**Санкт-Петербургский государственный университет**

**Факультет технической кибернетики**

**Кафедра компьютерных систем и программных технологий**

**Курсовая работа**

**Архитектура приложения**

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

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

**Санкт-Петербург 2012г**

1. **Образ и границы проекта**

Проект посвящен созданию компьютерной модели популярной логическо-арифметической игры «Судоку». Суть игры состоит в …

На рис. 1 представлен пример некоторой начальной конфигурации и примеры возможных ходов в данной позиции.

Предполагается поддержка следующих основных сценариев работы приложения:

1. Загрузка из файла начального состояния игрового поля.
2. Генерация начального состояния.
3. Выполнение игроком ходов и контроль их непротиворечивости.
4. Сохранение текущего состояния игры.
5. Загрузка ранее сохраненной игры.

Предметом разработки являются два слоя приложения: слой бизнес-логики и интерфейс пользователя.

Слой бизнес-логики обеспечивает представление игрового поля и действий над ним. Основными классами на данном уровне являются следующие классы:

* Игровое поле.
* Ячейка игрового поля.
* ….

Слой взаимодействия с пользователем должен предполагать две возможных реализации: консольный интерфейс, графический интерфейс пользователя. Основная задача консольного интерфейса – обеспечить возможности отладки бизнес-логики приложения. Графический интерфейс нацелен на конечных пользователей системы.

Моё приложение состоит из двух основных взаимодействующих частей. К первой я отнесу механическую составляющую игры. Это все методы и классы, с помощью которых я и буду строить игру. Игровое поле, методы взаимодействия с пользователем, игровая механика, формирование уровня сложности игры. Механическая составляющая будет основываться на следующих классах. Класс игровое поле будет формироваться с помощью некоторого алгоритма (описано ниже). Класс игровое поле будет основан на взаимодействии классов квадрат, строка и столбец (вспомогательные классы).

Вторая составляющая это методы взаимодействия с пользователем или графический интерфейс. Будет состоять из абстрактных классов: окно, игровое меню, игровое поле. Каждый из этих классов будет срабатывать при вызове классов игровой механики. Скажем, сформировав игровое поле, соответствующий абстрактный класс поля выведется нам на экран. То есть предусматривается, что будут взаимодействовать две составляющие части программы.

1. **Техническая спецификация**

Здесь на основе положений раздела об образе и границах проекта должны быть сформулированы конкретные требования к разрабатываемым классам.

Требования к консольному интерфейсу

Требования к графическому интерфейсу

1. **Описание системной архитектуры**

Предлагается следующая модель классов бизнес-логики.

а) Field (Игровое поле);

b) Square3x3 (Квадрат 3х3);

c) Row (Строка);

d) Column (Столбец);

e) Cell (Клетка);

f) Game Level (Уровень игры);

А) Класс Игровое поле.

Игровое поле будет формироваться по своему особому алгоритму. Сначала я воспроизведу случайную генерацию чисел в этом поле (пока что подразумевается, что игровое поле – это один объект) при условии, что числа не должны повторяться в строке, столбце и квадрате 3х3 (всего 9 таких квадратов) более 1 раза. Сохраняю это поле в памяти для дальнейшего использования и сравнения с результатами игры пользователя.

Затем я таким же случайным образом расставлю пропуски в этом поле. На сформированном поле будут размещаться нули, а на экране пустые клетки. Для каждого уровня сложности предусмотрено, что количество пропусков в игровом поле будет определённым. Количество пропусков равно 9\*n, где n – номер уровня. Будет производиться случайная генерация индексов строки, столбца. Каждый элемент по этим индексам будет заполняться нулём. Причем на начальных уровнях для каждого квадрата 3х3 будет один пропуск, и в каждой строке и в каждом столбце повторений не будет. А при более высоких уровнях сложности (предположительно уровней сложности будет 7) количество пропусков будет добавляться и условия заполнения будут менее строгими. Я создам класс GameLevel, в котором и будет определяться уровень сложности.   
Для связи формирования поля и его выведения на экран будет создан класс клетка, каждая клетка будет содержать значение из соответствующей клетки игрового поля. А при выведении на экран, будет создано графическое игровое поле, которое будет строиться из 81 графической клетки с соответствующим значением класса Cell. Графическое игровое поле будет помещаться на экран, имеющий определенным фон.

**3)Графический интерфейс.**

Включает следующие абстрактные классы:  
a) Background (Игровой фон);

b) Graphic Field (Графическое поле);

c) Graphic Cell (Клетка);

d) Cursor (Курсор);

e) Game Menu (Игровое меню);

f) Keys (Игровая панель);

Графический интерфейс я буду создавать используя библиотеку Qt. Все, что будет видно на экране, будет вызываться с помощью классов механической составляющей игры.

При сформированном игровом поле, каждое значение будет помещаться в объект Cell (клетка). Для каждой такой клетки будет создаваться объект Graphic Cell (графическая клетка), всего будет 81 клетка, которые вместе будут выводиться как игровое поле Graphic Field, через соответствующий класс. Пользователь сможет проходить по полю и менять в нем значения при помощи класса Cursor (курсор). Предусмотрено построение и выведение игрового меню и игровой панели, которые также будут управляться с помощью курсора.