**Санкт-Петербургский государственный политехнический университет**

**Факультет технической кибернетики**

**Отчет по курсовой работе по курсу «Программирование»**

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

**Выполнил студент: Киселёв Антон, группа 1081/3**

**Руководствовал: доцент Пышкин Е.В.**

**Санкт-Петербург, 2012**

1. **Описание и задачи проекта.**

**В качестве практического задания я взял игру «Судоку». Смысл игры в заполнении пустых клеток квадратной матрицы чисел 9 на 9. В игре предусмотрено несколько различных уровней сложностей и различные варианты игрового поля. Разрабатывая данный проект, передо мной стояло несколько задач.**

**Во-первых, сгенерировать игровое поле; для этого я сформировал двумерный массив 9 на 9, который заполнял случайными числами, с учетом необходимых требований строки, столбца и квадрата – объектов поля, которые и определяют условия заполнения поля случайными числами.**

**Во-вторых, разработать игровую механику. В консоли я реализовал класс Game, который обеспечивает ходы в игре. В графическом исполнении я описал методы взаимодействия с пользователем. Графическое поле и различные окна.**

**В третьих, сделать игру в какой-то степени играбельной. Я реализовал команду на проверку игрового поля и всплывающие окна по некоторым командам.**

1. **Описание бизнес логики приложения.  
   Бизнес логика отвечает за формирование математической модели игрового поля, сохранение и загрузку данных в файл, управление игрой в консоли, включает тестовый класс. Бизнес логика включает в себя 4 класса: Field, SavedGames, Test, Game.   
   Класс Field содержит методы формирующие квадратную матрицу чисел, изъятие и вставку числа в матрице, заполнение матрицы нулями, в этом классе формируется массив данных для построения графического поля – координаты клетки и текста. Также этот класс содержит методы определения победы игрока в консоли.  
   Класс SavedGames отвечает за сохранение игрового поля в текстовом файле, а также загрузке его из файла в случае продолжения игры.  
   Класс Test проверяет все методы классов Field и SavedGames.   
   Класс Game производит игру в консоли, а именно выводит матрицу на экран, производит команды и общение с игроком.**
2. **Описание работы классов графического интерфейса.  
   Классы графического интерфейса выполняют общение с пользователем, ждут от него команд, а также отправляют запросы в классы бизнес логики и получают от них требуемые данные (матрицу чисел, сохранение игрового поля). Графический интерфейс состоит из одного класса mainwindow, в котором реализуются методы общения с пользователем и представление приложения в цветовой гамме, со всеми необходимыми окнами и диалогами. Окно состоит из игрового меню, которое включает в себя 5 действий: начать новую игру, продолжить игру, сохранить игру, проверить поле и выйти. За формирование графического игрового поля отвечает метод paintEvent. После выбора новой игры, игрок выбирает в окне уровень сложности и на экране уже нарисованное зеленое игровое поле 9 на 9 клеток, заполняется цифрами, на месте, где цифр нету – в матрице записан нуль. После чего пользователь может вставлять на место пропусков числа – правой кнопкой мыши нажимая по клетке поля. Выплывает окно ввода, куда пользователь вписывает число, после нажатия клавиши Enter или кнопки Ok цифра отображается в той же клетке поля. Чтобы проверить свои действия на ошибки – пользователю необходимо в меню выбрать метод check, который закрасит те клетки, в которых есть ошибки. Если ошибок нет, то всплывет сообщение о победе игрока.   
   При выборе save game матрица сохранится в файл game.txt, а чтобы снова продолжить играть в ту же игру необходимо выбрать load game – после чего загрузиться сохраненная матрица и поле обрисуется необходимыми числами. При сохранении появляется диалоговое окно о подтверждении сохранения.**