ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук

Департамент программной инженерии

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО |  | УТВЕРЖДАЮ |
| Доцент департамента программной инженерии,  к.т.н. |  | Академический руководитель образовательной программы «Программная инженерия» |
| \_\_\_\_\_\_\_Ахметсафина Р. З. |  | \_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов |
| “\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_2015 |  | “\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_2015 |

Инв. № подл.

Подпись и дата

Взам. инв. №

Инв. № дубл.

Подпись и дата

**Трехмерная игра "Судоку"**

**Пояснительная Записка**

**Лист утверждения**

**А.В.00001-01 ТЗ 01**

**RU.17701729.509000-01 81 01-1-ЛУ**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  | Исполнитель |
|  |  | Студент группы 203ПИ |
|  |  | \_\_\_\_\_\_\_\_\_\_\_Жадобин Ю.Д. |
|  |  | “\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_2015 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**УТВЕРЖДЕНО**

RU.17701729.509000-01 81 01-1-ЛУ

**Трехмерная игра "Судоку"**

**Пояснительная записка**

**RU.17701729.5090000-01 81 01-1**

**Листов 28**

**Инв. № подл.**

**Подпись и дата**

**Взам. инв. №**

**Инв. № дубл.**

**Подпись и дата**

**АННОТАЦИЯ**

В данном программном документе приведена пояснительная записка к программе Трехмерная игра "Судоку".

В данном программном документе, в разделе «Введение» указано наименование программы, краткое наименование программы и документы, на основании которых ведется разработка.

В разделе «Назначение и область применения» указано функциональное назначение программы и краткая характеристика области применения программы.

В данном программном документе, в разделе «Технические характеристики» содержатся следующие подразделы:

* постановка задачи на разработку программы;
* описание алгоритма и функционирования программы с обоснованием выбора схемы алгоритма решения задачи и возможные взаимодействия программы с другими программами;
* описание и обоснование выбора состава технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указана предполагаемая потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Настоящий документ разработан в соответствии с требованиями:

1. ГОСТ 19.101-77 Виды программ и программных документов [1];
2. ГОСТ 19.102-77 Стадии разработки [2];
3. ГОСТ 19.103-77 Обозначения программ и программных документов [3];
4. ГОСТ 19.104-78 Основные надписи [4];
5. ГОСТ 19.105-78 Общие требования к программным документам [5];
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];
7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [7].

Изменения к данному Техническому заданию оформляются согласно ГОСТ 19.603-78 [8], ГОСТ 19.604-78 [9].

# Содержание

[1. Введение 3](#_Toc419453887)

[1.1. Наименование программы 3](#_Toc419453888)

[1.2. Основание для проведения разработки 3](#_Toc419453889)

[2. Назначение разработки и область применения 4](#_Toc419453890)

[2.1. Назначение разработки 4](#_Toc419453891)

[2.2. Область применения 4](#_Toc419453892)

[3. Технические характеристики 5](#_Toc419453893)

[3.1. Постановка задачи на разработку программы 5](#_Toc419453894)

[3.1.1. Описание правил игры 5](#_Toc419453895)

[3.2. Описание алгоритма и функционирования программы 6](#_Toc419453896)

[3.2.1. Описание алгоритма генерации Судоку 6](#_Toc419453897)

[3.2.2. Описание алгоритма решения судоку 9](#_Toc419453899)

[3.2.2.1 Задача полного покрытия 9](#_Toc419453900)

[3.2.2.2 Описание алгоритма перехода от задачи Судоку к задаче полного покрытия 10](#_Toc419453901)

[3.2.2.3 Описание алгоритма нахождения решения 12](#_Toc419453903)

[3.3. Описание и обоснование выбора состава технических и программных средств 13](#_Toc419453904)

[3.4. Описание и обоснование выбора состава программных средств 13](#_Toc419453905)

[4. Ожидаемые технико-экономические показатели 14](#_Toc419453906)

[5. Источники, использованные пр разработке 15](#_Toc419453907)

[6. Приложения 16](#_Toc419453908)

[6.1. Описание всех классов 16](#_Toc419453909)

[6.1.3. Класс моделирующий Cудоку 16](#_Toc419453910)

[6.1.4. Классы интерфейса 16](#_Toc419453911)

[6.1.5. Класс для решения Судоку 16](#_Toc419453912)

[6.2. Описание всех методов, полей, свойств, событий 17](#_Toc419453913)

[6.2.6. Класс моделирующий Судоку 17](#_Toc419453914)

[6.2.7. Класс интерфейса 21](#_Toc419453915)

[6.3. UML-диаграмма 25](#_Toc419453916)

# Введение

## Наименование программы

Наименование - «Трехмерная игра "Судоку"».

## Основание для проведения разработки

Разработка ведётся на основании приказа Национального исследовательского университета "Высшая школа экономики" № 6.18.1-02/1912-11 от 19.12.2014.

Тема разработки: Трехмерная игра "Судоку".

Тема разработки на английском: The Three-Dimensional Game "Sudoku".

# Назначение разработки и область применения

## Назначение разработки

Программный продукт предназначен для удовлетворения потребностей человека в желании интеллектуально развиваться в легкой, игровой форме.

## Область применения

Область применения – развлекательно-обучающая.

# Технические характеристики

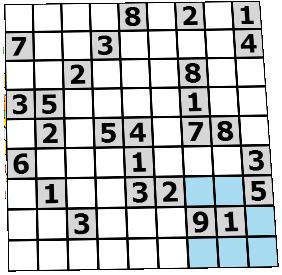
## Постановка задачи на разработку программы

Программа должна соответствовать требованиям, представленным в Техническом Задании.

### Описание правил игры

Судоку 3D состоит из трех граней куба, каждая из которых представляет собой классическую судоку с правилами:

Игровое поле представляет собой квадрат размером девять на девять, разделенный на меньшие квадраты размером три на три. В начале игры часть (количество определяется сложностью судоку) клеток открыта. Игроку предлагается заполнить свободные клетки цифрами от одного до девяти, так чтобы в каждой строке, в каждом столбце и в каждом малом квадрате три на три цифра встречалась всего один раз.



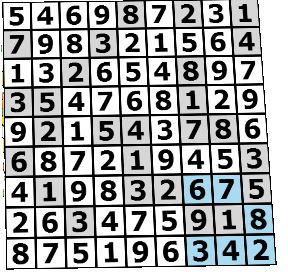


Рисунок 1 Пример решеного игрового поля

Рисунок 2 Пример начального игрового поля

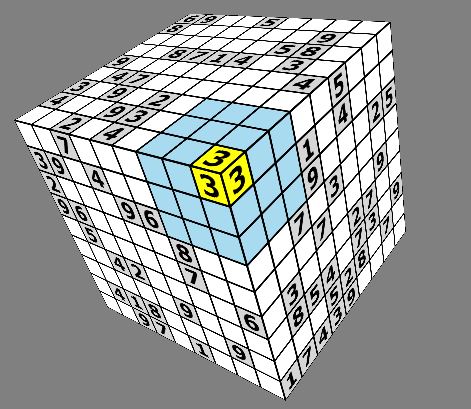
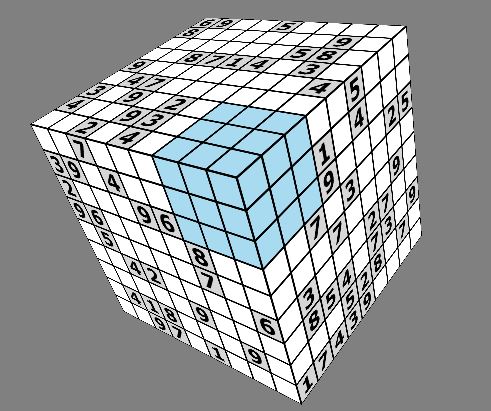
Так же в игре присутствуют специфические клетки (выделены синим), их особенность в том, что, когда пользователь заполняет клетку на одной из граней, автоматически заполняются этим клетки и на других гранях.

Рисунок 3 Пример заполнения

Рисунок 4 Пример начального игрового поля

Для победы игрок должен заполнить три грани куба, так что бы судоку оказались решены.

## Описание алгоритма и функционирования программы

### Описание алгоритма генерации Судоку

1. Генерация стартового поля Судоку.

Например: ;

1. Генерация новых полей судоку на основе комбинаций линейных преобразований стартового поля, не нарушающих правила игры:

* Транспонирование всей матрицы;
* Обмен двух строк в пределах одного квадрата;

Рисунок 5 Пример обмена двух строк

* Обмен двух колонок в пределах одного квадрата;

Рисунок 6 Пример обмена двух строк

* Обмен двух строк квадратов;

Рисунок 7 Пример обмена двух строк квадратов

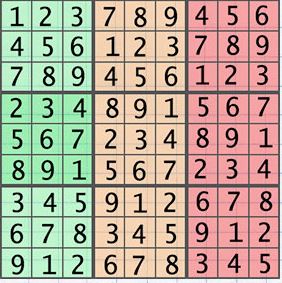
* Обмен двух колонок квадратов.

Рисунок 8 Пример обмена двух колонок квадратов

#### Скрываем цифры в клетках игрового поя. В эти клетки пользователь будет вносить значения. Количество скрываемых клеток зависит от сложности игры. Сложность игры – функция, зависящая от количества заполненных полей.

Данный алгоритм генерирует Судоку с как минимум одним решением.

### Описание алгоритма решения судоку

Программа реализует алгоритм Дональда Кнута Algorithm X [1] для решения задачи полного покрытия, к которой сводится задача Судоку.

#### **Задача полного покрытия**

Исходные данные-матрица, состоящая только из 0 и 1, требуются найти такое множество строк, что бы в каждой колонке была всего одна.

Например, для матрицы,

решением будет набор из строк 1,4,5: .

#### **Описание алгоритма перехода от задачи Судоку к задаче полного покрытия.**

В судоку есть 4 правила:

N – Размер судоку.

1. Каждая строка в таблице должна иметь все числа из отрезка [1; N];
2. Каждый столбец должен так же иметь все числа из отрезка [1; N];
3. Каждый квадрат должен иметь все числа из отрезка [1; N];
4. Каждая клетка должна содержать уникальное значение из отрезка

[1; N].

Таким образом у нас всего есть 4\*N\*N – требований и N\*N\*N - решений

## Создадим матрицу размерностью [N\*N\*N;4\*N\*N], где N- размер Судоку. Строки матрицы – это кандидаты на решение.

Столбцы матрицы – требования.

Дальше требуется ее заполнить, ниже представлен псевдокод.

// Итерирование по всем возможным цифрам

for(int d = 0; d < N; d++)

{

// Итерирование по всем возможным строкам

for(int r = 0; r < N; r++)

{

// Итерирование по всем возможным колонкам

for(int c = 0; c < N; c++)

{

Если клетка судоку открыта то

{

int rowIndex = c + (N \* r) + (N \* N \* d);

int blockIndex = ((c / SIZE) + ((r / SIZE) \* SIZE));

int colIndexRow = 3\*N\*d+r;

int colIndexCol = 3\*N\*d+N+c;

int colIndexBlock = 3\*N\*d+2\*N+blockIndex;

int colIndexSimple = 3\*N\*N+(c+N\*r);

matrix[rowIndex][colIndexRow] = 1;

matrix[rowIndex][colIndexCol] = 1;

matrix[rowIndex][colIndexBlock] = 1;

matrix[rowIndex][colIndexSimple] = 1;

}

}

}

}

1. Получается матица, в которой количество единиц мало. Для быстрого выполнения алгоритма требуется создать четырехсвязный список по единицам.

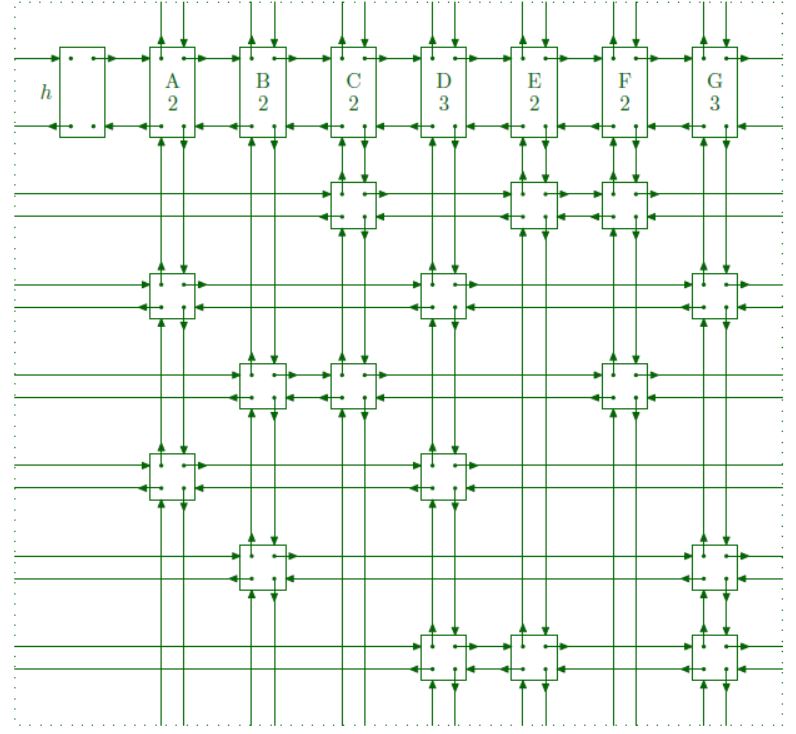


Рисунок 9 Четырехсвязный список

#### **Описание алгоритма нахождения решения**

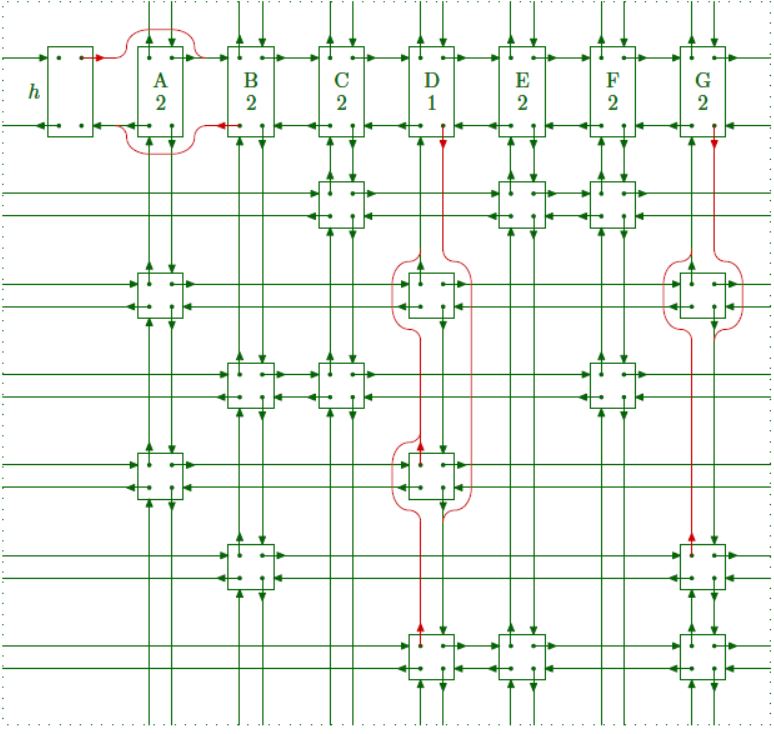
1. Введем операцию Cover в которой необходимо удалить строку и все столбцы, которые пересекает данная строка, а также все строки, пересекаемые столбцами.

Рисунок 10 Применение операции Cover для узла А

1. Последовательно перебираем требования. Если на текущем шаге не осталось требований, значит нужно проверить получившееся решение на корректность.
2. Для каждого требования пытаемся использовать строку-решение, с которой пересекается данный столбец. Для таких строк выполняем операцию Cover, исключая их из рассмотрения.
3. Переходим в новое состояние, к следующему требованию.
4. Когда выходим из рекурсии восстанавливаем состояние до вызова функции Cover и находим следующее решение.

## Описание и обоснование выбора состава технических и программных средств

«Трехмерная игра судоку» обладает средними по современным меркам требованиями к аппаратной части ПК: рекомендуется компьютер, оснащенный процессором с частотой 1 ГГц или более (одно- или двухпроцессорная система), имеющий 1024 МБ или более оперативной памяти. Исходя из возможностей программы, требуется монитор и видеоадаптер с минимальным разрешением 800 x 600. Для взаимодействия с программой необходимы: компьютерная мышь и клавиатура.

## Описание и обоснование выбора состава программных средств

«Трехмерная игра судоку» предназначена для работы под

управлением любой операционной системы, для которой реализована виртуальная java-машина не ниже версии 8. В качестве средств разработки была выбраны среды IntelliJ IDEA Studio и jMonkey Studio. Разработка программы велась на языке Java версии 1.8.

# Ожидаемые технико-экономические показатели

Данное приложение уникально на рынке в данный момент. Поэтому она является наилучшим решением для реализации поставленной задачи.

# Источники, использованные при разработке

Список литературы, использованной в разработке, содержится в Приложении документа «Техническое задание».

# Приложения

## Описание всех классов

### Класс моделирующий Cудоку

|  |  |  |
| --- | --- | --- |
| class | Cell | Моделирует одну ячейку |
| class | Matrix | Моделирует матрицу |
| class | Sudoku | Моделирует классическую судоку |
| class | UpperSudoku:Sudoku | Моделирует верхнюю плоскость судоку |
| class | FrontSudoku:Sudoku | Моделирует переднюю плоскость судоку |
| class | SideSudoku:Sudoku | Моделирует боковую плоскость судоку |
| class | Sudoku3D | Моделирует судоку 3D |
| class | Generator | Класс для генерации новых судоку |

### Классы интерфейса

|  |  |  |
| --- | --- | --- |
| class | MainFrame | Основное окно |
| class | SudokuShowerTable | Таблица для взаимодействия с программой |
| class | BoxViewer | Создает 3D модель |
| class | TablePanel | Инкапсулирует таблицы |
| class | ViewPanel | Инкапсулирует модель |

### Класс для решения Судоку

|  |  |  |
| --- | --- | --- |
| class | AlgorithmXSolver | Класс для решения Судоку |

## Описание всех методов, полей, свойств, событий

### Класс моделирующий Судоку

class Cell

Поля

|  |  |  |  |
| --- | --- | --- | --- |
| Мод. доступа | Тип | Имя | Назначение |
| private | Byte | value | Значение в клетке |
| public | Boolean | isChangable | Флаг, показывающий можно ли менять значение |
| public | Boolean | isSpecial | Флаг, показывает особая ли клетка |
| public | boolean | isSelected | Флаг, выбрана ли клетка |

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Мод. доступа | Тип | Имя | Назначение |
| public | byte | getValue | Возвращает значение |
| Public | void | setValue | Устанавливает значение в клетке |

Методы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Мод. доступа | Тип возвр. знач. | Имя | Параметры | Назначение |
| public | Cell | copy |  | Копирует данный объект |
| public |  | Cell | Byte | Создает клетку с заданным значением |

class Matrix

Поля

|  |  |  |  |
| --- | --- | --- | --- |
| Мод. доступа | Тип | Имя | Назначение |
| private | Cell[][] | matrix | Двумерный массив клето |

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Мод. доступа | Тип | Имя | Назначение |
| public | Cell[][] | getCells | Возвращает массив клеток |

Методы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Мод. доступа | Тип возвр. знач. | Имя | Параметры | Назначение |
| public | string | toString |  | Возвращает строковое представление объекта |
| public |  | Cell | Byte | Создает клетку с заданным значением |
| public |  | setValue | Int, int, byte | Устанавливает значение в клетку |
| public |  | transporation | Matrix | Транспонирует матрицу |
| Public | Matrix | generateBaseMatrix |  | Создает базовую матрицу |
| Public | Matrix | Copy |  | Возвращает копию объекта |

class Sudoku

Поля

|  |  |  |  |
| --- | --- | --- | --- |
| Мод. доступа | Тип | Имя | Назначение |
| private | int | size | Размер судоку |
| Public | List<Images> | Images | Лист для хранения картинок |
| Public | Boolean | isChanged | Флаг, показывающий изменилось ли состояние объекта |
| public | Boolean | isUpdating | Флаг, показывающий меняется ли в данный момент состояние объекта |

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Мод. доступа | Тип | Имя | Назначение |
| public | Matrix | getMatrix | Возвращает поле для игры |

Методы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Мод. доступа | Тип возвр. знач. | Имя | Параметры | Назначение |
| public | string | toString |  | Возвращает строковое представление объекта |
| public |  | Cell | Byte | Создает клетку с заданным значением |
| public |  | setValue | Int, int, byte | Устанавливает значение в клетку |
| public |  | transporation | Matrix | Транспонирует матрицу |
| Public | Matrix | generateBaseMatrix |  | Создает базовую матрицу |
| Public | Matrix | Copy |  | Возвращает копию объекта |

class FrontSudoku

Методы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Мод. доступа | Тип возвр. знач. | Имя | Параметры | Назначение |
| public |  | FrontSudoku | Sudoku | Конструктор |
| public |  | generate |  | Генерирует Судоку |
| public |  | setValue | Int, int, byte | Устанавливает значение в клетку |
| public |  | getValue | Int, int | Возвращает значение из клетки |
| Public | FrontSudoku | Copy |  | Возвращает копию объекта |

class SideSudoku

Методы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Мод. доступа | Тип возвр. знач. | Имя | Параметры | Назначение |
| public |  | SideSudoku | Sudoku | Конструктор |
| public |  | generate |  | Генерирует Судоку |
| public |  | setValue | Int, int, byte | Устанавливает значение в клетку |
| public |  | getValue | Int, int | Возвращает значение из клетки |
| Public | SideSudoku | Copy |  | Возвращает копию объекта |

class UpperSudoku

Методы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Мод. доступа | Тип возвр. знач. | Имя | Параметры | Назначение |
| public |  | UpperSudoku | Sudoku | Конструктор |
| public |  | generate |  | Генерирует Судоку |
| public |  | setValue | Int, int, byte | Устанавливает значение в клетку |
| public |  | getValue | Int, int | Возвращает значение из клетки |
| Public | UpperSudoku | Copy |  | Возвращает копию объекта |

class Sudoku3D

Поля

|  |  |  |  |
| --- | --- | --- | --- |
| Мод. доступа | Тип | Имя | Назначение |
| private | FrontSudoku | frontSudoku | Часть судоку |
| Public | UpperSudoku | upperSudoku | Часть судоку |
| Public | SideSudoku | sideSudoku | Часть судоку |

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Мод. доступа | Тип | Имя | Назначение |
| public | FrontSudoku | getFrontSudoku | Возвращает переднюю часть судоку |
| public | UpperSudoku | getupperSudoku | Возвращает верхнюю часть судоку |
| public | SideSudoku | getsideSudoku | Возвращает боковую часть судоку |

Методы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Мод. доступа | Тип возвр. знач. | Имя | Параметры | Назначение |
| public | string | toString |  | Возвращает строковое представление объекта |
| public | boolean | isSolved |  | Проверяет решена ли Судоку |
| public |  | reGenerate |  | Перегенерация поля, без создания нового объекта |
| public |  | solve |  | Решает данную задачу |
| Public | Sudoku3D | generate |  | Генерирует судоку |
| Public | Sudoku3D | Copy |  | Возвращает копию объекта |

class Generator

Поля

|  |  |  |  |
| --- | --- | --- | --- |
| Мод. доступа | Тип | Имя | Назначение |
| private | int | NUMBER\_OF\_SWAPS | Количество перестановок |
| private | int | NUMBER\_OF\_DELETED\_SELLS | Количество «открытых» клеток в судоку |

Методы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Мод. доступа | Тип возвр. знач. | Имя | Параметры | Назначение |
| public |  | deleteCells |  | “Открывает” ячейки в судоку |
| public |  | swapTwoVerticalRows | Int,int,Matrix | Меняет местами 2 колонки в судоку |
| public |  | swapTwoHorizontalRows | Int,int,Matrix | Перегенерация поля, без создания нового объекта |
| public |  | swapTwoHorizontalRowsOfCubes | Int,int,Matrix | Меняет местами 2 ряда кубиков(область 3\*3) |
| Public |  | swapTwoVerticalRowsOfCubes | Int,int,Matrix | Меняет местами 2 колонки кубиков |

### Класс интерфейса

class MainFrame

Поля

|  |  |  |  |
| --- | --- | --- | --- |
| Мод. доступа | Тип | Имя | Назначение |
| private | Sudoku3D | sudoku3D | Текущее состояние игры |
| private | Sudoku3D | startSudoku | Начальное состояние игры |
| private | TablePanel | tablePanel | Ссылка на панель таблиц |
| private | ViewPanel | viewPanel | Ссылка на панель, на которой находится модель |

Методы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Мод. доступа | Тип возвр. знач. | Имя | Параметры | Назначение |
| public |  | MainFrame | Sudoku3D | Конструктор |
| public |  | setLayout |  | Устанавливает разметку |
| public |  | createMenu |  | Создает меню формы |
| public |  | initComponents |  | Инициализация компонентов формы |

class ViewPanel

Поля

|  |  |  |  |
| --- | --- | --- | --- |
| Мод. доступа | Тип | Имя | Назначение |
| private | Sudoku3D | sudoku | Текущее состояние игры |
| private | SimpleApplication | canvasApplication | 3D Модель |

Методы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Мод. доступа | Тип возвр. знач. | Имя | Параметры | Назначение |
| public |  | ViewPanel | Sudoku3D | Конструктор |
| public |  | getCanvas |  | Создает модель |
| public |  | update |  | Перерисовывает модель |

class BoxViewer

Поля

|  |  |  |  |
| --- | --- | --- | --- |
| Мод. доступа | Тип | Имя | Назначение |
| private | Sudoku3D | sudoku | Текущее состояние игры |
| private | Geometry | upperMesh; |  |
| private | Geometry | sideMesh; |  |
| private | Geometry | frontMesh; |  |

Методы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Мод. доступа | Тип возвр. знач. | Имя | Параметры | Назначение |
| public |  | BoxViewer | Sudoku3D | Конструктор |
| public |  | simpleInitApp |  | Инициализация модели |
| private |  | initMeshes |  | Инициализация сторон |
| public |  | simpleUpdate |  | Обновление сцены |
| private | BufferedImage | convert | Image | Преобразовывает формат изображения |

class SudokuShowerTable

Поля

|  |  |  |  |
| --- | --- | --- | --- |
| Мод. доступа | Тип | Имя | Назначение |
| private | Sudoku3D | sud | Текущее состояние игры |
| private | Sudoku | sudoku | Судоку, которое отображает данная таблица |
| private | TablePanel | tablePanel | Ссылка, на обертывающую панель |

Методы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Мод. доступа | Тип возвр. знач. | Имя | Параметры | Назначение |
| public |  | SudokuShowerTable | Sudoku,int, TablePanel, Sudoku3D | Конструктор |
| public |  | update |  | Обновление состояния |
| private | boolean | isCorrect | Int, int, int | Проверка корректное ли данные введены |
| public |  | simpleUpdate |  | Обновление сцены |
| private | BufferedImage | convert | Image | Преобразовывает формат изображения |

Class TablePanel

Поля

|  |  |  |  |
| --- | --- | --- | --- |
| Мод. доступа | Тип | Имя | Назначение |
| private | Sudoku3D | Sudoku3D | Текущее состояние игры |
| private | SudokuShowerTable | frontSudokuShowerTable | Ссылка на таблицу |
| private | SudokuShowerTable | sideSudokuShowerTable | Ссылка на таблицу |
| private | SudokuShowerTable | upperSudokuShowerTable | Ссылка на таблицу |

Методы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Мод. доступа | Тип возвр. знач. | Имя | Параметры | Назначение |
| public |  | updateGrids |  | Обновление таблиц |
| public |  | update |  | Обновление состояния |
| private |  | initComponents |  | Инициализация компонентов |
| public |  | simpleUpdate |  | Обновление сцены |
| private |  | TablePanel | Sudoku3D | Конструктор |

Class AlgorithmXSolver

Поля

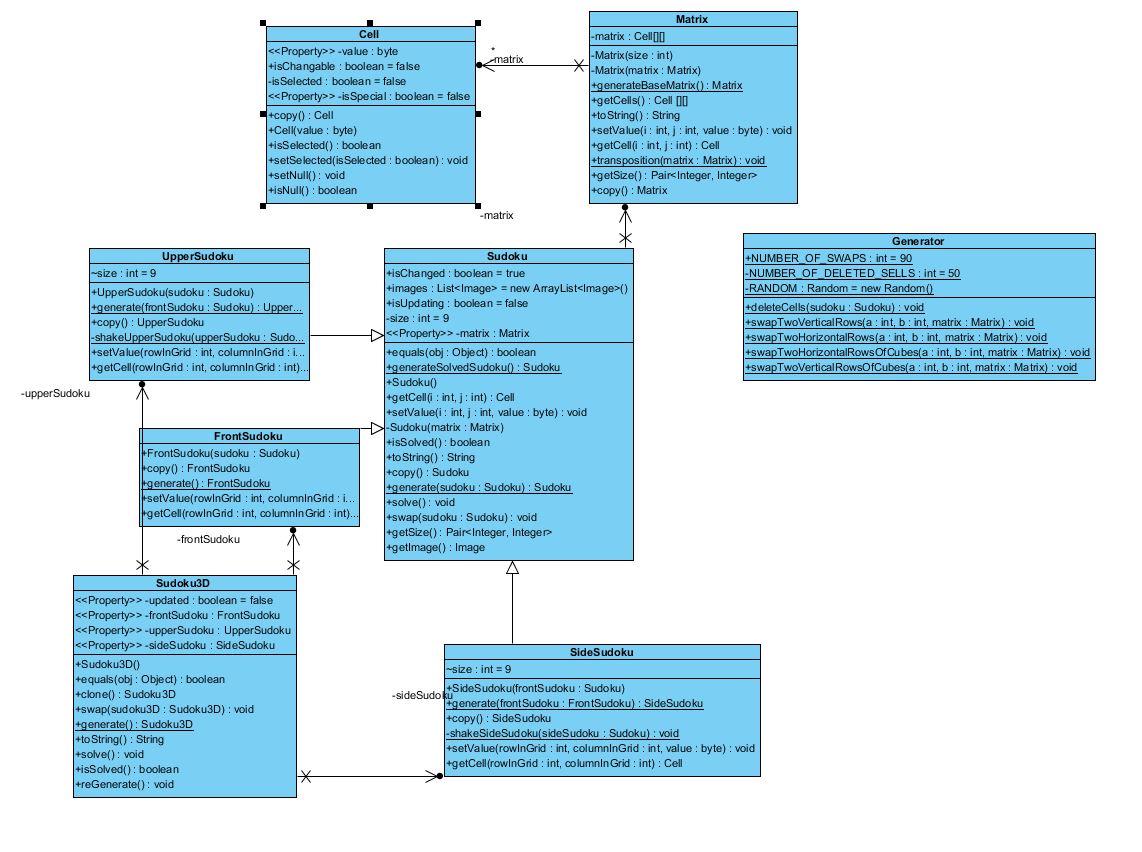
|  |  |  |  |
| --- | --- | --- | --- |
| Мод. доступа | Тип | Имя | Назначение |
| private | int | N | Размер |
| Private | int | SIZE | Размер одного блока |
| private | Matrix | Grid | поле |
| Private | Node | root | Корень связанного листа |

Методы

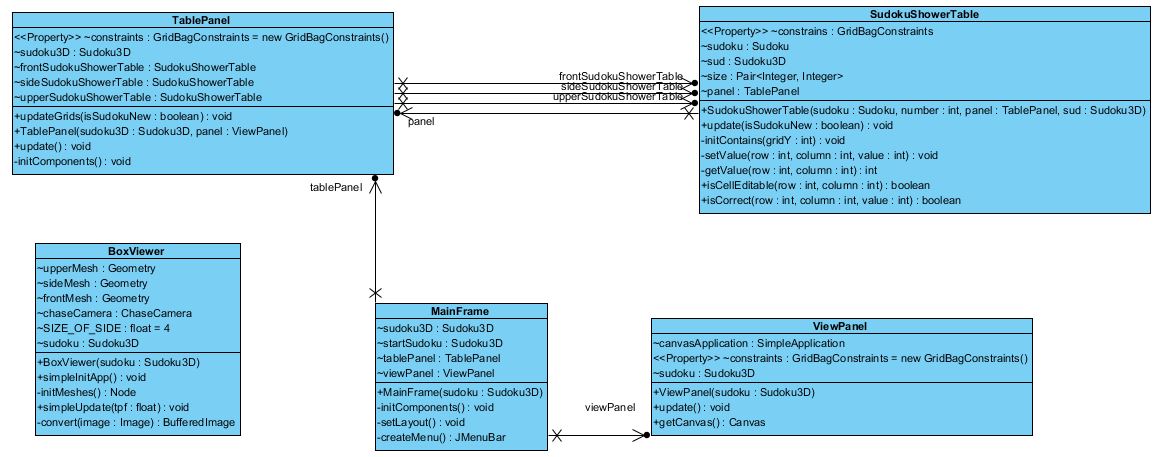
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Мод. доступа | Тип возвр. знач. | Имя | Параметры | Назначение |
| public |  | AlgorithmXSolver |  | Конструктор |
| public |  | solve |  | Запуск алгоритма |
| private |  | run | Matrix | Запуск алгоритма с параметром |
| private | Byte[][] | createMatrix | Matrix | Генерация разряженной матрицы |
| private | Root | createDoubleLinkedList | Byte[][] | Cоздает двухсвязанный список |
| private |  | Search |  | Поиск следующего решения |
| private | Boolean | filled | Int,int,int,int[][] | Проверка не внесена еще ли в таблицу данное значение |
| private |  | cover | Node | Покрытие данного узла |
| private |  | uncover | Node | Обратная операция к cover |
| private |  | mapSolvedToGrid |  | Преобразует разряженный список к решению |

## UML-диаграмма

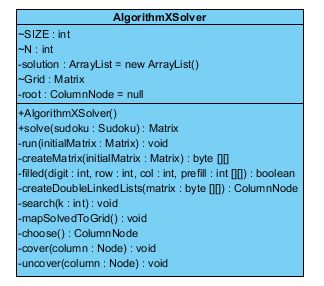
Классы представляющий модель



Классы интерфейса



Класс «солвера»

****

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Изм. | Номера листов | | | | Всего листов (страниц) в докум. | № докум. | Входящий №  сопроводительного докум. и дата | Подп. | Дата |
| изме-ненных | заме-ненных | новых | анну-лиро-ванных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |