**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Профессор  факультета компьютерных наук  доктор физико-математических наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.В. Аржанцев  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** | **RU.17701729.507140-01 81 01-1** | | **ПРОГРАММА КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ АЛГЕБРОГЕОМЕТРИЧЕСКИХ КОДОВ**  **Пояснительная записка**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.507140-01 81 01-1-ЛУ** | | |
|  |  | |
| Исполнитель:  студент группы БПИ162  \_\_\_\_\_\_\_\_\_\_\_\_ /Казанцева Анастасия Романовна /  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г. | |
|  | | |
|  | |  |

**2017**

|  |  |  |  |
| --- | --- | --- | --- |
| **УТВЕРЖДЕНО**  **RU.17701729.507140-01 81 01-1-ЛУ** | |  |  |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** | **RU.17701729.507140-01 12 01-1** | | **ПРОГРАММА КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ АЛГЕБРОГЕОМЕТРИЧЕСКИХ КОДОВ**  **Пояснительная записка**  **RU.17701729.507140-01 81 01-1**  **Листов 30** | | | | |
|  |  | | | |
|  | | | |
| **2017** | | | | |

|  |  |
| --- | --- |
|  |  |
|  |  |

СОДЕРЖАНИЕ

[1. ВВЕДЕНИЕ 5](#_Toc482710269)

[1.1. Наименование программы 5](#_Toc482710270)

[1.2. Документы, на основании которых ведется разработка 5](#_Toc482710271)

[2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 6](#_Toc482710272)

[2.1. Назначение программы 6](#_Toc482710273)

[2.1.1. Функциональное назначение 6](#_Toc482710274)

[2.1.2. Эксплуатационное назначение 6](#_Toc482710275)

[2.2. Краткая характеристика области применения 6](#_Toc482710276)

[3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 7](#_Toc482710277)

[3.1. Постановка задачи на разработку программы 7](#_Toc482710278)

[3.2. Описание алгоритма и функционирования программы 7](#_Toc482710279)

[3.3. Обоснование выбора алгоритма решения задачи 10](#_Toc482710280)

[3.4. Описание и обоснование выбора метода организации входных и выходных данных 11](#_Toc482710281)

[3.5. Описание и обоснование выбора состава технических и программных средств 11](#_Toc482710282)

[3.5.1. Состав технических и программных средств 11](#_Toc482710283)

[3.5.2. Обоснование выбора технических и программных средств 11](#_Toc482710284)

[4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 13](#_Toc482710285)

[4.1. Предполагаемая потребность 13](#_Toc482710286)

[4.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами 13](#_Toc482710287)

[5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ 14](#_Toc482710288)

[ПРИЛОЖЕНИЕ 1.ТЕРМИНОЛОГИЯ 15](#_Toc482710289)

[ПРИЛОЖЕНИЕ 2.ОПИСАНИЕ ФОРМАТА .nk 16](#_Toc482710291)

[ПРИЛОЖЕНИЕ 3. ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ 17](#_Toc482710293)

[ПРИЛОЖЕНИЕ 4. ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ 20](#_Toc482710295)

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ………………………………………………..31

**АННОТАЦИЯ**

В данном программном документе приведена пояснительная записка к программе «Генератор алгеброгеометрических кодов» («Программа кодирования и декодирования алгеброгеометрических кодов»), предназначенной для создания алгеброгеометрических кодов и демонстрации их действия.

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, эксплуатационное назначение программы и краткая характеристика области применения программы.

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 метода организации входных и выходных данных; − 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 Пояснительной записке оформляются согласно ГОСТ 19.603-78 [8], ГОСТ 19.604-78 [9].

Перед прочтением данного документа рекомендуется ознакомиться с терминологией, приведенной в Приложении 1 настоящей пояснительной записки.

1. **ВВЕДЕНИЕ**
   1. **Наименование программы**

**Наименование программы:** «Программа для кодирования и декодирования алгеброгеометрических кодов».

**Краткое наименование программы:** «Генератор алгеброгеометрических кодов».

**Условное обозначение темы разработки:** «A Program for Coding and Decoding of Algebraic Geometry Codes».

* 1. **Документы, на основании которых ведется разработка**

Программа выполнена в рамках темы курсовой работы «Программа кодирования и декодирования алгеброгеометрических кодов» в соответствии с учебным планом подготовки бакалавров (НИУ ВШЭ, факультет компьютерных наук) по направлению «Программная инженерия».

Разработка ведется на основании приказа Национального исследовательского университета "Высшая школа экономики" № 2.3-02/0812-01 от 08.12.2016

1. **НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ**
   1. **Назначение программы**
      1. **Функциональное назначение**

Программа на основе входных данных составляет алгеброгеометрический код и демонстрирует пользователю основные характеристики этого кода. После получения характеристик пользователь может вводить соответствующие, корректные данные, которые он хочет закодировать или декодировать. Программа может кодировать, декодировать полученные строки и, опираясь на индивидуальные возможности каждого кода, исправлять ошибки, допущенные в декодируемых строках.

* + 1. **Эксплуатационное назначение**

Программа демонстрирует процесс и результат кодирования и декодирования алгеброгеометрических кодов, а также дает возможность пользователю исследовать некоторые их особенности.

Конечными пользователями программы могут быть научные деятели, преподаватели, а также студенты, интересующиеся кодированием, в частности алгеброгеометрическими кодами.

* 1. **Краткая характеристика области применения**

«Программа кодирования и декодирования алгеброгеометрических кодов» – прикладная программа учебного назначения, позволяющая демонстрировать процесс и результат работы алгеброгеометрических кодирующих систем.

1. **ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ**
   1. **Постановка задачи на разработку программы**

Разрабатываемая программа должна:

1. Создавать алгеброгеометрические коды;
2. Выполнять кодирование с помощью созданных алгеброгеометрических кодов;
3. Выполнять декодирование с помощью созданных алгеброгеометрических кодов;
   1. **Описание алгоритма и функционирования программы**
      1. **Алгоритм решения системы нелинейных уравнений над полем Z2**

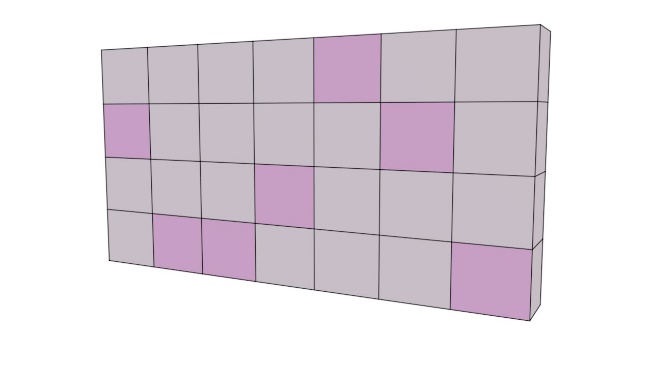
Алгоритм решения системы нелинейных уравнений над полем Z2  реализуется следующим образом: пользователь набирает в специально отведенном поле систему уравнений, соответствующую по формату примеру, приведенному в справке, указывает количество переменных, относительно которых данная система уравнений будет решаться. Эти входные данные поступают в программу, где система уравнений представляется в удобном для обработки виде List<Equation>, где Equation – это уравнение, представленное в виде Tuple<List<bool[]>, bool>, где List<bool[]> – это часть уравнения с переменными, а bool – это итоговый результат суммы всех свободных членов. Наглядное представление системы изображено на рис. 1, где блоками одного цвета указаны элементы списка List. На рис. 2 показан общий вид уравнения (той части где содержатся переменные). Каждый столбец – это слагаемое в уравнении, состоящее из перемноженных переменных, каждая из которых отображается кубом. Высота столбца – это общее количество переменных в данной системе уравнений, цвет куба – отражает принадлежность или отсутствие каждой переменной в слагаемом.

Рисунок 2. Уравнение

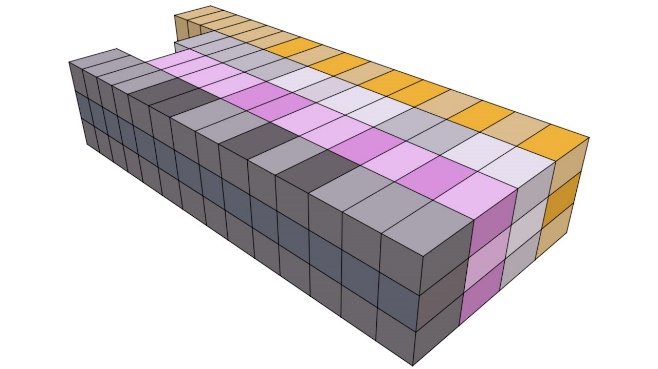


Рисунок 1. Система уравнений

Решение уравнения производит метод Solve класса Solver.

/// <summary>

/// Решает систему уравнений.

/// </summary>

/// <param name="equationsSystem">Система уравнений</param>

/// <returns>Решение системы уравнений</returns>

public SolutionsMatrix Solve(List<Equation> equationsSystem)

{

if (equationsSystem == null)

throw new NullReferenceException("Система уравнений пуста!");

if (equationsSystem.Count==0)

throw new NullReferenceException("Введите систему уравнений!");

matrix = new SolutionsMatrix();

// Перебор всех возможных решений.

for (int possibleSolution=0; possibleSolution<Math.Pow(2, equationsSystem[0].GroupSize); possibleSolution++)

{

// Представление номера возможного решения в двоичном виде.

int[] intArrayPossibleSolution = MyStatics.ToBinaryIntArray(possibleSolution, equationsSystem[0].GroupSize);

// Создание проверочного листа - элемента, помогающего определить принадлежит ли решение множеству решений данной системы.

checkingList = new СheckList(equationsSystem);

// Заполнение проверочного листа возможным решением.

checkingList.FillingTheCheckList(intArrayPossibleSolution);

/// Сверка значений полученного в результате подстановки решения в проверочный лист и значения,

/// которое должно быть по условию.

/// В случае полного совпадения решение, которое подставлялось в проверочный лист

/// добавляется в матрицу решений системы уравнений.

if (checkingList.IsItRightSolution()) matrix.Matrix.Add(intArrayPossibleSolution);

}

return matrix;

}

Описание использованных в алгоритме методов доступно в Приложении 4.

Результатом применения алгоритма решения системы нелинейных уравнений над полем Z2 является объект класса SolutionsMatrix. Матрица решений в этом классе представлена полем List<int[]>, где int[] – это каждое решение системы.

* + 1. **Алгоритм поиска минимального расстояния кода**

Алгоритм поиска минимального расстояния кода реализуется следующим образом: матрица решений системы уравнений используется в качестве порождающей матрицы для создаваемого программой алгеброгеометрического кода типа Code. В связи с тем, что вся работа программы происходит над конечным полем Z2, количество слов, которые будет возможно закодировать алгеброгеометрическим кодом ограничено и равно 2k, где k – длина кодируемого слова. Умножением всех вариантов слов на порождающую матрицу кода создается список всех кодовых слов.

allCodeWords = new BaseMatrix();

for (int possibleCodeNumber = 0; possibleCodeNumber < Math.Pow(2, K); possibleCodeNumber++)

{

int[] intArrayPossibleCodeNumber = MyStatics.ToBinaryIntArray(possibleCodeNumber, K);

BaseMatrix vectorPossibleCodeNumber = new BaseMatrix();

for (int i = 0; i < K; i++)

{

int[] nextLine = { intArrayPossibleCodeNumber[i] };

vectorPossibleCodeNumber.Matrix.Add(nextLine);

}

BaseMatrix vectorPossibleCode = MyStatics.Multiplication(vectorPossibleCodeNumber, generatingMatrix);

allCodeWords.Matrix.Add(vectorPossibleCode.Matrix[0]);

}

Перебором в статическом методе производится поиск наименьшего расстояния между кодовыми словами.

/// <summary>

/// Ищет минимальное расстояние в коде.

/// </summary>

/// <param name="allCodeWords">Множество всех кодовых слов</param>

/// <returns>Минимальное расстояние</returns>

public static int FindMinDistance(BaseMatrix allCodeWords)

{

int minDistance = Int32.MaxValue, currentDistance = 0;

for (int i = 0; i < allCodeWords.Matrix.Count; i++)

{

for (int j = i + 1; j < allCodeWords.Matrix.Count; j++)

{

for (int k = 0; k < allCodeWords.Matrix[0].Length; k++)

currentDistance += (allCodeWords.Matrix[i][k] + allCodeWords.Matrix[j][k]) % 2;

if (currentDistance < minDistance) minDistance = currentDistance;

currentDistance = 0;

}

}

return minDistance;

}

* + 1. **Алгоритм кодирования сообщения**

Алгоритм кодирования сообщения реализуется следующим образом: пользователь в специальное поле вводит строку определенной длины из нулей и единиц. Строка поступает в метод Encode класса Code, где производится умножение ее на порождающую матрицу выбранного кода.

/// <summary>

/// Кодирует полученное сообщение.

/// </summary>

/// <param name="ourMessage">Кодируемое слово</param>

/// <returns>Код</returns>

public int[] Encode(int[] ourMessage)

{

BaseMatrix A = new BaseMatrix(ourMessage);

BaseMatrix C = MyStatics.Multiplication(A, generatingMatrix); ;

return C.Matrix[0];

}

* + 1. **Алгоритм декодирования сообщения с исправлением ошибок**

Алгоритм декодирования сообщения с исправлением ошибок реализуется следующим образом: пользователь в специальное поле вводит строку определенной длины из нулей и единиц. Строка поступает в метод Decode класса Code, где производится сравнение поступившего сообщения со всеми кодовыми словами. То кодовое слово, которое находится на наименьшем расстоянии от поступившего, и будет исправленным кодовым словом. Декодирование производится исходя из алгоритма построения базы кодовых слов: номер кодового слова соответствует его декодируемой версии.

/// <summary>

/// Декодирует полученное сообщение, исправляя ошибки.

/// </summary>

/// <param name="ourMessage">Декодируемое слово</param>

/// <returns>Исходное сообщение</returns>

public int[] Decode(int[] ourMessage)

{

int currentDistance = 0, minDistance = N + 1, number = 0;

// Совершаем проход по всем возможным кодовым словам.

for (int i = 0; i < allCodeWords.Matrix.Count; i++)

{

for (int j = 0; j < allCodeWords.Matrix[i].Length; j++)

if (ourMessage[j] != allCodeWords.Matrix[i][j]) currentDistance++;

if (currentDistance < minDistance)

{

minDistance = currentDistance;

number = i;

}

currentDistance = 0;

if (minDistance == 0) return MyStatics.ToBinaryIntArray(number, K);

}

if (minDistance > T)

throw new MistakesNumberException("Количество ошибок, сделанных в кодовом слове, превышено!");

return MyStatics.ToBinaryIntArray(number, K);

}

Описание использованных в алгоритме методов доступно в Приложении 4.

* 1. **Обоснование выбора алгоритма решения задачи**

Данный алгоритм решения был выбран мною, так как я считаю его наиболее верным и удобным с точки зрения реализации в рамках условий поставленной задачи.

* 1. **Описание и обоснование выбора метода организации входных и выходных данных**

Входные данные могут быть внесены пользователем вручную и быть получены из файла формата .nk [см. Приложение 2].

Вручную пользователь может ввести систему уравнений, задать количество переменных, относительно которых данное уравнение должно решаться, и, нажав кнопку «Вперед» получить алгеброгеометрический код, либо десериализовать код из файла с разрешением .nk[см. Приложение 2]. Помимо этого пользователь может ввести сообщения для кодирования или декодирования в специально отведенные поля в программе.

Выходные данные представляются в виде алгеброгеометрического кода и сохраняются в файл формата .nk [см. Приложение 2], либо в виде сообщений, состоящих из символов 0 и 1,полученных в результате кодирования или декодирования.

При работе с файлами формата .nk [см. Приложение 2] используется встроенный механизм сериализации в двоичный формат и десериализации из двоичного формата.

Выбор входных и входных данных обусловлен установленным функционалом программы.

* 1. **Описание и обоснование выбора состава технических и программных средств**
     1. **Состав технических и программных средств**

Для работы программы необходим следующий состав программных средств:

1) операционная система Microsoft Windows XP SP3 или более поздняя версия;

2) установленный Microsoft .NET Framework 2.0, требующий Windows Installer 3.1 или более поздняя версия;

Для работы программы необходим следующий состав технических средств:

1) персональный компьютер, оснащенный 32-разрядным (x86) или 64-разрядным (x64) процессором Pentium с тактовой частотой 400 MГц и выше или аналогичный процессор (рекомендуется Pentium с тактовой частотой 1 ГГц и выше или аналогичный процессор);

2) 96 МБ оперативной памяти или больше (рекомендуется 256 МБ оперативной памяти или больше);

3) не менее 1,5 ГБ свободного места на жестком диске;

4) видеокарта и монитор, поддерживающие режим Super VGA с разрешением не менее чем 850x600 точек;

5) мышь Microsoft Mouse или совместимое указывающее устройство;

6) клавиатура;

7) CD-ROM привод.

* + 1. **Обоснование выбора технических и программных средств**

При реализации программы использован тип List<T>, где T – тип элементов в списке, из пространства имен System.Collections.Generic, который впервые был представлен в платформе Microsoft .NET Framework 2.0. Остальные используемые в программе пространства имен были представлены и в более ранних версиях Microsoft .NET Framework.

Microsoft .NET Framework 2.0 в свою очередь требует [21,22]:

1) операционную систему Windows XP SP3 и выше;

2) процессор Pentium с тактовой частотой 400 MГц и выше или аналогичный процессор (рекомендуется Pentium с тактовой частотой 1 ГГц и выше или аналогичный процессор);

3) 96 МБ оперативной памяти или больше (рекомендуется 256 МБ оперативной памяти или больше);

4) установщик Windows 3.1 или более поздней версии;

5) Internet Explorer 6.0 или более поздней версии;

6) 0,28 гигабайт (ГБ) (для 32-разрядной системы) или 0,61 ГБ (для 64-разрядной системы) пространства на жестком диске и выше.

Для установки Windows XP необходимы [16]:

1) процессор Pentium с частотой 233 МГц или более быстрый (рекомендуется не менее 300 МГц);

2) не менее 64 МБ оперативной памяти (рекомендуется не менее 128 МБ);

3) не менее 1,5 ГБ свободного места на жестком диске;

4) дисковод для компакт- или DVD-дисков;

5) клавиатура, мышь Microsoft Mouse или совместимое указывающее устройство;

6) видеокарта и монитор, поддерживающие режим Super VGA с разрешением не менее чем 800x600 точек;

Главное окно программы имеет разрешение 625x450 точек, поэтому разрешение монитора должно быть не менее чем 625x450 точек.

1. **ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ**

В рамках данной работы расчет экономической эффективности не предусмотрен.

* 1. **Предполагаемая потребность**

Данное приложение будет интересно преподавателям, а также студентам, изучающим теорию кодирования и желающим проводить исследования в области алгеброгеометрических кодов.

* 1. **Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами**

Данное приложение:

1. распространяется бесплатно;
2. не требует вложения денежных средств во время использования;
3. имеет неограниченный срок службы;
4. интуитивно просто в использовании;
5. **ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ**
6. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
11. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
12. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
13. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
14. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
15. ГОСТ 15150-69 Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды. – М.: Изд-во стандартов, 1997.
16. ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
17. Влэдуц С.Г., Ногин Д.Ю., Цфасман М.А. // Алгеброгеометрические коды. Основные понятия. - М.: МЦНМО, 2003. – 503 с.

# ПРИЛОЖЕНИЕ 1

**ТЕРМИНОЛОГИЯ**

|  |
| --- |
| Таблица 2 |

|  |  |
| --- | --- |
| **Термин** | **Определение** |
| **Алгебраическая геометрия** | Раздел алгебры, основным предметом изучения которого являются алгебраические многообразия. |
| **Алгебраическое многообразие** | Множество решений системы алгебраических уравнений над комплексными или действительными числами. |
| **Кодирование** | Процесс преобразования исходной информации в удобную для передачи форму. |
| **Алгеброгеометрический код** | Линейный блочный код, построенный на основе алгебраического многообразия какой-либо системы алгебраических уравнений.  Основной принцип его работы:   * составляется порождающая матрица – базис подпространства кодовых слов; * для кодирования порождающая матрица умножается на вектор – слово, заданное пользователем; * для декодирования происходит проверка кода с помощью проверочной матрицы, для которой порождающая матрица является базисом ядра гомоморфизма линейных подпространств кодовых и кодируемых слов. В случае обнаружения ошибки, последняя исправляется. Если ошибок не найдено, происходит процесс декодирования: матрица, обратная порождающей, умножается на вектор кодового слова. |
| **Расстояние Хэмминга** | Число позиций, в которых два слова одной длины отличаются. |
| **Токен** | Условное название наименьшей значащей группы символов в уравнении. |

# **ПРИЛОЖЕНИЕ 2**

**ОПИСАНИЕ ФОРМАТА .nk**

Файл формата .nk содержит информацию об алгеброгеометрическом коде, созданном в программе Генератор алгеброгеометрических кодов, в сериализованном виде.

Открыть файл формата .nk можно программой Генератор алгеброгеометрических кодов.

Программа Генератор алгеброгеометрических кодов при этом осуществляет корректное открытие только файлов формата .nk, созданных или измененных в этой программе.

# **ПРИЛОЖЕНИЕ 3**

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ**

Описание и функциональное назначение классов.

Таблица 2

|  |  |
| --- | --- |
| Класс | Назначение |
| Code | Класс, представляющий структуру адгеброгеометрического кода. Содержит методы для кодирования и декодирования информации. |
| Equation | Класс, представляющий структуру нелинейного уравнения над полем Z2. |
| TokenizerException | Класс исключения, которое возникает, если происходит какая-либо ошибка в классе Tokenizer. Наследуется от ApplicationException. |
| VariableGroupTokenException | Класс исключения, которое возникает, если индекс при переменной выходит за границы допустимфых значений. Наследуется от TokenizerException. |
| UnexpectedSymbolException | Класс исключения, которое возникает, если в обрабатываемом уравнении встречается недопустимый символ. Наследуется от TokenizerException. |
| ParserException | Класс исключения, которое возникает, если происходит какая-либо ошибка в классе Parser. Наследуется от ApplicationException. |
| UnexpectedTokenException | Класс исключения, которое возникает, если порядок появления токенов уравнения нарушен. Наследуется от ParserException. |
| CodeException | Класс исключения, которое возникает, если происходит какая-либо ошибка в классе Code. Наследуется от ApplicationException. |
| MistakesNumberException | Класс исключения, которое возникает, если происходит попытка декодировать слово, в котором превышено допустимое количество ошибок. Наследуется от CodeException. |
| CodeGeneratingException | Класс исключения, которое возникает, если в сгенерированном коде длина кодового слова меньше или равна длине кодируемого. Наследуется от CodeException. |
| UnknownCodeMessageException | Класс исключения, которое возникает, если происходит попытка декодировать или закодировать строку, содержащую в себе символы, отличные от 0 и 1. Наследуется от ApplicationException. |
| BaseMatrix | Класс, представляющий структуру матрицы. |
| SolutionsMatrix | Класс, представляющий структуру матрицы решений системы нелинейных уравнений над полем Z2. Наследуется от BaseMatrix. |
| MyStatics | Статический класс, содержащий в себе все статические методы, потребовавшиеся для решения поставленной задачи. |
| Parser | Класс, который содержит методы, проверяющие корректность следования токенов друг за другом и делящие уравнение на 2 части: зависящую от переменных и свободную. |
| Solver | Класс, который содержит методы, решающие систему уравнений. |
| Tokenizer | Класс, методы которого производят деление входной строки на токены, если это возможно. |
| Token | Класс, представляющий собой общий вид токенов. |
| VariableGroupToken | Класс, представляющий собой токен-группу переменных. Наследуется от Token. |
| OperationToken | Класс, представляющий собой токен-операции сложения. Наследуется от Token. |
| EquationToken | Класс, представляющий собой токен-знака равенства. Наследуется от Token. |
| ScalarToken | Класс, представляющий собой токен-свободный член. Наследуется от Token. |
| СheckList | Класс, представляющий структуру проверочного листа. Содержит методы заполняющие проверочный лист и сравнивающие его результат с эталонным. |
| CodeDescriptionWindow | Класс, представляющий окно программы с описанием основных характеристик кода. Наследуется от Window. |
| CodeGeneratingWindow | Класс, представляющий окно программы для создания новых алгеброгеометрических кода. Наследуется от Window. |
| CodeWindow | Класс, представляющий окно программы для кодирования и декодирования сообщений. Наследуется от Window. |
| DescribtionWindow | Класс, представляющий окно программы с кратким описанием основных характеристик кода. Наследуется от Window. |
| FQWindow | Класс, представляющий основное справочное окно программы. Наследуется от Window. |
| InfoWindow | Класс, представляющий окно программы с информацией о разработчике. Наследуется от Window. |
| MainWindow | Класс, представляющий главное окно программы. Наследуется от Window. |
| RulesWindow | Класс, представляющий окно программы с описанием правил ввода системы уравнений. Наследуется от Window. |
| SelectCodeWindow | Класс, представляющий окно программы для выбора кода для дальнейшей работы. Наследуется от Window. |
| СhoiceWindow | Класс, представляющий окно программы, запрашивающее подтверждение какого-либо действия. Наследуется от Window. |

# **ПРИЛОЖЕНИЕ 4**

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ**

Таблица 3.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса CodeDescriptionWindow.xaml.cs | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| AGCode | private | Code | Выбранный код | |
| formatter | private | Binary  Formatter | Инструмент для бинарной сериализации | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| CodeDescriptionWindow | public | конструктор | Code AGCode | Конструктор окна CodeDescriptionWindow |
| Previous  Window  Button\_Click | private | void | object sender, RoutedEventArgs e | Возвращается к предыдущему окну. |
| NextWindow  Button\_Click | private | void | object sender, RoutedEventArgs e | Открывает следующее окно. |
| SaveButton\_  Click | private | void | object sender, RoutedEventArgs e | Выполняет сохранение полученного кода. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса CodeGeneratingWindow.xaml.cs  Таблица 3.2 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| groupSize | private | int | Количество переменных в системе уравнений | |
| line | private | string | Строковое представление системы уравнений. | |
| lines | private | string[] | Строковое представление уравнений. | |
| systemOf  Equations | private | List  <Equation> | Система уравнений. | |
| solver | private | Solver | Инструмент для решения системы уравнений. | |
| theAnswer | private | Solutions  Matrix | Матрица из решений системы уравнений. | |
| AGCode | private | Code | Алгеброгеометрический код. | |
| flag | private | bool | Индикатор наличия ошибки в системе уравнений. | |
| regex | private | Regex | Допустимые для ввода символы. | |
| t | private | ToolTip | Подсказка для ползователя | |
| groupSize | private | int | Количество переменных в системе уравнений. | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| Code  Generating  Window | public | конструктор | - | Конструктор окна CodeGeneratingWindow. |
| SizeComboBox\_Selection  Changed | private | void | object sender, SelectionChangedEventArgs e | Заполняет поле groupSize. |
| NextWindow  Button\_Click | private | void | object sender, RoutedEventArgs e | Генерирует алгеброгеометрический код. |
| Previous  Window  Button\_Click | private | void | object sender, RoutedEventArgs e | Возвращает на одно окно назад. |
| HelpButton\_  Click | private | void | object sender, RoutedEventArgs e | Вызывает справку. |
| Equations\_  Preview  TextInput | private | void | object sender, RoutedEventArgs e | Проверяют допустимость символа, который вводится. |
| Свойства | | | | |
| Имя | Модификатор  доступа | Тип | Доступ | Назначение |
| GroupSize | private | int | set | Количество переменных в системе уравнений. |
| GroupSize | public | int | get | Количество переменных в системе уравнений. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса CodeWindow.xaml.cs  Таблица 3.3 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| AGCode | private | Code | Выбранный код | |
| flag | private | bool | Указывает на выбор операции. | |
| regex | private | Regex | Допустимые для ввода символы. | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| CodeWindow | public | конструктор | Code AGCode | Конструктор окна CodeWindow. |
| Previous  Window  Button\_Click | private | void | object sender, RoutedEventArgs e | Возвращает на одно окно назад. |
| CodeButton\_  Click | private | void | object sender, RoutedEventArgs e | Кодирует или декодирует сообщение. |
| CodeOr  DecodeComboBox\_SelectionChanged | private | void | object sender, SelectionChangedEventArgs e | Переключает режим кодирования. |
| First\_Text  Changed | private | void | object sender, TextChangedEventArgs e | Синхронизирует работу полей для кодового и кодируемого сообщения. |
| FirstPreview  KeyDown | private | void | object sender, KeyEventArgs e | Запрещает ввод пробелов. |
| First\_Preview  TextInput | private | void | object sender, TextCompositionEventArgs e | Проверяет допустимость вводимых символов. Допускает ввод только 0 и 1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса FQWindow.xaml.cs  Таблица 3.4 | | | | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| FQWindow | public | конструктор | - | Конструктор окна полной справки |
| Previous  Window  Button\_Click | private | void | object sender, RoutedEventArgs e | Возвращает к предыдущему окну. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса HelpWindow.xaml.cs  Таблица 3.5 | | | | | |
| Методы | | | | | |
| Имя | Модификатор  доступа | | Тип | Аргументы | Назначение |
| HelpWindow | public | конструктор | | Code code | Конструктор окна HelpWindow. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса InfoWindow.xaml.cs  Таблица 3.6 | | | | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| InfoWindow | public | конструктор | - | Конструктор окна информации о разработчике. |
| Previous  Window  Button\_Click | private | void | object sender, RoutedEventArgs e | Возврат к главному окну. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса MainWindow.xaml.cs  Таблица 3.7 | | | | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| MainWindow | public | конструктор | - | Точка входа в программу. Конструктор главного окна MainWindow. |
| startButton\_  Click | private | void | object sender, RoutedEventArgs e | Начало работы. Переход к выбору алгеброгеометрического кода. |
| fqButton\_  Click | private | void | object sender, RoutedEventArgs e | Переход к руководству оператора. |
| infoButton\_  Click | private | void | object sender, RoutedEventArgs e | Переход к информаци о разработчике програмыю |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса RulesWindow.xaml.cs  Таблица 3.8 | | | | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| RulesWindow | public | конструктор | - | Конструктор окна RulesWindow. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса SelectCodeWindow.xaml.cs  Таблица 3.9 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| codes | private | List<Code> | Список сохраненных кодов. | |
| formatter | private | BinaryFormatter | Инструмент для бинарной сериализации | |
| itemName | private | string | Имя выбранного элемента codeSelector. | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| SelectCode  Window | public | конструктор | - | Конструктор окна SelectCodeWindow. |
| previousWindowButton\_  Click | private | void | object sender, RoutedEventArgs e | Открывает предыдущее окно. |
| trashButton\_  Click | private | void | object sender, RoutedEventArgs e | Готовит код к возможному удалению. |
| nextWindow  Button\_Click | private | void | object sender, RoutedEventArgs e | Переходит к окну с характеристиками выбранного кода или к окну для создания нового кода. |
| CreateComboBox | private | void | - | Переписывает текст codeSelector. |
| codeSelector  ComboBox\_  Selection  Changed | private | void | object sender, SelectionChangedEventArgs e | Отмечает, какой код выбран, или открывает окно для создания нового кода. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса СhoiceWindow.xaml.cs  Таблица 3.10 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| codes | private | List<Code> | Список сохраненных кодов. | |
| formatter | private | BinaryFormatter | Инструмент для бинарной сериализации | |
| win | private | SelectCodeWindow | Окно, вызвавшее конструктор СhoiceWindow. | |
| index | private | int | Номер кода в списке сохраненных. | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| Сhoice  Window | public | конструктор | string name, int index, List<Code> codes, SelectCodeWindow win | Конструктор окна СhoiceWindow. |
| YesButton\_  Click | private | void | object sender, RoutedEventArgs e | Удаляет выбранный код. |
| NoButton\_  Click | private | void | object sender, RoutedEventArgs e | Закрывает окно. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса Code.cs  Таблица 3.11 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| generating  Matrix | private | Solutions  Matrix | Порождающая матрица кода. | |
| allCodeWords | private | BaseMatrix | Матрица всех возможных кодовых слов. | |
| t | private | int | Максимальное количество исправляемых ошибок | |
| systemOf  Equations | private | string[] | Система уравнений, порождающая данный код. | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| Code | public | Конструктор | SolutionsMatrix generatingMatrix, string[] line | Создает код на основе полученной матрицы, устанавливая его основные параметры. |
| Encode | public | int[] | int[] ourMessage | Кодирует полученное сообщение. |
| Decode | public | int[] | int[] ourMessage | Декодирует полученное сообщение, исправляя ошибки. |
| ToString | public | string | - | Формирует имя кода в соответствии с основными характеристиками. |
| Свойства | | | | |
| Имя | Модификатор  доступа | Тип | Доступ | Назначение |
| Generating  Matrix | public | SolutionsMatrix | get | Порождающая матрица кода. |
| Generating  Matrix | private | SolutionsMatrix | set | Порождающая матрица кода. |
| AllCodeWords | public | BaseMatrix | get | Матрица всех возможных кодовых слов. |
| AllCodeWords | private | BaseMatrix | set | Матрица всех возможных кодовых слов. |
| K | public | int | get | Длина кодируемого слова. |
| K | private | int | set | Длина кодируемого слова. |
| N | public | int | get | Длина кодового слова. |
| N | private | int | set | Длина кодового слова. |
| T | public | int | get | Максимальное количество исправляемых ошибок. |
| T | private | int | set | Максимальное количество исправляемых ошибок. |
| SystemOf  Equations | public | string[] | get | Система уравнений, порождающая данный код. |
| SystemOf  Equations | private | string[] | set | Система уравнений, порождающая данный код. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса Equation.cs  Таблица 3.12 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| tokenizer | private | Tokenizer | Разбивает строку на токены, характерные для уравнений. | |
| parser | private | Parser | Проверяет возможно ли разбиение строки на токены. | |
| groupSize | private | int | Максимальное количество переменных в уравнении. | |
| tokens | private | List<Token> | Уравнение в памяти. | |
| eq | private | Tuple<List<bool[]>, bool> | Уравнение. | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| Equation | public | Конструктор | string input, int groupSize, int index | Строит в памяти уравнение из полученной строки. |
| Свойства | | | | |
| Имя | Модификатор  доступа | Тип | Доступ | Назначение |
| Eq | public | Tuple<List<bool[]>, bool> | get | Уравнение. |
| Eq | private | Tuple<List<bool[]>, bool> | set | Уравнение. |
| GroupSize | public | int | get | Максимальное количество переменных в уравнении. |
| GroupSize | private | int | set | Максимальное количество переменных в уравнении. |
| VariableTokenCount | public | int | get | Количество токенов. |
| VariableTokenCount | private | int | set | Количество токенов. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса Equation.cs  Таблица 3.13 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| tokenizer | private | Tokenizer | Разбивает строку на токены, характерные для уравнений. | |
| parser | private | Parser | Проверяет возможно ли разбиение строки на токены. | |
| groupSize | private | int | Максимальное количество переменных в уравнении. | |
| tokens | private | List<Token> | Уравнение в памяти. | |
| eq | private | Tuple<List<bool[]>, bool> | Уравнение. | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| Equation | public | Конструктор | string input, int groupSize, int index | Строит в памяти уравнение из полученной строки. |
| Свойства | | | | |
| Имя | Модификатор  доступа | Тип | Доступ | Назначение |
| Eq | public | Tuple<List<bool[]>, bool> | get | Уравнение. |
| Eq | private | Tuple<List<bool[]>, bool> | set | Уравнение. |
| GroupSize | public | int | get | Максимальное количество переменных в уравнении. |
| GroupSize | private | int | set | Максимальное количество переменных в уравнении. |
| VariableTokenCount | public | int | get |  |
| VariableTokenCount | private | int | set |  |
| Описание полей методов и свойств класса BaseMatrix.cs  Таблица 3.14 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| matrix | private | List<int[]> | Матрица | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| Индексатор | public | int | int i, int j | Обращается к элементу матрицы. |
| BaseMatrix | public | Конструктор | - | Конструктор пустой матрицы. |
| BaseMatrix | public | Конструктор | int[] ourLine | Конструктор матрицы. |
| ToString | public | string | - | Возвращает матрицу в строковом формате |
| Свойства | | | | |
| Имя | Модификатор  доступа | Тип | Доступ | Назначение |
| Matrix | public | List<int[]> | get | Матрица |
| Matrix | private | List<int[]> | set | Матрица |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса MyStatics.cs  Таблица 3.15 | | | | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| ToBinaryInt  Array | public | int[] | int value, int size | Представляет число в двоичном виде. |
| Multiplication | public | BaseMatrix | BaseMatrix A, BaseMatrix B | Перемножает матрицы. |
| ToIntArray | public | int[] | string input, int size | Преобразует строковое сообщение в массив |
| FindMin  Distance | public | int | BaseMatrix allCodeWords | Ищет минимальное расстояние в коде. |
| Reading | public | void | string input, int groupSize, List<Equation> systemOfEquations, int index | Считывает уравнение и добавляет его в систему уравнениц. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса Parser.cs  Таблица 3.16 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| START | private | const int | Константа, обозначающая тип предыдущего токена | |
| AFTER\_  VARIABLE  GROUP | private | const int | Константа, обозначающая тип предыдущего токена | |
| AFTER\_  OPERATION | private | const int | Константа, обозначающая тип предыдущего токена | |
| AFTER\_  EQUATION | private | const int | Константа, обозначающая тип предыдущего токена | |
| AFTER\_  SCALAR | private | const int | Константа, обозначающая тип предыдущего токена | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| Parse | public | Tuple<List<bool[]>, bool> | List<Token> tokens, int index | Проверка правильности порядка токенов и преобразование их в уравнение |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса Solver.cs  Таблица 3.17 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| matrix | private | SolutionsMatrix | Матрица решений. | |
| checkingList | private | СheckList | Проверочный лист. | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| Solve | public | SolutionsMatrix | List<Equation> equationsSystem | Решает систему уравнений. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса Tokenizer.cs  Таблица 3.18 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| group\_size | public | int | Количество различных переменных в системе уравнений. | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| Tokenizer | public | Конструктор | int group\_size | Конструктор |
| SplitToTokens | public | List<Token> | string input, int mainIndex | Разбивает строку на токены, характерные для уравнений. |
| Описание полей методов и свойств класса СheckList.cs  Таблица 3.19 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| checkingList | private | bool[] | Результат подставления в систему уравнений конкретного решения. | |
| prototype | private | List<Equation> | Система уравнений, которую будут проверять данным проверочным листом. | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| Индексатор | public | bool | int i | Возвращает значение уравнения с заданным индексом для заданной системы уравнений и заданного решения. |
| СheckList | public | Конструктор | List<Equation> prototype | Создание проверочного листа на основе системы уравнений. |
| FillingThe  CheckList | public | void | int[] possibleSolution | Заполняет проверочный лист для конкретного решения. |
| IsItRight  Solution | public | bool | - | Проверяет, совпадает ли правильный ответ к системе с полученным в проверочном листе. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса VariableGroupToken (Tokens.cs)  Таблица 3.20 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| variables | public | bool[] | Массив, показывающий наличие или отсутствие каждой возможной для системы уравнений переменной в текущей группе. | |
| size | public | int | Количество переменных в текущей группе. | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| VariableGroupToken | public | Конструктор | bool[] variables | Конструктор токена-группы переменных. |
| Описание полей методов и свойств класса ScalarToken (Tokens.cs)  Таблица 3.21 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| variable | public | bool | Значение свободного члена. | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| ScalarToken | public | Конструктор | bool variable | Конструктор токена-свободного члена. |
| ToString | public | string | - | Строковое представление токена-свободного члена. |

# **ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированх |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |