Содержание

[Введение 3](#_Toc533207633)

[1. Анализ объекта 4](#_Toc533207634)

[1.1 Описание предметной области. 4](#_Toc533207635)

[1.2 Построение концептуальной модели предметной области. 6](#_Toc533207636)

[2 Постановка задачи 7](#_Toc533207637)

[2.1 Определение требований к программной системе. 7](#_Toc533207638)

[2.2 Описание аналогов системы. 8](#_Toc533207639)

[2.3 Обзор и обоснование выбора средств реализации 9](#_Toc533207640)

[3 Проектирование 10](#_Toc533207641)

[3.1 Разработка архитектуры программного продукта 10](#_Toc533207642)

[3.2 Проектирование структур хранения данных. 10](#_Toc533207643)

[4 Реализация 11](#_Toc533207644)

[4.1 Разработка архитектуры программного продукта 11](#_Toc533207645)

[4.2 Разработка интерфейса программного продукта 13](#_Toc533207646)

[4.3 Разработка алгоритмов реализации вариантов использования. 15](#_Toc533207647)

[5 Тестирование 18](#_Toc533207648)

[Литература 20](#_Toc533207649)

[Приложение А 21](#_Toc533207650)

# Введение

Транспортная задача - это математическая задача линейного программирования специального вида о поиске оптимального распределения однородных объектов с минимизацией затрат на перемещение. Существует несколько методов решения транспортной задачи:

1. решение транспортной задачи методом потенциалов
2. решение транспортной задачи с использованием симплекс метода.

Мой проект представляет собой программу для решение транспортных задач методом потенциалом с нахождением опорного плана методом минимального элемента. Данная курсовая работа выполнена на языке программирования высокого уровня Java с использованием IDE Eclipse.

# Анализ объекта

## Описание предметной области.

Симплекс метод - [метод линейного программирования](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html), который реализует рациональный перебор базисных допустимых решений, в виде конечного итеративного процесса, необходимо улучшающего значение целевой функции на каждом шаге.

Применение симплекс-метода для [задачи линейного программирования](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp01.html) предполагает предварительное приведение ее формальной постановки к [канонической форме](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp01.html#$lp01.1) с *n* неотрицательными переменными: *(X1, ..., Xn)*, где требуется минимизация линейной целевой функции при *m* линейных ограничениях типа равенств. Среди переменных задачи выбирается начальный базис из *m* переменных, для определенности *(X1, ..., Xm)*, которые должны иметь неотрицательные значения, когда остальные *(n-m)* [свободные переменные](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html#$lp03.1) равны 0. Целевая функция и ограничения равенства преобразуются к диагональной форме относительно [базисных переменных](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html#$lp03.1), переменных, где каждая [базисная переменная](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html#$lp03.1) входит только в одно уравнение с коэффициентом 1.

Данная формальная модель задачи линейного программирования обычно задается в форме, так называемой *симплекс-таблицы*, удобной для выполнения операций симплекс-метода:

**Симплекс-таблица**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | X1 | X2 | ... | Xm | Xm+1 | ... | Xn |
| X0 | A0,0 | 0 | 0 | ... | 0 | A0,m+1 | ... | A0,n |
| X1 | A1,0 | 1 | 0 | ... | 0 | A1,m+1 | ... | A1,n |
| X2 | A2,0 | 0 | 1 | ... | 0 | A2,m+1 | ... | A2,n |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| Xm | Am,0 | 0 | 0 | ... | 1 | Am,m+1 | ... | Am,n |

Верхняя строка симплекс-таблицы представляет целевую функцию задачи. Каждая строка симплекс-таблицы, кроме первой, соответствует определенному ограничению-равенству задачи. Свободные члены ограничений составляют крайний левый столбец таблицы. Слева от таблицы записаны текущие [базисные переменные](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html#$lp03.1) *(X1, ..., Xm)*. Сверху от таблицы приведен набор всех переменных задачи, где *Xm+1, ..., Xn* - [свободные переменные задачи](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html#$lp03.1).

На начальном шаге алгоритма симплекс-метода должно быть выбрано [базисное допустимое решение](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html#$lp03.1) *(X1, ..., Xm) >= 0* при *Xj = 0 (j = m+1, ..., n)*, следовательно, все свободные члены ограничений *Ai,0 >= 0 (i = 1, ..., m)*. Когда это условие выполнено, симплекс-таблица называется *прямо-допустимой*, так как в этом случае [базисные переменные](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html#$lp03.1), равные *Ai,0*, определяют допустимое решение прямой задачи линейного программирования. Если все коэффициенты целевой функции *A0,j >= 0 (j = 1, ..., m)*, то симплекс-таблица называется *двойственно-допустимой*, поскольку соответствующее решение является допустимым для [двойственной задачи линейного программирования](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp02.html).

Если симплекс-таблица является одновременно прямо и двойственно допустимой, т.е. одновременно все *Ai,0 >= 0* и *A0,j >= 0*, то решение оптимально.

Действительно, поскольку допустимыми являются лишь неотрицательные значения управляемых параметров, то изменение целевой функции за счет вариации [свободных переменных](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html#$lp03.1), через которые она выражена, возможно только в сторону увеличения, т.e. будет ухудшаться. Если среди ее коэффициентов имеются *A0,j < 0*, то значение целевой функции еще можно уменьшить (т.e. улучшить), увеличивая значение любой [свободной переменной](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html#$lp03.1) *Xj* с отрицательным коэффициентом *A0,j* при побочном уменьшении [базисных переменных](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html#$lp03.1), чтобы оставались справедливы ограничения задачи. Теоретически можно использовать любую [свободную переменную](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html#$lp03.1) *Xj* с *A0,j < 0*, но на практике обычно действуют в соответствии со стратегией наискорейшего спуска, выбирая минимальный элемент *A0,p < 0* из всех отрицательных *A0,j <&nbsp0*:

A0,p = **min** A0,j < 0.

j

Столбец *p* [симплекс-таблицы](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp04.html#$lp04.1), соответствующий выбранному коэффициенту *A0,p < 0*, называется *ведущим столбцом*. [Свободная переменная](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html#$lp03.1) ведущего столбца должна быть введена в базис вместо одной из текущих [базисных переменных](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html#$lp03.1). Очевидно, из базиса следует исключить такую переменную *Xq*, которая раньше других обращается в нуль при увеличении переменной *Xp* ведущего столбца.

Её индекс легко определить, если среди положительных элементов ведущего столбца *p* найти элемент, минимизирующий отношение *(Ai,0 / Ai,p)*:

Aq,0 Ai,0

------ = **min** ------ , i = 1,...,m.

Aq,p i Ai,p

Элемент *Aq,p* называется *ведущим элементом*, cтрока *q* [симплекс-таблицы](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp04.html#$lp04.1), содержащая ведущий элемент, называется, соответственно, *ведущей строкой*. Переменная ведущей строки *Xq* заменяется в базисе переменной ведущего столбца *Xp* и становится [свободной переменной](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html#$lp03.1) с значением 0, в то время как новая [базисная переменная](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html#$lp03.1) *Xp* достигнет максимально возможного значения, равного: **max** Xp = ( Aq,0 / Aq,p).

После указанного взаимообразного обмена переменными *Xp* и *Xq* между наборами [свободных и базисных переменных](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html#$lp03.1) нужно модифицировать исходную [каноническую модель](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp01.html#$lp01.1) задачи путем приведения ее к диагональной форме относительно нового множества [базисных переменных](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html#$lp03.1). Для указанного преобразования можно формально использовать процедуру *исключения Гаусса*, которая, как известно, состоит из двух элементарных операций, применяемых к системе алгебраических уравнений ( в данном случае ограничений - равенств):

* умножение уравнения *E1(X) = 0* на константу *K1* и замена уравнения *E1(X) = 0* уравнением *K1\*E1(X) = 0;*
* сложение уравнений *E1(X) = 0* и *E2(X) = 0* c последующей заменой уравнения *E2(X) = 0* уравнением *E1(X) + E2(X) = 0*.

Исключения Гаусса позволяют привести систему уравнений к диагональной форме относительно желаемого множества переменных. В данном случае исключение Гаусса применяется так, чтобы все элементы [симплекс-таблицы](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp04.html#$lp04.1) в ведущем столбце, кроме ведущего элемента *Aq,p*, стали нулевыми, а ведущий элемент стал равным единице:

Ai,p = 0, если i не равно q и Ai,p = 1, если i = q.

Указанные шаги симплекс-метода повторяются, пока не будет получена [симплекс-таблица](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp04.html#$lp04.1), которая одновременно является прямо и двойственно допустимой. Если положит в такой [симплекс-таблице](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp04.html#$lp04.1) текущие [базисные переменные](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html#$lp03.1) равными *Ai,0*, а свободные - нулю, то будет получено оптимальное решение.

Практика применения симплекс метода показала, что число итераций, требуемых для решения [задачи линейного программирования](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp01.html) обычно колеблется от *2m* до *3m*, хотя для некоторых специально построенных задач вычисления по правилам симплекс метода превращаются в прямой перебор [базисных допустимых решений](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp01.html). Однако, трудные для симплекс метода задачи на практике встречаются крайне редко, что объясняет широкое распространение и большую популярность данного [метода линейного программирования](http://www.kgtu.runnet.ru/WD/TUTOR/lp/lp03.html#$lp03.2) по сравнению с другими подходами.

## Построение концептуальной модели предметной области.

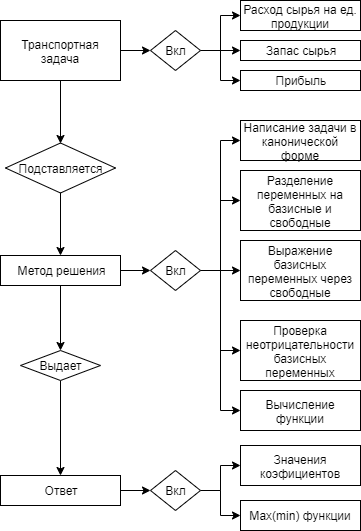


Рисунок 1.6 – модель предметной области

Концептуальная модель состоит транспортной задачи, данные которой пользователю необходимо ввести (Расход сырья, запас, прибыль(вводить функцию)). После ввода пользователь нажимает кнопку «Решить», и данные задачи подставляются в метод решения. Задача записывается в канонической форме, переменные разделяются на базисные и свободные. Затем выражаем базисные переменные через свободные и проверяем их неотрицательность. В конце вычисляем значения переменных и функции. Как только задача решена, метод выдает пользователю ответ.

# Постановка задачи

## Определение требований к программной системе.

1. Приложение, позволяющее решить уравнение симплекс-методом
2. Поддержка ввода данных как с экрана, так и с файла
3. Поддержка вывода данных как на экран, так и в файл
4. Дружелюбный интерфейс
5. Поддержка Java версии 9 и более

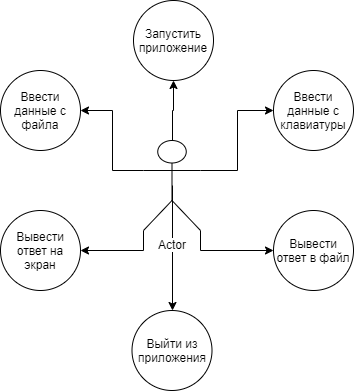


Рисунок 2.1 - диаграмма вариантов использования

## Описание аналогов системы.

В сети Интернет можно найти большое количество работ, которые реализуют решение данных задач. Практически все они представляют собой сайты, так что для решения уравнений необходимо подключение к интернету. Примером таким сайтов являются «math.semestr.ru» и «matworld.ru».

Мой курсовой проект представляет собой приложение для ПК. Программа не требует подключения к интернету и имеет функции загрузки и выгрузки данных в файл.

### 2.3 Обзор и обоснование выбора средств реализации

Eclipse —интегрированная среда разработки ПО. Написана на языке Java в 2003 году. Также является кроссплатформенной. За счёт присоединяемых к этой среде дополнений — имеется возможность создавать программные продукты более чем на пяти языках программного кода.

Достоинства: Постоянное обновление версий среды разработки, поддержка многих языков (в том числе и русского), является бесплатной, поддержка многих языков программирования, среда имеет промышленный уровень, является гибкой — то есть легко настраивается как под любую платформу, так и под любого пользователя.

Для создания интерфейса было применено:

JFrame – класс, реализующий отображение окон приложения.

JLabel – элементы отображения надписей

JButton – элементы отображения кнопок

JTextField – элементы отображение полей ввода

JTable – элемент отображения таблиц

# Проектирование

## Разработка архитектуры программного продукта

Поскольку приложение не сетевое, для лучшей скорости сделано однопоточным.

Главное окно будет содержать следующие элементы:

* Поле «Количество строк» - количество ограничений
* Поле «Количество столбцов» - количество переменных (х1, х2…)
* Таблицу, созданную по заданным размерам для хранения строк ограничений
* Кнопку «Создать таблицу»
* Кнопку «help» - подсказка к заполнению строк таблицы
* Кнопку «Открыть файл» - для ввода данных из файла
* Кнопку «Решить» - для решение задачи и выдачи ответа
* Кнопку «Сохранить» - для сохранения ответа в файл

## Проектирование структур хранения данных.

Для хранения данных извне будет использован текстовый файл, данные в нем будут храниться построчно. Для хранения таблицы используется переменная table.

# Реализация

## Разработка архитектуры программного продукта

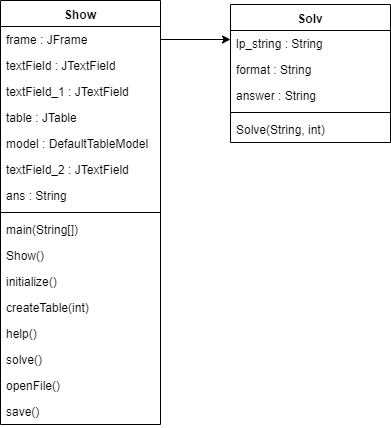


Рисунок 4.1 – Диаграмма классов

Главным классом является класс Show, у которого есть переменные для ввода данных задачи.

При нажатии кнопки «Создать таблицу» создается таблица с выбранным количеством строк ограничений. В таблицу построчно записываются ограничения. В поле «Функция» пишется функция прибыли. Данные таблицы передаются в класс “Solv”. При помощи стандартных библиотек и магии программирования задача решается и нам возвращается ответ. Если пользователь ввел не все данные, некорректные или же не ввел их, ему выдастся соответствующее уведомление.

## Разработка интерфейса программного продукта

Главное окно

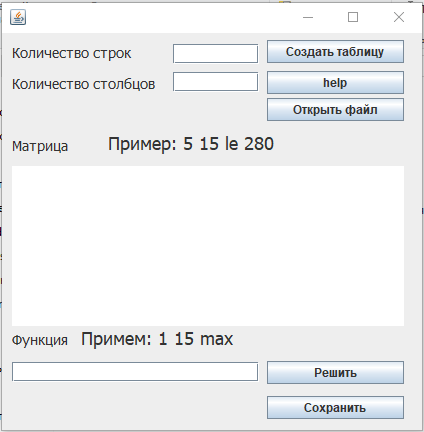


Рисунок 4.2.1 – Вид главного окна

Введем данные и создадим таблицу

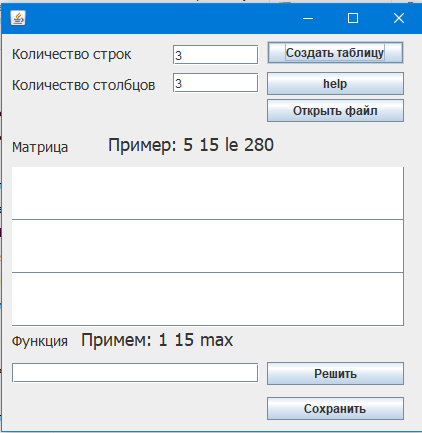


Рисунок 4.2.2 – Создание таблицы

Попробуем загрузить данные из файла

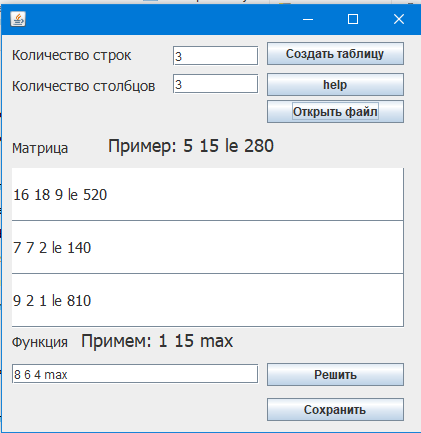


Рисунок 4.2.3 – Заполнение данных из файла

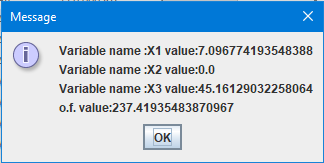


Рисунок 4.2.4 – Окно ответа.

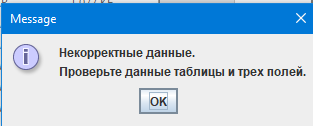


Рисунок 4.2.5– Попытка решить без введенных данных

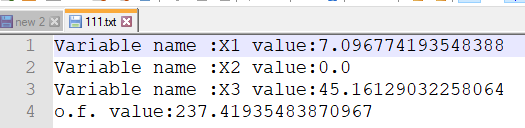


Рисунок 4.2.6 – Вид сохраненного файла

## Разработка алгоритмов реализации вариантов использования.

Общий алгоритм работы приложения:

1. Открывается главное меню, переход к 2 или 9
2. Вводит количество ограничений, если еще не введена, переход к 3
3. Вводит количество неизвестных, если еще не введен, переход к 4
4. Кнопка создать таблицу создает таблицу, переход к 5
5. Ввод матрицы ограничений, если еще не введена, переход к 6
6. Если нажата кнопка из фала, вывод окна выбора файла, заполнение матрицы, переход к 7
7. Если нажата кнопка решение, решается задача и переход к 8
8. Открывается окно ответа, переход к 2
9. Завершение работы приложения
10. Создание таблицы

**void** createTable(**int** row){

**try** {

table.removeAll();

table.setModel(model);

model.setRowCount(row);

model.setColumnCount(1);

table.setAutoResizeMode(JTable.***AUTO\_RESIZE\_ALL\_COLUMNS***);

table.setRowHeight(160 / row);

}**catch**(Exception ex) {

JOptionPane.*showMessageDialog*(**null**, "Ошибка при создании таблицы. \nПроверьте правильность ввода");

}

}

1. Передача данных в метод решения

**public** **static** String Solve(String str, **int** col) **throws** Exception {

InputString lp\_input = **new** InputString(str);

**for**(**int** i=1; i<=col;i++) {

*format*+="X"+i+":double, ";

}

lp\_input.setInputFormat(*format* + "TYPE:varstring(5), RHS:double");

LP lp = **new** LP(lp\_input);

SolutionType solution\_type = lp.resolve();

**if** (solution\_type == SolutionType.***OPTIMUM***) {

Solution solution = lp.getSolution();

**for** (Variable var : solution.getVariables()) {

SscLogger.*log*("Variable name :" + var.getName() + " value:" + var.getValue());

*answer*+="Variable name :" + var.getName() + " value:" + var.getValue()+"\n";

}

SscLogger.*log*("o.f. value:" + solution.getOptimumValue());

*answer*+="o.f. value:" + solution.getOptimumValue();

} **else** {

SscLogger.*log*("no optimal solution:" + solution\_type);

*answer*+="no optimal solution:" + solution\_type;

}

**return** *answer*;

}

1. Чтение данных из файла

**void** openFile() {

JFileChooser fileopen = **new** JFileChooser();

**int** ret = fileopen.showDialog(**null**, "Открыть файл");

**if** (ret == JFileChooser.***APPROVE\_OPTION***) {

File file = fileopen.getSelectedFile();

Scanner sc = **null**;

**try** {

sc = **new** Scanner(file);

//число ограничений, число столбцов

//функция

// матрица

**int** row = sc.nextInt();

**int** col=sc.nextInt();

sc.nextLine();

createTable(row);

textField.setText(row+"");

textField\_1.setText(col+"");

textField\_2.setText(sc.nextLine());

**for**(**int** i=0; i<row;i++)table.setValueAt(sc.nextLine(), i, 0);

} **catch** (FileNotFoundException e) {

JOptionPane.*showMessageDialog*(**null**, "Файл не найден или расширенеи не поддерживается.");

e.printStackTrace();

}

}

}

# 5. Тестирование

Проведем тестирование программы, пройдя все варианты взаимодействия. Запустим само приложение. Запуск происходит без проблем.

Попробуем найти решение не вводя значениЙ.

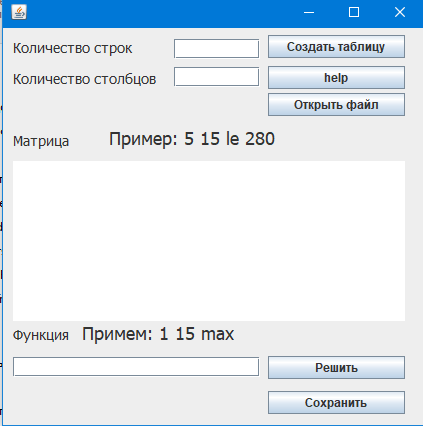


Рисунок 5.1 – Запуск главного окна

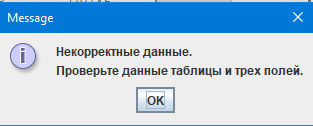


Рисунок 5.2 – Попытка запуска игры без введенных значений

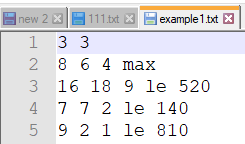
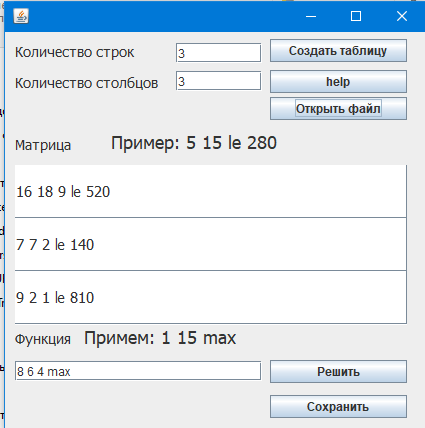
 

Рисунок 5.2 – Загрузка данных из файла

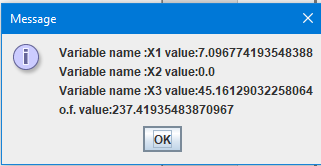


Рисунок 5.4 – Вывод ответа

Работа приложения происходит без ошибок

## 

## Литература

1. Хорстманн, К.С. Java 2. Библиотека профессионала, том1. Основы. 8-е издание, : Пер. с англ. / К. С. Хорстманн, Г. Корнелл - М.: ООО «Вильямс», 2012. - 816 с.
2. Дирк, Л. Самоучитель Java 7: Пер. с нем. / Л. Дирк, П. Мюллер. – СПб: БХВ-Петербург, 2013. – 464 с.: илл.
3. Шилдт, Г. Java. Полное руководство. 8-е издание, : Пер. с англ. / Г. Шилдт. – М.:ООО «Вильямс», 2012. – 1104 с.
4. Блинов, И.Н. Java. Промышленное программирование./ И.Н. Блинов, В.С. Романчик –Минск: «Четыре четверти», 2013. – 896 с.
5. Онлайн калькулятор [Электронный ресурс]. – Режим доступа: math.semestr.ru. – Дата доступа: 20.11.2019.
6. Делюсь знаними [Электронный ресурс]. – Режим доступа: http://matecos.ru/. – Дата доступа: 20.11.2019.
7. Сайт преподователя экономики [Электронный ресурс]. – Режим доступа: http://galyautdinov.ru/. – Дата доступа: 20.11.2019.
8. Мир математики [Электронный ресурс]. – Режим доступа: https://matworld.ru/. – Дата доступа: 20.11.2019.

## Приложение А

Исходный код, откомпилированная программа и записка находятся на диске.