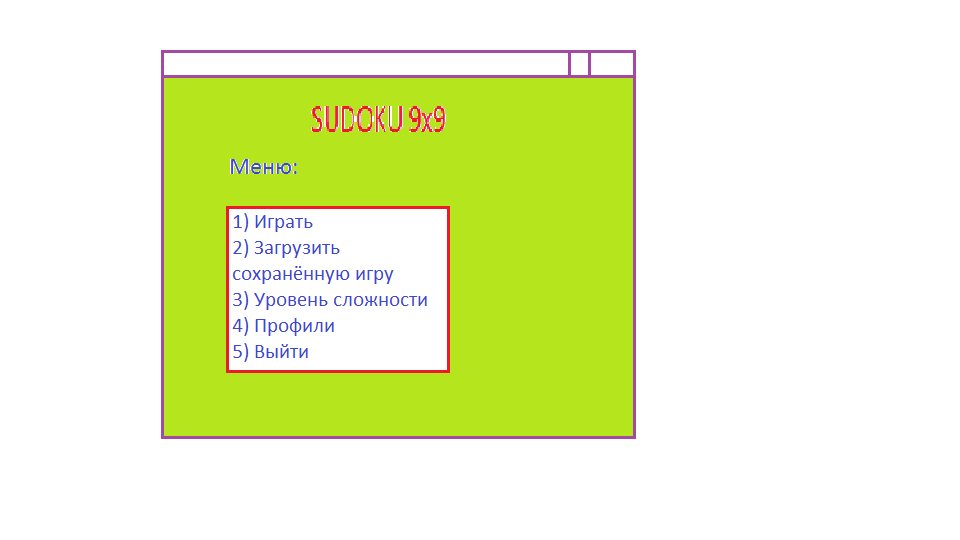
**Проект посвящен созданию популярной логико-арифметической игры «Sudoku». Суть игры заключается в следующем. Перед игроком выводится игровое поле размера 9 на 9 клеток, в некоторых клетках которого поля пропущены цифры. Цель игрока расставить недостающие цифры по полю так, чтобы цифры не повторялись в строке, столбце и квадрате 3 на 3 поля. Во время игры пользователь может автоматически проверить свою расстановку цифр. Если сделано правильно, то игрок победил.**

**На рисунке изображена примерная композиция поля.**

****

**На данном рисунке пропуски в поле - это недостающие цифры. Предположим, игрок заполнил левую верхнюю клетку единицей, то при проверке, оказывается, что такая цифра уже есть и определитель ошибок обозначит данную клетку красным.**

**На следующем рисунке представлено примерный вид приложения при запуске.**

****

**Предполагается поддержка следующих основных сценариев работы приложения:**

**А) Загрузка из файла начального состояния игрового поля.**

**Начальное состояние будет загружено из текстового файла для дальнейшей генерации.**

**Б) Генерация начального состояния.**

**После загрузки программа сгенерирует поле в необходимый вариант, то есть расставит все цифры в правильном порядке во избежание неединичных вариантов расстановки цифр пользователем.**

**В) Выполнение игроком ходов и контроль их непротиворечивости.**

**Игроку будет дана возможность свободной расстановки цифр в поле и автоматической проверки поля на ошибки. Предполагается, что готовое поле будет сохраняться в памяти, а игроку будет выдано поле с пропусками (начальное состояние). Подробный алгоритм взаимодействия рассмотрен в пункте 5.**

**Г) Сохранение текущего состояния игры.**

**Предполагается создание специальной папки для сохранения игр (документа). В неё будет помещаться номер игры, название, дата, готовое поле и поле, незаконченное игроком.**

**Д) Загрузка ранее сохранённой игры.**

**При запросе от пользователя какого-нибудь сохранения, будет происходить открытия папки с сохранёнными играми и поиск необходимой игры. Будет происходить считывание данных и вывод их на экран перед пользователем.**

**Предметом разработки являются два слоя приложения: бизнес-логика и графический интерфейс.**

**Слой бизнес-логики обеспечивает составления игрового поля и выведения его она экран. Включает в себя следующие классы:**

* **Игровое поле. Состоит из операций: построение поля, вывести поле. А также поле будет предоставляться на обработку другим методам и функциям.**
* **Строка (вспомогательный класс поля). Будет использован при генерации поля.**
* **Столбец (вспомогательный класс поля). Будет использован при генерации поля.**
* **Квадрат 3 на 3. Будет использован при генерации игрового поля.**
* **Ячейка игрового поля. Будет считывать значение и отдавать его графическому интерфейсу для вывода на экран.**

**Слой графического интерфейса будет обеспечивать взаимодействие с пользователем. Включает в себя следующие классы:**

* **Графическое игровое поле. Будет выводить созданное игровое поле на экран.**
* **Игровой фон. Создает внешний облик приложения.**
* **Классы консольной панели. Добавляют различные кнопки к приложению.**

**Также предусмотрено написание тестовых классов, которые будут тестировать всю программу.**

1. **Техническая спецификация.**

**Требования к консольному интерфейсу:**

**Консольный интерфейс должен показывать нам правильность генерации игрового поля, показывать, как будет происходить вставка цифр в ячейки игрового поля.**

**Требования к графическому интерфейсу:**

**Графический интерфейс должен обеспечивать взаимодействие приложения с пользователем на доступном (понятном) уровне. Обеспечить вывод игрового поля в цветах и основных функций в виде кнопок.**

1. **Описание системной архитектуры.**

**Дадим описание объектам бизнес логики.**

**Игровое поле будет сначала считано из файла для дальнейшей генерации. На основе требований объектов строка, столбец и квадрат будет производиться полное заполнение числами игрового поля. Во время игры поле будет изменяться при каждом заполнении. А при подаче команды на проверку ошибок будет сравниваться с готовым игровым полем. Три других объекта строки, столбца и квадрата будут вызываться при генерации поля.**

1. **Графический интерфейс.**

**Объекты графического интерфейса – это игровой фон, игровое поле, кнопки, окна, то есть все, что добавит дизайна, удобства и сформирует высокоуровневое приложение.**

1. **Методы взаимодействия с пользователем.**

**Моё приложение (игра SUDOKU) состоит из двух частей. Первая часть – это бизнес-логика, которая включает в себя конструктор игрового поля и структуру сохранённых игр. Вторая часть - Графический интерфейс - состоит из классов создания графического игрового поля, консольной панели, игрового меню. Пользователь будет взаимодействовать с интерфейсом, а бизнес-логика будет вызываться внутри интерфейса. Ей будут подаваться данные, запросы, которые будут обрабатываться, и дальше она будет выводить результаты обработки. То есть при вызове определённого запроса, например, построение игрового поля, команда будет подана функциям логики построения игры, после создания игрового поля, оно будет передаваться функциям графики.**

**Следующие виды взаимодействия между частями приложения.**

**1) Создание игрового поля.**

**При подаче команды от пользователя сгенерировать игровое поле, сначала поле будет сформировано классами логики. После автоматически создадутся 81 объект клетка игрового поля, каждому объекту присвоится соответствующее значение из заданного поля. Затем объект клетка передастся на обработку графического интерфейса, в котором создастся 81 экземпляр графическая клетка, с соответствующим значением из предыдущего объекта клетка. Далее будет формироваться графическое игровое поле, в которое будут помещаться 81 объект клетка.**

**2) Сохранение текущей игры.**

**При подаче команды от пользователя сохранить игру, запрос отправиться на обработку соответствующим функциям сохранения игры. В созданном файле запишется соответствующий номер сохранения игры, дата, имя игры и массив чисел. При загрузке игры данные из файла будут направлены на обработку графическому интерфейсу для создания графического игрового поля и выведения его на экран.**

**3) Определение победы игрока.**

**Пользователь нажмёт на панели команду проверить поле. Запрос будет отправлен на обработку в бизнес-логику. При этом текущее поле будет сравниваться с правильным полем, сохранённым в игре. Ошибки будут указывать на соответствующие клетки, которые будут выделены на поле красным как ошибки, и пользователь получит возможность их исправить.**