**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |
| --- | --- |
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. №подл |  |

СОГЛАСОВАНО

Доцент департамента программной инженерии, кандидат технических наук

Р.З. Ахметсафина

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

УТВЕРЖДАЮ

Академический руководитель образовательной программы «Программная инженерия», к.т.н., профессор ДПИ ФКН

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. В. Шилов

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

**ВЕБ ПРИЛОЖЕНИЕ ДЛЯ ГЕНЕРАЦИИ**

**ЗАДАЧ ПО АЛГЕБРЕ**

**Пояснительная записка**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.05.15-01 81 01-1-ЛУ 1**

**Исполнитель**

Студент группы БПИ199

\_\_\_\_\_\_\_\_\_\_\_\_ / К. Пике /

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

**Москва 2021**

**УТВЕРЖДЁН**

**RU.17701729.05.15-01 81 01-1 ЛУ**

|  |  |
| --- | --- |
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. №подл |  |

**ВЕБ ПРИЛОЖЕНИЕ ДЛЯ ГЕНЕРАЦИИ**

**ЗАДАЧ ПО АЛГЕБРЕ**

**Пояснительная записка**

**RU.17701729.05.15-01 81 01-1**

**Листов 44**

**Москва 2021**

[1 Введение 4](#__RefHeading___Toc5482_1933433957)

[1.1 Наименование приложения 4](#__RefHeading___Toc5484_1933433957)

[1.2 Документы, на основании которых ведётся разработка 4](#__RefHeading___Toc5486_1933433957)

[2 Назначение и область применения 5](#__RefHeading___Toc5488_1933433957)

[2.1 Функциональное назначение 5](#__RefHeading___Toc5490_1933433957)

[2.2 Эксплуатационное назначение 5](#__RefHeading___Toc5492_1933433957)

[3 Технические характеристики 6](#__RefHeading___Toc5494_1933433957)

[3.1 Постановка задачи на разработку приложения 6](#__RefHeading___Toc5496_1933433957)

[3.2 Описание применяемых технологических методов. 6](#__RefHeading___Toc5498_1933433957)

[3.3 Алгоритм перемешивания содержимого матриц. 6](#__RefHeading___Toc5500_1933433957)

[3.3.1 Простая перетасовка. 6](#__RefHeading___Toc5502_1933433957)

[3.3.2 Сильная перетасовка. 6](#__RefHeading___Toc5504_1933433957)

[3.3.3 Слабая перетасовка. 7](#__RefHeading___Toc5506_1933433957)

[3.4 Алгоритм формирования случайной ортогональной матрицы 7](#__RefHeading___Toc5508_1933433957)

[3.5 Алгоритм составления задач. 7](#__RefHeading___Toc5510_1933433957)

[3.5.1 Алгоритм составления задач на тему «Сложение, вычитание, умножение матриц». 7](#__RefHeading___Toc5512_1933433957)

[3.5.2 Алгоритм составления задач на тему «Поиск ранга матрицы». 8](#__RefHeading___Toc5514_1933433957)

[3.5.3 Алгоритм составления задач на тему «Поиск n-ой степени матрицы» 8](#__RefHeading___Toc5516_1933433957)

[3.5.4 Алгоритм составления задач на тему «Определение обратимости матрицы». 9](#__RefHeading___Toc5518_1933433957)

[3.5.5 Алгоритм составления задач на тему «Нахождение собственных значений матрицы» 9](#__RefHeading___Toc5520_1933433957)

[3.5.6 Алгоритм составления задач на тему «Приведение матрицы к канонической Жордановой форме» 9](#__RefHeading___Toc5522_1933433957)

[3.5.7 Алгоритм составления задач на тему «QR разложение матрицы» 10](#__RefHeading___Toc5524_1933433957)

[3.5.8 Алгоритм составления задач на тему «Нахождение ортогонального преобразования для перехода к диагональной матрице». 10](#__RefHeading___Toc5526_1933433957)

[3.5.9 Алгоритм составления задач на тему «Проведение сингулярного разложения» 10](#__RefHeading___Toc5528_1933433957)

[3.5.10 Алгоритм составления задач на тему «Нахождение угла и оси вращения ортогонального оператора». 11](#__RefHeading___Toc5530_1933433957)

[3.5.11 Алгоритм составления задач на тему «Нахождение размерности ядра и отображения матрицы перехода». 11](#__RefHeading___Toc5532_1933433957)

[3.6 Обоснование выбора алгоритмов решения задачи. 11](#__RefHeading___Toc5534_1933433957)

[3.7 Возможные взаимодействия приложения с другими программами. 12](#__RefHeading___Toc5536_1933433957)

[3.8 Описание и обоснование выбора метода организации входных и выходных данных. 12](#__RefHeading___Toc5538_1933433957)

[3.8.1 Описание метода организации входных и выходных данных. 12](#__RefHeading___Toc5540_1933433957)

[3.8.2 Обоснование выбора метода входных и выходных данных. 13](#__RefHeading___Toc5542_1933433957)

[4 Описание и обоснование выбора технических и программных средств. 14](#__RefHeading___Toc5544_1933433957)

[4.1 Состав технических и программных средств. 14](#__RefHeading___Toc5546_1933433957)

[4.2 Состав программных средств, необходимых для работы системы. 14](#__RefHeading___Toc5548_1933433957)

[5 Ожидаемые технико-экономические показатели. 15](#__RefHeading___Toc5550_1933433957)

[5.1 Ориентировочная экономическая эффективность 15](#__RefHeading___Toc5552_1933433957)

[5.2 Предполагаемая потребность 15](#__RefHeading___Toc5554_1933433957)

[5.3 Преимущества разработки по сравнению с отечественными и зарубежными аналогами. 15](#__RefHeading___Toc5160_46856381)

[6 Источники 16](#__RefHeading___Toc5556_1933433957)

[Терминология 17](#__RefHeading___Toc5558_1933433957)

[ПРИЛОЖЕНИЕ 1 25](#__RefHeading___Toc5560_1933433957)

[ПРИЛОЖЕНИЕ 2 27](#__RefHeading___Toc5562_1933433957)

# **Введение**

## Наименование приложения

Наименование приложения: «Веб приложение для генерации задач по алгебре.» / «Web application for algebra problems generation».

Краткое наименование приложения: «Algebrator».

## Документы, на основании которых ведётся разработка

Разработка ведётся на основании учебного плана подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утверждённой академическим руководителем тема курсового проекта

# **Назначение и область применения**

## Функциональное назначение

Функциональным назначением приложения является генерация задач по алгебре с ответами на несколько тем на выбор и дальнейшим предоставлением возможности экспорта задач для работы с ними. В возможности веб приложения должно входить:

1. Генерация задач на одну из тем:
   1. Сложение/Вычитание/Умножение матриц
   2. Поиск ранга матрицы
   3. Поиск n-ой степени матрицы
   4. Поиск обратной матрицы
   5. Приведение матрицы к канонической Жордановой форме
   6. QR разложение матрицы
   7. Нахождение ортогонального преобразования для перехода к диагональной матрице
   8. Проведение сингулярного разложения
   9. Нахождение угла и оси вращение ортогонального оператора
   10. Нахождение размерности ядра и отображения матрицы перехода
2. Составление списка задач
3. Сброс составленного списка
4. Экспорт задач в формате .pdf
5. Экспорт задач в формате .tex

## Эксплуатационное назначение

Генерация задач на какую-либо тему изначально предполагает образовательные цели. Данное приложение предназначено для студентов и учителей в целях самостоятельной практики решения задач, составления домашних и/или контрольных работ.

# **Технические характеристики**

## Постановка задачи на разработку приложения

Программа разработана в рамках выполнения курсовой работы за 2020-2021 учебный год.

Цель работы предоставить веб приложение, отвечающее функциональному назначению, описанному в **2.1.**

## Описание применяемых технологических методов.

Приложение разрабатывалось при помощи фреймворков Spring Boot, Boostrap 5, шаблонизатор Freemarker.

Для работы с фреймворком Spring Boot использовался язык Java.

Для работы с клиентской частью приложения использовался HTML5.

## Алгоритм перемешивания содержимого матриц.

Для некоторых из задач, мы можем провести рандомизацию строк матрицы при помощи применения к ним элементарных преобразований, то есть сложение, перемножение и обмен местами строк. Умножение строк было опущено, дабы как можно больше избегать дробей в ответах и/или задачах.

В программе приложения присутствуют алгоритмы простой перетасовки, слабой перетасовки и сильной перетасовки.

### Простая перетасовка.

Строки в матрице берутся в случайном порядке и далее складываются каждая с отличной от себя в этом порядке с коэффициентами в определённом разбросе. Далее, строки в том же случайном порядке переставляются местами.

### **Сильная** перетасовка.

Работает по тому же принципу, что и алгоритм простой перетасовки, описанный в **3.2.1.** , с тем отличием, что на выходе возвращается матрица, которая отменяем все проведённые преобразования. То есть в случае, когда на вход подаётся единичная матрица, мы получаем на выходе обратную марицу полученной.

### Слабая перетасовка.

Строкам матрицы присваивается случайный порядок и далее две отличные друг от друга строки прибавляются со случайным коэффициентом к одной из других, отличных от этих двух строк.

На выход получается матрица, которая может обратить все произведённые изменения, как было описано в пункте **3.3.2.**

## Алгоритм формирования случайной ортогональной матрицы

Так как ортогональные матрицы формулируются путём перемножения уже ортогональных матриц с ортогональными матрицами, изначальной идеей было взять случайные матрицы поворота и перемножать с ними единичную матрицу размера 3 на 3. Однако такой подход приводил либо к слишком простым значениям матрицы, либо к слишком сложным.

Поэтому для генерации случайных ортогональных матриц была взята за основу формула из статьи о рациональных автоморфов

В данной статье упоминалось, что для нечётного числа формата , где a,b,c,d случайные целые числа, существует ортогональная матрица, которая задаётся следующим выражением:

(1)

Таким образом, мы генерировали 3 случайных чётных числа и одно нечётное, так как квадрат на чётность не влияет.

## Алгоритм составления задач.

Каждый из тип задачей требует аргумент — число, при помощи которого инициализируется генератор псевдослучайных чисел для каждой из задач.

### **Алгоритм составления задач на тему «Сложение, вычитание, умножение матриц».**

Пользуясь генератором псевдослучайных чисел, мы задаём случайным образом операцию сложения, вычитания или умножения. Далее, пользуясь все тем же генератором, мы задаём размер для обеих матриц в случае сложения или вычитания. В случае умножения, для второй матрицы мы устанавливаем высоту равную ширине первой матрицы и случайным образом подбираем ей ширину.

Значения внутри этих матриц задаются случайным образом, в разбросе от -5 до 5 для всех членов матрицы, чтобы не иметь дело со слишком большими цифрами.

Результат этой задачи получается путём произведения подобранной операции (сложение, вычитание, умножение).

### Алгоритм составления задач на тему «Поиск ранга матрицы».

Пользуясь генератором псевдослучайных чисел, мы подбираем случайным образом размеры матрицы и ранг. Далее составляем матрицу из случайных значений с искомым рангом. Для этого мы заполняем её строками, где на каждый шаг, пивотальный элемент сдвигается на определённое количество шагов вправо.

Далее мы перемешиваем содержимое матрицы при помощи простой перетасовки и таким образом формируем задачу данного типа.

### Алгоритм составления задач на тему «Поиск n-ой степени матрицы»

Для составления задачи такого типа воспользуемся формулой

(2)

где А это невырожденная матрица, а D диагональная. Тогда задача по нахождению сводится к и при целых значениях матрицы А, можно получить матрицу , которую можно вычислить по индукции.

Пользуясь генератором псевдослучайных чисел, мы генерируем размер матрицы, так как для данной задачи, она должна быть квадратной, то для этого нам достаточно одного числа. Далее мы задаём единичную матрицу А, перетасовываем её слабой перетасовкой, чтобы не переусложнить задачу. После этой перетасовки, мы получаем так же и обратную матрицу к матрице А.

Далее генерируем случайную диагональную матрицу, и производим вычисления по формуле (2).

### Алгоритм составления задач на тему «Определение обратимости матрицы».

Для генерации этого типа задач мы пользуемся тем свойством матриц, что вырожденные матрицы остаются вырожденными после применения элементарных операций, а невырожденные — невырожденными.

Мы генерируем матрицу размера 3 на 3, так как размер больше или меньше не имеет смысла или пользы для вычисления. Случайным образом определяем, будет ли она невырождена. Если будет, то мы перетасовываем единичную матрицу и получаем её обратную матрицу, в ином случае, задаём матрицу случайного ранга, меньшего чем её размер и перетасовываем её.

### Алгоритм составления задач на тему «Нахождение собственных значений матрицы»

Из разложения матрицы с помощью собственных векторов, нам известно, что для квадратной матрицы А из линейно независимых собственных векторов можно получить разложение

(3)

где матрица D диагональная с собственными значениями А по диагонали.

Таким образом для получения матрицы А мы можем сгенерировать диагональную матрицу со случайными значениями по диагонали и обратимую матрицу Q при помощи сильной перетасовки и перемножить их соответственно формуле (3).

### Алгоритм составления задач на тему «Приведение матрицы к канонической Жордановой форме»

Из определения Жордановой нормальной формы, мы знаем, что существует такая матрица P, что для матрицы А верно

(4)

где J это матрица Жордановой канонической формы.

Таким образом единственное условие, которое нужно наложить на матрицу P, это чтобы она была невырождена, то есть мы создаём единичную, перемешиваем ей ряды и получаем обратную матрицу.

Для матрицы J, мы просто составляем случайным образом Жордановы блоки и производим вычисления по формуле (4).

### Алгоритм составления задач на тему «QR разложение матрицы»

QR разложение это представление матрицы а в форме , где Q это ортогональная матрица, а R — верхнетреугольная. Для того, чтобы разложение матрицы А было уникальным, мы так же накладываем ограничение на верхнетреугольную матрицу, чтобы у неё все элементы на диагонали были положительными.

Для генерации матрицы Q мы воспользовались алгоритмом, описанным в **3.4.** Генерация верхнетреугольной матрицы проходит по её определению со случайными значениями.

### **Алгоритм составления задач на тему «Нахождение ортогонального преобразования для перехода к диагональной матрице».**

Матрицу, которую можно привести к диагональной при помощи ортогонального преобразования, можно представить в следующем виде:

(5)

где D — диагональная матрица, Т — ортогональная матрица.

Мы пользуемся алгоритмом описанным в **3.4** чтобы сгенерировать ортогональную матрицу Т и составляем случайную диагональную матрицу в качестве матрицы D и с их помощью получаем матрицу, к которой даётся задание.

### Алгоритм составления задач на тему «Проведение сингулярного разложения»

У любой вещественной квадратной матрицы существуют две ортогональные матрицы U и V, такие что соблюдается следующее выражение

(6)

где D это диагональная матрица с собственными значениями А по диагонали.

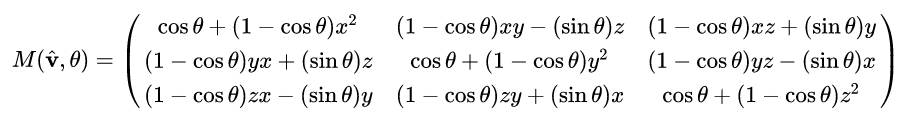
Таким образом мы генерируем две случайные ортогональные матрицы и случайную диагональную матрицу. Так как по формуле из статьи Гордона Пола ортогональная матрица получается с каким-то знаменателем для всех членов, чтобы избавиться от дробей в итоговой матрице для задачи, мы домножаем члены диагональной матрицы на знаменатели матриц U и V.

Однако в случае квадратной матрицы, при сингулярном разложении А, матрицы U и V не уникальны, так как

где S, F, W — это ортогональные матрицы. Ответ на основе которого генерируется матрица не является единственным возможным.

### Алгоритм составления задач на тему «Нахождение угла и оси вращения ортогонального оператора».

Для составления матрицы поворота вокруг произвольной оси была использована формула из википедии

Формула матрицы поворота вокруг произвольной оси

где значения вектора задано случайным образом и значение угла поворота выбирается случайным образом из табличных значений, кроме 0, так как угол поворота 0 делает все слишком очевидным.

### Алгоритм составления задач на тему «Нахождение размерности ядра и отображения матрицы перехода».

Мы хотим составить задачу вида

где А — невырожденная матрица и Х — матрица перехода

Матрицу перехода мы генерируем как матрицу случайного ранга. Матрицу А мы генерируем как случайную диагональную матрицу и перетасовываем её. Далее подставляем их в выражение и находим матрицу В для задачи.

## Обоснование выбора алгоритмов решения задачи.

При генерации задачи, зачастую можно избежать прохождения шагов для ее решения. Можно отталкиваться от уже готовых ответов (**3.7, 3.9, 3.11, 3.13 — 3.16**) или опираясь на свойства тех матриц в контексте задания (**3.6-3.11**).

Таким образом можно заранее подготовить ответ и исключить случаи, когда его может не оказаться.

## Возможные взаимодействия приложения с другими программами.

Взаимодействия с другими программами отсутствуют.

## Описание и обоснование выбора метода организации входных и выходных данных.

### Описание метода организации входных и выходных данных.

В приложении клиента, пользователь имеет возможность взаимодействовать с приложением при помощи кнопок на экране.

При нажатии на кнопку «Сгенерировать задачу», на открывается вкладыш с разными названиями тем. При нажатии на область с одной из тем, на серверную часть часть отправляется GET запрос на отображение задачи на данную тему на экране.

Нажатие на кнопку «Другая задача» обновляет страницу и генерирует таким образом новую задачу на текущую тем.

Нажатие на кнопку «Добавить задачу» отправляет на сервер запрос на добавление текущей задачи в финальный документ задач на экспорт и обновляется страница.

Нажатие на кнопку «Показать ответ» открывает часть текста, на которой написан ответ на текущую задачу.

Нажатие на кнопку «Сбросить текущий выбор» (возможно только при непустом списке задач) отправляет POST запрос на серверную часть, где текущий документ очищается и страница затем обновляется с новой задачей по текущей тематике.

Нажатие на кнопку «Экспортировать задачи и ответы» отправляет на серверную часть запрос GET, ответом на который является загрузка файла с задачами и ответами.

той же теме, для скрытия/ раскрытия ответа, добавления задачи в список экспорта, очищения списка на экспорт и сам экспорт задач.

Выходные данные представляют из себя отображение задач на экране пользователя в коне браузера или же в случае со списком задач, который возможно экспортировать из веб приложения представляет из себя .zip файл, который в свою очередь содержит 4 файла

1. Файл с условием задач в формате .pdf
2. Файл с условием задач в формате .tex
3. Файл с ответам на задачи в формате .pdf
4. Файл с ответами на задачи в формате .tex

### Обоснование выбора метода входных и выходных данных.

Взаимодействие с пользователем через нажатие кнопок позволяет генерировать и обновлять выбор задач пользователем с минимальным количеством действий.

Выходные данные были организованы в виде отображения задач т, чтобы у пользователя была возможность оценить, хочет ли он добавлять ту или иную задачу.

Экспорт данных был организован в формате .zip, содержащим 4 файла в форматах указанных в **3.8.1.** , для того, чтобы пользователь имел возможность решать и смотреть ответы к задачам в раздельном порядке.

К файлам формата .pdf были добавлены файлы в формате .tex на случай если пользователь захочет добавить свои какие-то задачи в список, которые не предоставлены в веб приложении.

# Описание и обоснование выбора технических и программных средств.

## Состав технических и программных средств.

Состав технических средств, необходимых для работы системы:

1. Серверная часть:
   1. Рекомендуется от 512 Мб ОЗУ или более
   2. Место на жёстком диске: от 4 Гб
   3. Доступ к интернету
   4. Открытый порт 80
   5. Наличие доступа по ssh
2. Клиентская часть:
   1. Устройство поддерживающее выход в интернет
   2. Рекомендуемая диагональ экрана от 7 дюймов и выше
   3. Мышь или совместимое указывающее устройство и клавиатура или тачскрин.

## Состав программных средств, необходимых для работы системы.

1. Серверная часть:
   1. Операционная система Linux (Ubuntu 20.04)
   2. Java 11
   3. nginx
   4. Компилятор файлов формата .tex
   5. Языковой пакет для компиляции файлов формата .tex содержащих кириллицу
2. Клиентская часть
   1. Браузер, поддерживающий HTML5 спецификацию

# **Ожидаемые технико-экономические показатели.**

## Ориентировочная экономическая эффективность

Для данного проекта подсчёт экономической эффективности не предусмотрен.

## Предполагаемая потребность

Предполагается, что приложением будут пользоваться студенты и преподавали для практики предоставленных задач по алгебре и/или составления домашних/контрольных работ.

## **П**реимущества разработки по сравнению с отечественными и зарубежными аналогами.

Быстрый поиск в интернете выявил следующие отечественные аналоги:

1. Генератор примеров по математике URL: http://l1158.ru/generator/main.php/
2. Библиотека Московской Электронной Школы
3. Генератор задач 1Gb URL: http://generatorzadach.1gb.ru/readme.htm
4. Android — приложение «Математика: Генератор задач»

Так же уже была проведена аналогичная работа студентами МГТУ им. Н.Э. Баумана на тему «Генератор контрольных заданий по высшей математике: опыт создания и применения». Однако данные аналоги не затрагивают те же темы для генерации задач, что и данное веб приложение.

Поиск зарубежных аналогов выявил, что самым близким по содержанию задач является «Wolfram Problem Generator».

Темы генерации «Сложение/Вычитание умножение матриц», «Поиск обратной матрицы», «Поиск собственных значений» совпадают с темами данного приложения. Однако они не затрагивают остальные темы веб приложения «Algebrator».

Таким образом, преимуществом приложения «Algebrator» в сравнении с «Wolfram Problem Generator» являются:

1. Темы затрагивают более сложные задачи по курсу алгебры.
2. Возможность генерации листа с задачами без платной подписки.

# Источники

1. ГОСТ 19.101-77 Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. Singular value decomposition [Электронный ресурс] URL: https://en.wikipedia.org/wiki/Singular\_value\_decomposition режим доступа — свободный, дата обращения 25.04.2020
11. Gordon Paul, On the rational automorphs of .Annals of mathematics vol.41, No. 4, October, 1940, 754 — 766. http://zakuski.math.utsa.edu/~kap/Pall\_Automorphs\_1940.pdf
12. Jordan normal form [Электронный ресурс] URL: https://en.wikipedia.org/wiki/Jordan\_normal\_form. Режим доступа — свободный. Дата обращения 30.04.2020
13. Linear algebra for scientific thinkers: Inverse matrix and matrix algebra [Электронный ресурс] URL: https://people.math.carleton.ca/~kcheung/math/notes/MATH1107/index.html дата обращения : 26.04.2020

# **Терминология**

|  |  |
| --- | --- |
| Термин | Определение |
| Веб приложение | Клиент серверное приложение, в котором клиент взаимодействует с веб сервисом при помощи браузера. |
| Ранг матрицы | Рангом системы строк (столбцов) матрицы A с m строк и n столбцов называется максимальное число линейно независимых строк (столбцов). |
| Жорданова нормальная форма | Квадратичная блочно-диагональная матрица, где каждый блок является жордановой клеткой. |
| QR разложение | Представление матрицы в виде произведения унитарной (или ортогональной матрицы) и верхнетреугольной матрицы. |
| Ортогональная матрица | Квадратная матрица A с вещественными элементами, результат умножения которой на транспонированную матрицу равен единичной матрице |

# ПРИЛОЖЕНИЕ 1

**Описание классов**

|  |  |
| --- | --- |
| **Класс** | **Назначение** |
| TasksController | Класс контроллера, который отвечает на все запросы клиента |
| Column | Класс столбцов матрицы |
| Row | Класс рядов матрицы |
| Matrix | Класс матрицы |
| DimKer | Класс задач на нахождение размерности ядра и отображения матрицы перехода |
| FindEignevalues | Класс задач на нахождение собственных значений |
| InverseMatrix | Класс задач на нахождение обратной матрицы |
| JordanCanonical | Класс задач на нахождение жордановой нормальной формы |
| MatrixAddSubMult | Класс задач на сложение/вычитание или умножение |
| MatrixPowerN | Класс задач на нахождение n-ой степени матрицы |
| MatrixProblem | Абстрактный класс задач, содержащих матрицу |
| MatrixProblemFactory | Класс фабрики задач |
| MatrixRank | Класс задач на нахождение ранга матрицы |
| OrthogonalDiag | Класс задач на нахождение ортогонального приведение к диагональной форме |
| Qrdecomposition | Класс задач на QR разложение |
| RotationAroundAxis | Класс задач на нахождения матрицы поворота и оси вращения |
| SVDdecomposition | Класс задач на сингулярное разложение |
| AlgebraGeneratorApplication | Класс для запуска приложения |
| TasksDocument | Класс для хранения задач |
| ExprUtils | Класс дополнительных утилит для объектов с интерфейсом IExpr |

# ПРИЛОЖЕНИЕ 2

Описание класса TasksController Таблица 2.1

| **Поля** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| document | | private | | TasksDocument | | Хранит список задач на экспорт | |
| currentProblem | | private | | MatrixProblem | | Хранит текущую задачу | |
| currentType | | private | | String | | Хранит текущий тип задач | |
| **Методы** | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| greeting | public | | String | | Map<String,Object> | | Перенаправляет пользователя на страницу содержащую задачу на сложение/вычитание умножение матриц |
| problem | public | | String | | Map<String, Object>, String | | Берет аргумент из запроса и по нему генерирует случайную задачу на тему из аргумента |
| add | public | | String | | Map<String, Object>, String | | Пост запрос, который добавляет текущую задачу в список на экспорт и обновляет текущую задачу |
| download | public | | ResponseEntity<byte[]> | |  | | Создаёт .tex файлы из задач и ответов, компилирует их в формат .pdf и возвращает для скачивания в формате .zip |
| createZip | private | | void | | String, ArrayList<String>, ArrayList<String> | | Создаёт .zip файл на основе путей до файлов, которые в него включены, по определённому пути |
| getRandomName | private | | String | | Random | | Генерирует случайное имя из 16 цифра для того, чтобы назвать файл с задачами |
| endSession | public | | String | |  | | Обновляет документ с задачами в поле, перенаправляет на корневую ссылку |
| reset | public | | String | |  | | Обновляет документ с задачами в поле, перенаправляет на корневую ссылку |

Описание класса Column Таблица 2.2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| content | | private | | ArrayList<Iexpr> | | Содержит значения столбца | |
| size | | private | | int | | Содержит размер столбца | |
| **Методы** | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| isWeak | public | | boolean | |  | | Оценивает, достаточно ли ненулевых значений в столбце |
| Column | public | | Column | | ArrayList<Iexpr> | | Конструктор |
| mult | Public | | Column | | IExpr | | Умножает содержимое столбца на выражение |
| add | public | | Column | | Column,IExpr | | Прибавляет к текущему столбцу столбец, умноженный на выражение |
| getSize | public | | int | |  | | Получает размер столбца |
| getContent | public | | ArrayList<Iexpr> | |  | | Получает список выражений внутри столбца |
| set | public | | void | | int,IExpr | | Задаёт элемент столбцу по индексу |
| get | public | | IExpr | | int | | Получаем элемент столбца по индексу |
| toString | public | | String | |  | | Переводит столбец в формат строки |
| toRow | public | | Row | |  | | Трансформирует столбец в строку |

Описание класса Matrix Таблица 2.3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | | | | | | | |
| Имя | Модификатор доступа | | | Тип | | Назначение | |
| rows | private | | | ArrayList<Row> | | Строки матрицы | |
| cols | private | | | ArrayList<Column> | | Столбцы матрицы | |
| width | private | | | int | | Ширина матрицы | |
| Height | private | | | int | | Высота матрицы | |
| **Методы** | | | | | | | |
| Имя | | Модификатор доступа | Тип | | Аргументы | | Назначение |
| Matrix | | public | Matrix | | ArrayList<Row> | | Конструктор |
| add | | public | Matrix | | Matrix | | Прибавляет одну матрицу к другой и возвращает результат |
| addRow | | public | void | | Int,int,IExpr | | Добавляет одну строку к другой с коэффициентом |
| convertRowsToCols | | private | void | |  | | Переводит строки в столбцы |
| diag | | public | Matrix | | Int, ArrayList<Iexpr> | | Создаёт диагональную матрицу с элементами по диагонали |
| elementaryOpAdd | | private | Matrix | | Int,int, Iexpr, int | | Создаёт матрицу в которой провели элементарную операцию сложения строк |
| get | | public | IExpr | | Int, int | | Возвращает элемент матрицы по индексу |
| getId | | private | Int | | Int, int, Random | | Возвращает случайный индекс строки, отличный от текущего |
| identity | | public | int | | Matrix | | Создает единичную матрицу |
| mult | | public | Matrix | | Matrix | | Перемножает две матрицы и возвращает результат |
| multCurrent | | private | void | | Matrix | | Умножает текущую матрицу и изменяет ее |
| multRow | | public | void | | Int,IExpr | | Умножает заданный ряд на выражение |
| ofRank | | public | Matrix | | Int,int, int, Random | | Создает матрицу заданного ранга |
| orthogonal | | public | Matrix | | Random, ArrayList<IExpr> | | Создает ортогональную матрицу со значениями a,b,c,d для алгоритма |
| randDiag | | public | Matrix | | Int, Random | | Создает случайную ортогональную матрицу |
| randomMatrix | | public | Matrix | | Random, int, int, int, int | | Создает случайную матрицу |
| setCol | | private | void | | Int, int, IExpr | | Задает в столбце по идексу, элементу по индексу, значение |
| simpleShuffle | | public | void | | Random, int, int | | Простая перетасовка, которая не возвращает обратную матрицу |
| strongShuffle | | public | Matrix | | Random, int, int, int | | Сильная перетасовка, которая возвращает обратную матрицу |
| weakShuffle | | public | Matrix | | Random ,int, int | | Слабая перетасовка, которая возвращает обратную матрицу |
| sub | | public | Matrix | | Matrix | | Вычитание матриц |
| swapRow | | public | void | | Int, int | | Меняет местами строки по идексам |
| toString | | public | String | |  | | Приводит к виду строки в формате Latex |
| transpose | | public | Matrix | |  | | Возвращает транспонированную матрицу от текущей |
| updateCols | | private | void | | int | | Обновляет значения столбца по индексу |

Описание класса Row Таблица 2.4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| content | | private | | ArrayList<IExpr> | | Содержимое строки | |
| size | | private | | int | | Размер строки | |
| **Методы** | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| add | public | | Row | | Row | | Добавляет строку к текущей |
| add | public | | Row | | Row, IExpr | | Добавляет строку к текущей, умноженной на коэффицент |
| clone | protected | | Object | |  | | Клонирует строку |
| get | public | | IExpr | | int | | Получает элемент строки по индексу |
| getSize | public | | int | |  | | Получает размер строки |
| mult | public | | IExpr | | Column | | Умножает строку на столбец и возвращает выражение |
| set | public | | void | | Int, IExpr | | Задает элементу строке по индексу значение |
| sub | public | | Row | | Row | | Вычитает из текущей строки строку |
| toString | public | | String | |  | | Переводит строку в формат Latex |
| Row | public | | Row | | ArrayList<Iexpr> | | Конструктор |

Описание класса DimKer Таблица 2.4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| A | | private | | Matrix | | Матрица из которой происходит переход | |
| B | | private | | Matrix | | Матрица в которую происходит переход | |
| X | | private | | Matrix | | Матрица перехода | |
| rank | | private | | int | | Ранг матрицы перехода | |
| problemText | | private | | String | | Строка с текстом задачи | |
| **Методы** | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| DimKer | public | | DimKer | | int | | Конструктор |
| getAnswersContent | public | | Stirng | |  | | Возвращает строку, содержащую выражение ответа |
| getAnswerText | public | | String | |  | | Возвращает строку содержащую текст ответа |
| getProblemContent | public | | String | |  | | Возвращает строку с выражение задачи |
| getProblemText | public | | String | |  | | Возвращает строку с текстом задачи |
| initA | private | | Matrix | |  | | Инициализирует матрицу А |
| initB | private | | Matrix | |  | | Инициализирует матрицу В |
| initX | private | | Matrix | |  | | Инициализирует матрицу Х |

Описание класса FindEigenvalues Таблица 2.6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| eigenvalues | | private | | ArrayList<IExpr> | | Собственные значения | |
| matrix | | private | | Matrix | | Матрица содержимого задачи | |
| A | | private | | Matrix | | Невырожденнная матрица | |
| invA | | private | | Matrix | | Обратная матрица матрицы А | |
| problemText | | private | | String | | Текст задачи | |
| answerText | | private | | String | | Текс ответа | |
| **Методы** | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| getAnswersContent | public | | Stirng | |  | | Возвращает строку, содержащую выражение ответа |
| getAnswerText | public | | String | |  | | Возвращает строку содержащую текст ответа |
| getProblemContent | public | | String | |  | | Возвращает строку с выражение задачи |
| getProblemText | public | | String | |  | | Возвращает строку с текстом задачи |

Описание класса InverseMatrix Таблица 2.7

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| matrix | | private | | Matrix | | Матрица задачи | |
| inverse | | private | | Matrix | | Обратная матрица матрице задачи | |
| isInvertible | | private | | boolean | | Определяет, обратима ли матрица задачи | |
| **Методы** | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| getAnswersContent | public | | Stirng | |  | | Возвращает строку, содержащую выражение ответа |
| getAnswerText | public | | String | |  | | Возвращает строку содержащую текст ответа |
| getProblemContent | public | | String | |  | | Возвращает строку с выражение задачи |
| getProblemText | public | | String | |  | | Возвращает строку с текстом задачи |

Описание класса JordanCanonical Таблица 2.8

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| size | | private | | int | | Размер квадратной матрицы | |
| jordan | | private | | Matrix | | Жорданова матрица | |
| problemMatrix | | private | | Matrix | | Матрица задачи | |
| **Методы** | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| generateJordan | private | | Matrix | |  | | Генерирует Жорданову матрицу |
| getAnswersContent | public | | Stirng | |  | | Возвращает строку, содержащую выражение ответа |
| getAnswerText | public | | String | |  | | Возвращает строку содержащую текст ответа |
| getProblemContent | public | | String | |  | | Возвращает строку с выражение задачи |
| getProblemText | public | | String | |  | | Возвращает строку с текстом задачи |

Описание класса MatrixAddSubMult Таблица 2.9

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| sign | | private | | String | | Знак операции задачи | |
| **Методы** | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| generateSign | private | | String | |  | | Создаёт знак операции для задачи |
| getAnswersContent | public | | Stirng | |  | | Возвращает строку, содержащую выражение ответа |
| getAnswerText | public | | String | |  | | Возвращает строку содержащую текст ответа |
| getProblemContent | public | | String | |  | | Возвращает строку с выражение задачи |
| getProblemText | public | | String | |  | | Возвращает строку с текстом задачи |

Описание класса MatrixPowerN Таблица 2.10

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| P | | private | | Matrix | | Невырожденная матрица | |
| invP | | private | | Matrix | | Матрица обратная матрице Р | |
| Q | | private | | Matrix | | Диагональная матрица | |
| Qn | | private | | Matrix | | Диагональная матрица степени n | |
| A | | private | | Matrix | | Матрица для задачи | |
| width | | private | | int | | Размер матрицы | |
| **Методы** | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| createMatrixA | private | | void | |  | | Создаёт матрицу А |
| createMatrixQ | private | | void | |  | | Создаёт матрицу Q |
| createMatrixQn | private | | void | |  | | Создаёт матрицу Qn |
| getAnswersContent | public | | Stirng | |  | | Возвращает строку, содержащую выражение ответа |
| getAnswerText | public | | String | |  | | Возвращает строку содержащую текст ответа |
| getProblemContent | public | | String | |  | | Возвращает строку с выражение задачи |
| getProblemText | public | | String | |  | | Возвращает строку с текстом задачи |

Описание класса MatrixProblem Таблица 2.11

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| firstTerm | | protected | | Matrix | | Первый член выражения | |
| secondTerm | | protected | | Matrix | | Второй член выражения | |
| answer | | protected | | Matrix | | Ответ | |
| rand | | protected | | Matrix | | Генератор псевдослучайных чисел | |
| **Методы** | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| getMatrixValues | protected | | String | | Matrix | | Получение значений матрицы в формате Latex |
| texExpression | protected | | String | | String | | Получение выражения в формате Latex |

Описание класса MatrixProblemFactory Таблица 2.12

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| getRandomProblem | public | MatrixProblem | Random, int | Получение задачи случайного типа |
| typeOf | public | MatrixProblem | String, int | Получение задачи определённого типа |

Описание класса MatrixRank Таблица 2.14

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| matrix | | private | | Matrix | | Матрица для задачи | |
| initialMatrix | | private | | String | | Изначальная матрица до перетасовки | |
| rank | | private | | int | | Ранг матрицы | |
| **Методы** | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| getAnswersContent | public | | Stirng | |  | | Возвращает строку, содержащую выражение ответа |
| getAnswerText | public | | String | |  | | Возвращает строку содержащую текст ответа |
| getProblemContent | public | | String | |  | | Возвращает строку с выражение задачи |
| getProblemText | public | | String | |  | | Возвращает строку с текстом задачи |

Описание класса OrthogonalDiag Таблица 2.15

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| problemText | | private | | String | | Текст задачи | |
| answerText | | private | | String | | Текс ответа | |
| T | | private | | Matrix | | Ортогональная матрица | |
| invT | | private | | Matrix | | Обратная матрица матрице Т | |
| D | | private | | Matrix | | Диагональная матрица с собственными значениями по диагонали | |
| A | | private | | Matrix | | Матрица для задачи | |
| n | | private | | int | | Знаменатель ортогональной матрицы | |
| **Методы** | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| createA | private | | Matrix | |  | | Создаёт матрицу задачи |
| createD | private | | Matrix | |  | | Создаёт матрицу D |
| createT | private | | Matrix | |  | | Создаёт матрицу Т |
| getAnswersContent | public | | Stirng | |  | | Возвращает строку, содержащую выражение ответа |
| getAnswerText | public | | String | |  | | Возвращает строку содержащую текст ответа |
| getProblemContent | public | | String | |  | | Возвращает строку с выражение задачи |
| getProblemText | public | | String | |  | | Возвращает строку с текстом задачи |

Описание класса Qrdecomposition Таблица 2.16

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| Q | | private | | Matrix | | Матрица Q для QR разложения | |
| R | | private | | Matrix | | Матрица R для QR разложения | |
| A | | private | | Matrix | | Матрица задачи | |
| n | | private | | int | | Знаменатель ортогональной матрицы | |
| **Методы** | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| createQ | private | | Matrix | |  | | Создаёт ортогональную матрицу Q |
| createR | private | | Matrix | |  | | Создаёт верхнетреугольную матрицу R |
| getAnswersContent | public | | Stirng | |  | | Возвращает строку, содержащую выражение ответа |
| getAnswerText | public | | String | |  | | Возвращает строку содержащую текст ответа |
| getProblemContent | public | | String | |  | | Возвращает строку с выражение задачи |
| getProblemText | public | | String | |  | | Возвращает строку с текстом задачи |

Описание класса RotationAroundAxis Таблица 2.17

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| problemText | | private | | String | | Текст задачи | |
| possibleFractions | | private | | ArrayList<Iexpr> | | Возможные значения дроби для аргумента угла | |
| arg | | private | | IExpr | | Аргумент или угол | |
| A | | private | | Matrix | | Матрица для задачи | |
| Frac | | private | | FractionSym | | Дробь аргумента | |
| axis | | private | | ArrayList<IExpr> | | Ось вращения | |
| **Методы** | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| initA | private | | Matrix | |  | | Инициализирует матрицу задачи |
| initArg | private | | IExpr | |  | | Инициализирует аргумент |
| initAxis | private | | ArrayList<Iexpr> | |  | | Инициализирует ось вращения |
| getAnswersContent | public | | Stirng | |  | | Возвращает строку, содержащую выражение ответа |
| getAnswerText | public | | String | |  | | Возвращает строку содержащую текст ответа |
| getProblemContent | public | | String | |  | | Возвращает строку с выражение задачи |
| getProblemText | public | | String | |  | | Возвращает строку с текстом задачи |

Описание класса SVDdecomposition Таблица 2.18

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| U | | private | | Matrix | | Ортогональная матрица | |
| A | | private | | Matrix | | Матрица для задачи | |
| V | | private | | Matrix | | Ортогональная матрица | |
| D | | private | | Matrix | | Диагональная матрица с собственными значениями | |
| nU | | private | | int | | Делитель ортогональной матрицы U | |
| nV | | private | | int | | Делитель ортогональной матрицы V | |
| problemText | | private | | String | | Строка с текстом задачи | |
| **Методы** | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| createD | private | | Matrix | |  | | Создает матрицу D |
| createU | private | | Matrix | |  | | Создает матрицу U |
| createV | private | | Matrix | |  | | Создает матрицу V |

Описание класса TasksDocument Таблица 2.19

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| text | | private | | ArrayList<String> | | Список с текстами задач | |
| content | | private | | ArrayList<String> | | Список с содержимом задач | |
| answers | | private | | ArrayList<String> | | Список с ответами | |
| **Методы** | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| addTask | public | | void | | MatrixProblem | | Разбивает задачу и добавляет части по спискам |
| createTaskTex | public | | void | | String, String | | Создаёт строку задачи в формате Latex |
| createTex | public | | String | | String | | Создаёт строку задачи в формате Latex по обозначенному пути |
| documentSolutions | private | | String | |  | | Создаёт документ с решениями по адресу |
| documentTaks | private | | String | |  | | Создаёт документ с задачами по адресу |
| getSize | public | | int | |  | | Получает размер документа с задачами |

Описание класса AlgebraGeneratorApplication Таблица 2.20

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| main | public | void | String[] | Входная точка приложения |

Описание класса ExprUtils Таблица 2.21

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| getPositiveRandom | public | IExpr | Random, int, int | Получает случайное положительное выражение в пределах |
| getRandom | public | IExpr | Random, int, int | Получает случайное значение в пределах |
| getRandomNonNull | public | IExpr | Random, int, int | Получает случайное ненулевое значение в пределах |

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Изм. | Номера листов (страниц) | | | | Всего листов (страниц) в документе | № документа | Входящий № сопроводительного документа и дата | Подпись | Дата |
| изменённых | заменённых | новых | аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |