

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	7
1 Анализ проблемы и постановка задачи	9
1.1 Анализ предметной области	9
1.2 Содержательная постановка проблемы.....	9
1.3 Обзор существующего математического обеспечения для решения поставленной задачи	10
1.4 Формальная постановка задачи	14
1.5 Структура решения задачи, декомпозиция задачи на подзадачи	15
2 Анализ моделей и методов решения	18
2.1 Классификация задач оптимизации	18
2.2 Аналитический обзор известных математических моделей задач коммивояжера	19
2.3 Сравнительный анализ подходов, методов для решения задач коммивояжера	20
2.3.1 Точные алгоритмы	20
2.3.2 Неточные алгоритмы	25
3 Математическое, информационное и методическое обеспечение.....	29
3.1 Математическая модель задачи коммивояжера	29
3.2 Методы решения задачи коммивояжера генетическим алгоритмом	30
3.2.1 Выбор представления пространства решения (способ кодирования)	31
3.2.2 Функция оценки допустимости решения (целевая функция)	34
3.2.3 Выбор начального подмножества решений	34
3.2.4 Выбор операторов скрещивания	35
3.2.5 Выбор оператора мутации	39
3.2.6 Выбор оператора селекции	41

Инв. № подп	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата
-------------	--------------	--------------	--------------	--------------

1303.102002.001 ПЗ

Изм.	Лист	№ докум.	Подпись	Дата
Разраб.	Абзалтдинов Л.И.			
Провер.	Тарасова Т.Д.			
Реценз.	Тарасова Т.Д.			
Н. Контр.	Иванова Л.Ш.			
Утверд.	Юсупова Н.И.			

Математическое и программное обеспечение для построения и визуализации рациональных маршрутов с использованием генетических алгоритмов

Лит. Лист Листов
УГАТУ МО 405

3.3 Алгоритмы и структуры данных	43
4 Программное обеспечение	47
4.1 Язык программирования и инструментальные средства разработки ..	47
4.1.1 Выбор языка программирования.....	47
4.1.2 Выбор системы управления базой данных.....	48
4.1.3 Технология ORM	49
4.1.4 Стек используемых технологий	51
4.2 Описание структуры программного продукта	52
4.3 Описание интерфейса пользователя.....	52
5 Оценка качества решения	53
5.1 Тестирование ПО	53
5.1.1 Тестирование в нормальных условиях	53
5.1.2 Тестирование в экстремальных условиях	57
5.1.3 Тестирование в исключительных условиях	59
5.1.4 Анализ тестирования	59
5.2 Оценка качества программного обеспечения.....	59
5.3 Вычислительный эксперимент и анализ результатов	60
ЗАКЛЮЧЕНИЕ.....	66
СПИСОК ЛИТЕРАТУРЫ.....	67
ПРИЛОЖЕНИЕ А - ТЕХНИЧЕСКОЕ ЗАДАНИЕ	70
ПРИЛОЖЕНИЕ Б - РУКОВОДСТВО ПРОГРАММИСТА	78
ПРИЛОЖЕНИЕ В - РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ	92
ПРИЛОЖЕНИЕ Г - ОЦЕНКА КАЧЕСТВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	102
ПРИЛОЖЕНИЕ Д - РЕЗУЛЬТАТЫ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА (ВТОРОЙ ЭТАП)	115

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв. №	Подп. и дата
Изм.	Лист	№ докум.	Подпись	Дата

ВВЕДЕНИЕ

Большое количество задач, с которой человечество сталкивается ежедневно, можно свести к задаче построения рационального маршрута. Эти задачи могут быть как бытовыми, так и специфичными для определенной предметной области. Приведем некоторые примеры:

- построение кратчайшего маршрута между городами по существующей сети дорог;
- определение схемы перевозки нефти от точек нефтедобычи к нефтеперерабатывающим заводам с минимальной стоимостью транспортировки;
- проектирование газопровода, соединяющего буровые скважины морского базирования с находящейся на берегу приемной станцией. Целевая функция соответствующей модели должна минимизировать стоимость строительства газопровода; [1]
- прокладка оптоволоконных кабелей вычислительных сетей между мультиплексорами ввода/вывода с минимизацией длины используемых кабелей. [2]

Вышеприведенные задачи имеют общие черты: 1) легко формализуемы; 2) представимы в виде графа; 3) простота представления; 4) высокая вычислительная сложность; 5) один тип объектов (город, мультиплексор) и одно отношение («быть на расстоянии ... от ...»). Все эти свойства характеризуют классическую задачу коммивояжера (ЗК), а перечисленные задачи могут быть представлены в виде ЗК и решаться методами, предназначенными для их решения.

Задача коммивояжера — очень известная задача комбинаторной оптимизации, заключающаяся в нахождении рационального маршрута, проходящего через заданные местоположения ровно по одному разу с последующим возвратом в исходное местоположение. В условиях также указываются критерий выгодности маршрута (кратчайший, самый дешёвый,

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

7

совокупный критерий и т. п.) и соответствующие матрицы расстояний, стоимости и т. п.

Выпускная квалификационная работа посвящена разработке математического и программного обеспечения построения и визуализации рациональных автомобильных маршрутов с использованием генетических алгоритмов.

Цели выпускной квалификационной работы – снижение стоимости проезда по заданным объектам, а также исследование эффективности генетических алгоритмов для поиска рациональных маршрутов

В связи с поставленной целью, необходимо решить основные задачи:

1. Провести обзор существующих программных решений задачи построения рационального маршрута;
2. Провести обзор и анализ методов решения данной задачи;
3. Разработать алгоритмы поиска рационального маршрута;
4. Разработать программное обеспечение для построения и визуализации рационального маршрута;
5. Провести вычислительный эксперимент, проанализировать полученные результаты;
6. Сформировать выводы по итогам проведенной работы.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

8

1 Анализ проблемы и постановка задачи

1.1 Анализ предметной области

Наиболее распространено применение задачи построения рационального маршрута в транспортной логистике. В современном мире практически каждая компания предлагает сервис по доставке товаров, будь то доставка еды, мебели или строительных материалов, для того, чтобы успешно конкурировать на плотном рынке. Но для оптимизации расходов компании на доставку необходимо грамотно выстраивать маршруты. Поэтому маршруты должны быть рациональными, то есть в конечном счете снизить расходы компании на услуги доставки. Но так как затраты на проезд по маршруту напрямую зависят от количества потраченного в дороге топлива, а количество топлива – от длины построенного маршрута или суммарного времени проезда по нему, то для построения рационального маршрута следует минимизировать его длину и время.

Таким образом, рациональным маршрутом будем называть такой маршрут, что его длина или время проезда минимальны.

1.2 Содержательная постановка проблемы

Рассмотрим постановку задачи поиска рационального маршрута.

Дано:

- Географический адрес расположения пункта производства товара данного предприятия
- Перечень заявок на доставку товара с указанием адреса и даты доставки, а также описание заказа.

Требуется найти маршрут минимальной стоимости, начинающийся и заканчивающийся в пункте производства и проходящий по всем адресам заявок на доставку не менее одного раза, а также представить его:

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв.	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

9

- в графическом виде на карте одного из картографических сервисов;
- в форме текстовых подсказок для водителя вида «поворнуть ... через ... метров»

1.3 Обзор существующего математического обеспечения для решения поставленной задачи

В настоящее время довольно часто появляются новые программные средства для построения рационального маршрута. Наличие ПО – один из весомых стимулирующих факторов в поддержании роста фирмы и повышении ее рентабельности.

Анализ существующих автоматизированных информационных систем для построения рациональных маршрутов позволяет разделить имеющиеся программные продукты, присутствующие на российском рынке, на две основные группы.

Первая группа программ – это инструменты, плагины или библиотеки систем, решающих задачи маршрутизации. Данные продукты не имеют графического интерфейса и решают задачу коммивояжера для ограниченного числа адресов. Чаще всего данная группа программ применяется разработчиками полноценного ПО для построения рационального маршрута. В данную группу относятся:

- сервис Optimization API от компании Mapbox;
- функция Optimized Route в Directions API от компании MAPQUEST;
- сервис Route Optimization API от компании GraphHopper.

Сравнение характеристик данных сервисов приведены в таблице 1.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

10

Таблица 1 – Сравнительный анализ возможностей программных продуктов первой группы

Критерии	Mapbox	MAPQUEST	GraphHopper
Максимальное количество местоположений	12	25	До 150
Дополнительная функциональность	Возможность учитывать тип маршрута: пеший / велосипедный / автомобильный	Выбор типа оптимизации: самый быстрый или короткий маршрут	Возможность учитывать количество используемых транспортных средств
Данные, используемые для оптимизации	Время в пути	Дистанция / время в пути	Дистанция / время в пути
Формат выходных данных	JSON	JSON/XML	JSON
Цена	Бесплатно	Бесплатно	До 239\$ в месяц

Основной недостаток программ данной группы в том, что они выдают результат в текстовом виде определенного формата и не предоставляют средств визуализации построенного маршрута. Также нет бесплатных решений для решения задачи с большим количеством местоположений.

Вторая группа программ – это полноценные программные продукты для решения задач оптимизации в транспортной логистике, имеющие гибкие настройки и учитывающие множество критериев при построении рационального маршрута. Данные программы имеют интуитивно понятный графический интерфейс, импорт и экспорт данных из существующих БД и пр.

Характерными представителями данной группы являются:

- ПК «Логистика развоза»;
- Муравьиная логистика;
- ANTOR LogisticsMaster.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

По принципу построения, расчета и функциональным возможностям большинство программ похожи друг на друга. Для работы с подобными программами обычно требуется специалист, хорошо знакомый с задачей.

Основными блоками в каждом программном продукте является:

- Импорт данных о заявках на доставку / ввод данных;
- Выбор транспортного средства или водителя;
- Ввод дополнительных параметров;
- Расчет маршрута и вывод его на карте.

Рассмотрим подробнее представителей этой группы программ.

ПК «Логистика развоза»

Программное обеспечение разрабатывается в России. Программный комплекс «Логистика развоза» версия 3 – это современная система автоматизации транспортной логистики, разработанная в режиме «управляемое приложение» на базе платформы 1С:Предприятие 8.3, поэтому она оптимальна для применения предприятиям, использующим учетную программу 1С. Данный ПК предназначен для торговых и производственных компаний, осуществляющих развоз товаров по городу. Особенностью программы является то, что в ней нет ограничений на количество рассчитываемых автомобилей и заказов на развоз, пользователи программы могут самостоятельно создавать и настраивать различные схемы расчета доставки и использовать их для дальнейшей работы, есть возможность диспетчеризации водителей с использованием мобильных устройств с ОС Android.

Муравьиная логистика

Программное обеспечение разрабатывается в России и предлагает решение двух ключевых задач: расчет рациональных маршрутов и контроль выполнения маршрутов. Программа управляется через Web-интерфейс, поэтому не имеет ограничений на используемую ОС. Основным отличием от остальных программ этого сегмента является наличие собственного API, которое позволяет выполнить расчет рациональных маршрутов и сформировать результаты в любой учетной

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

12

системе. Данный ПК актуален для логистических, транспортно-экспедиторских, курьерских, почтовых компаний, незаменима для интернет-магазинов.

ANTOR LogisticsMaster

Программное обеспечение разрабатывается компанией ANTOR в России. Данная система является одним из компонентов комплекса программных средств «Управление транспортной логистикой». Система позволяет крупным и средним предприятиям, осуществляющим доставку товаров клиентам или транспортировку грузов на торговые точки и склады, автоматизировать процессы управления доставкой и планирования маршрутов, оптимально загружать весь парк транспортных средств, обеспечивать своевременную доставку продукции клиентам, эффективно контролировать работу водителей и экспедиторов. ANTOR LogisticsMaster™ автоматически строит и оптимизирует маршруты доставки на основе около 100 параметров, среди которых: данные о дорожной сети и статистике пробок; характеристики груза и автомобиля; время прибытия и длительность разгрузки; себестоимость выезда машины и др. Программа позволяет логисту не держать в голове десятки факторов и ограничений при планировании как доставки со склада, так и сбора грузов в процессе выполнения рейса.

Сводные данные о приведённых выше ПО представлены в таблице 2.

Таблица 2 – Сравнительный анализ возможностей программных продуктов второй группы

Критерии	ПК «Логистика развоза»	Муравьиная логистика	ANTOR LogisticsMaster
Скорость работы	Средняя	Медленная	Быстрая
Дополнительная функциональность	Учет грузоподъемности автомобиля	Учет грузоподъемности автомобиля	Учет грузоподъемности автомобиля, статистики пробок, планирование доставки с дозагрузкой

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

13

Окончание таблицы 2

Наличие приложения для водителя	Есть (за дополнительную плату)	Есть	Есть (отдельная система)
Цена	>= 50 000 руб	35\$ – 500\$ (зависит от количества маршрутов в день)	>= 200 000 рублей

Особенность данной группы программных продуктов в том, что они разработаны специально для снижения расходов на доставку компаний, которые будут пользоваться ею, и имеют гибкие настройки. Однако все они платны, и потому не будут выгодны мелкому бизнесу.

Анализ систем решения задач транспортной логистики показал, что существует много систем для решения задач данного вида, в которых для решения задачи построения рационального маршрута используются различные подходы. Но не существует бесплатного программного решения, которое бы находило рациональный маршрут для большого количества местоположений.

1.4 Формальная постановка задачи

Формальная постановка задачи исследования представлена на Рисунке 1:

Входные данные:

- Географический адрес расположения пункта производства товара данного предприятия
- Перечень заявок на доставку товара с указанием адреса и даты доставки, а также описание заказа.
- Тип оптимизации – по суммарному времени или по общей дистанции маршрута

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

14

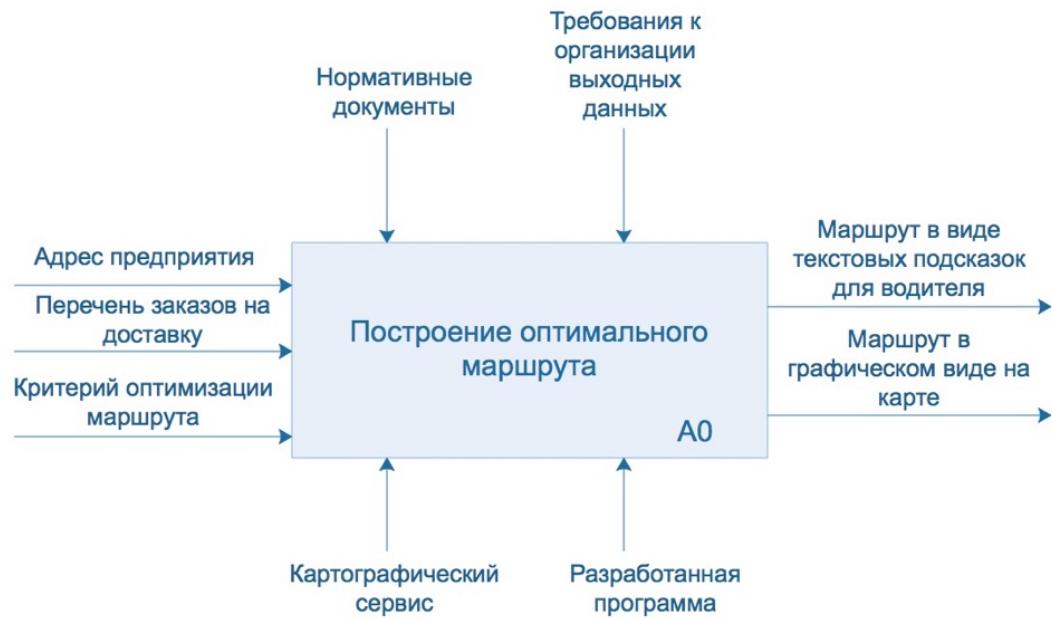


Рисунок 1 - Формальная постановка процесса построения рационального маршрута

Выходные данные:

- Рациональный маршрут в графическом виде на карте одного из картографических сервисов;
- Рациональный маршрут в виде текстовых подсказок для водителя вида «поворнуть ... через ... метров»

1.5 Структура решения задачи, декомпозиция задачи на подзадачи

Декомпозиция формальной постановки процесса построения рационального маршрута включает 3 блока в соответствии с рисунком 2.

1. Подготовка исходных данных:

На вход блоку подается:

- Географический адрес расположения пункта производства товара данного предприятия;

Инв. № подл.	Подл. и дата	Инв. № дубл.	Взам. № инв. №	Подл. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

15

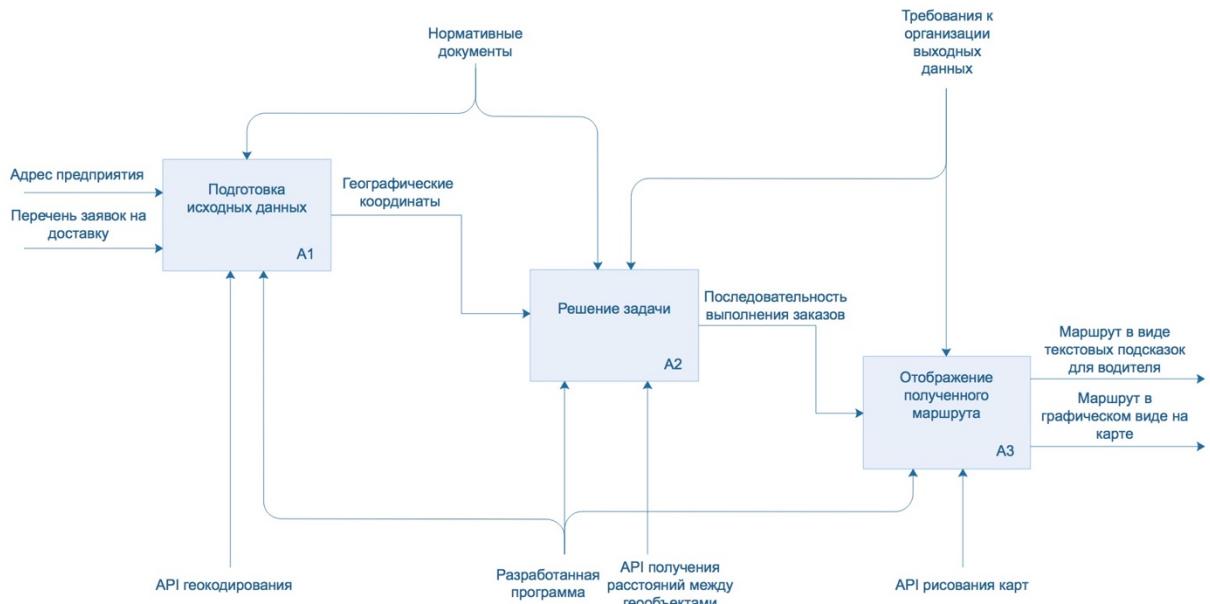


Рисунок 2 - Декомпозиция формальной постановки процесса построения
рационального маршрута

- Перечень заявок на доставку товара с указанием адреса и даты доставки, а также описание заказа.

На выходе формируются географические координаты адресов заявок и самого предприятия в виде пар чисел [широта; долгота]. Методический аспект подготовки таблиц данных регламентируется нормативной документацией, циркулирующей в рамках предприятия. Инструменты включают разработанную программу с использованием API геокодирования.

2. Решение задачи:

На вход блоку подается:

- Перечень географических координат;

На выходе производственные программы без учета вероятностной неопределенности и при различных уровнях вероятности. Выходная информация состоит из последовательности обхода заказов. В качестве инструмента используется разработанная программа с помощью API получения расстояний между геообъектами.

3. Отображение полученного маршрута:

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

16

На вход блоку подаются результаты решения задачи в качестве последовательности выполнения заказов.

На выходе формируются:

- Рациональный маршрут в графическом виде на карте одного из картографических сервисов;
- Рациональный маршрут в форме текстовых подсказок для водителя вида «поворнуть ... через ... метров».

Для отображения маршрута в графическом виде на карте используется API рисования карт.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

17

2 Анализ моделей и методов решения

2.1 Классификация задач оптимизации

По типу множества альтернатив задачи оптимизации разбиваются на две группы: если множество альтернатив конечно или счетно, то это задачи дискретной (комбинаторной) оптимизации; иначе – это задачи непрерывной (динамической) оптимизации. [3]

Задачами непрерывной оптимизации являются задачи создания и реализации системы рационального управления процессом при неустановившихся режимах эксплуатации. Особенностью задач динамической оптимизации является то, что значение критерия оптимальности определяется не только положением, существующим в рассматриваемый момент времени, но и предысторией процесса, начиная с некоторого начального момента. Поэтому оценка эффективности процесса должна учитывать его поведение в течение всего исследуемого нестационарного периода.

Задачами дискретной (комбинаторной) оптимизации же называются такие задачи, в которых требуется найти лучшую комбинацию значений переменных оптимизации, которые могут принимать только дискретные значения. Для многих таких задач решение может рассматриваться как размещение набора дискретных объектов в соответствии с заданными ограничениями. Во многих задачах комбинаторной оптимизации полный перебор нереален. Комбинаторная оптимизация включает в себя задачи оптимизации, в которых множество допустимых решений дискретно или может быть сведено к дискретному множеству. В исследованиях комбинаторной оптимизации были достигнуты значительные успехи. Однако для многих важных комбинаторных задач, встречающихся на практике и классифицированных как NP-полные, эффективные алгоритмы поиска оптимальных решений все еще неизвестны. NP-полными называются задачи, решение которых с вычислительными затратами,

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв.	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

18

описывающимися полиномиальной функцией характерного масштаба задачи, крайне маловероятно. [4]

К задачам комбинаторной оптимизации относятся следующие задачи:

- минимальное оствовное дерево;
- задача о ранце;
- задача раскюя;
- **задача коммивояжера;**
- задача о назначениях и др.

2.2 Аналитический обзор известных математических моделей задач коммивояжера

В рамках данной выпускной квалификационной работы далее будет рассматриваться только задача коммивояжера.

Существует несколько частных случаев общей постановки задачи коммивояжера (ЗК): [5]

- геометрическая ЗК: случай, когда матрица стоимостей между городами отражает расстояния между точками на плоскости
- метрическая ЗК: случай, когда на матрице стоимостей выполняется неравенство треугольника ($d_{AC} \leq d_{AB} + d_{BC}$), то есть прямой путь от города А к городу В никогда не дороже пути через город С;
- симметричная ЗК: в случае симметричной задачи все пары ребер между одними и теми же вершинами имеют одинаковую длину, то есть, для ребра (i,j) одинаковы длины $c_{ij} = c_{ji}$;
- асимметричная ЗК: случай, когда не выполняется условие симметричной ЗК, то есть не обязательно все пары ребер между одними и теми же вершинами имеют одинаковую длину.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв.	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

19

Модель асимметричной ЗК наиболее полно отражает реальную дорожную сеть, потому что не всегда прямой и обратный путь имеют одну и ту же стоимость, например, из-за наличия односторонних дорог или пробок на них.

2.3 Сравнительный анализ подходов, методов для решения задач коммивояжера

Алгоритмы для решения задачи коммивояжера можно разделить на точные (exact algorithm) и неточные (non-exact algorithm). Точные алгоритмы включают в себя перебор всех возможных вариантов, в частных случаях решения могут быть быстро найдены, но в целом осуществляется перебор $n!$ циклов. Вторые в общих случаях применяются для задач, которые невозможно решить точно (вычисление определенных интегралов, решение нелинейных уравнений, извлечение квадратного корня...), если существующие точные решения требуют значительных и неоправданных временных затрат при высокой сложности задачи, и как часть более сложного алгоритма, с помощью которого задача решается точно. [6]

2.3.1 Точные алгоритмы

В свою очередь существует две группы точных алгоритмов — одна из них использует методы релаксации линейного программирования TSP: алгоритм секущихся плоскостей, метод внутренней точки, метод ветвей и границ; вторая, меньшая группа, использует методы динамического программирования. Характерная особенность методов обеих групп — гарантия нахождения оптимальных решений при общей трудоемкости процесса. [7]

- Полный перебор (Brute Force)

Один из самых очевидных методов решения задачи коммивояжера — метод полного перебора или грубой силы.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

20

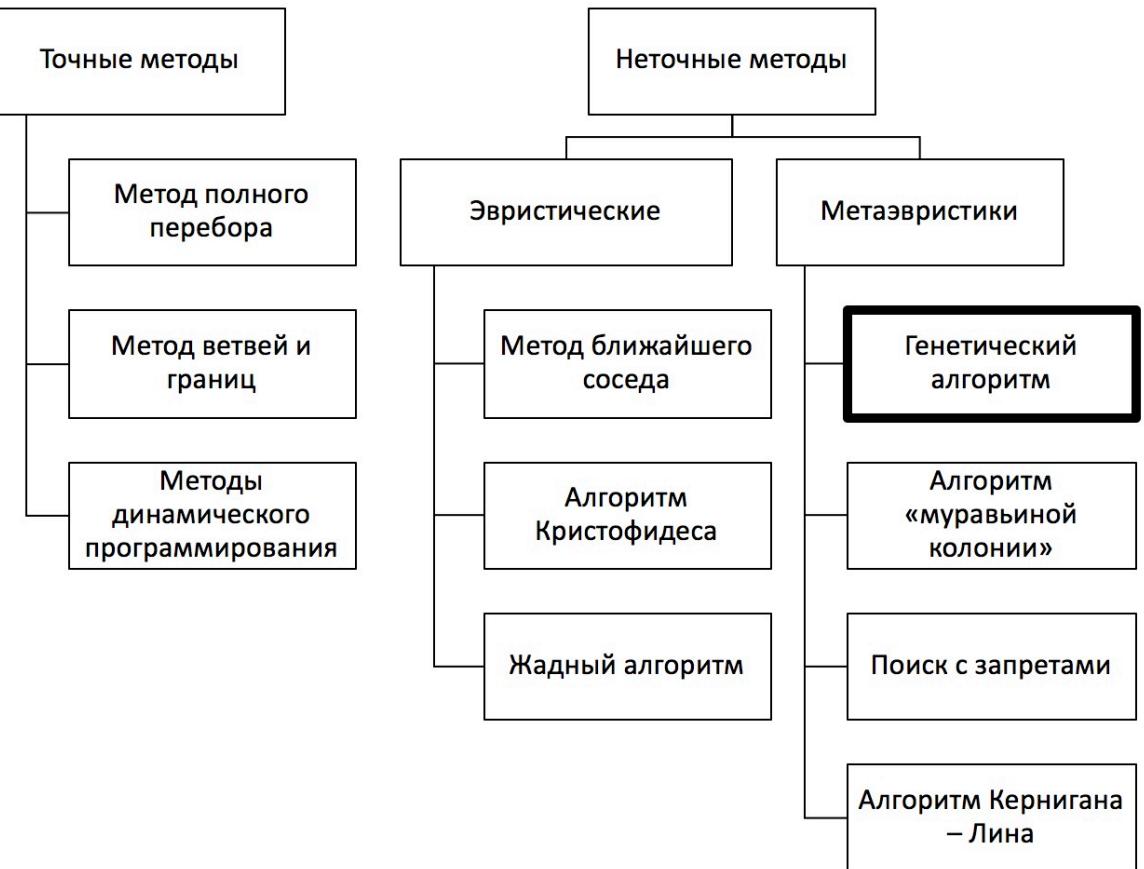


Рисунок 3 - Классификация алгоритмов решения задачи коммивояжера

Его суть заключается в переборе всех возможных вариантов путей, алгоритм решения можно записать как:

1. найти общее число возможных гамильтоновых контуров;
2. найти вес каждого гамильтонова контура, сложив вес всех его ребер;
3. выбрать гамильтонов контур с минимальным весом, который и будет оптимальным.

Метод полного перебора обладает рядом преимуществ – он гарантирует нахождение решения задачи коммивояжера, при этом он прямолинеен и прост в исполнении. В то же время, алгоритм считается неэффективным при работе с большим объемом данных, так как для нахождения рационального маршрута требует найти вес $(n - 1)!$ гамильтоновых контуров.

- Метод ветвей и границ (Branch and Bound)

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв.	Подп. и дата

Метод ветвей и границ часто используется для нахождения рационального решения задач комбинаторной оптимизации. Его суть заключается в разбитии множества на подзадачи и исключении заведомо неоптимальных решений.

Пусть граф V содержит все города, Π – множество всех перестановок городов, покрывающее все возможные решения. Рассмотрим перестановку $\pi \in \Pi$, в которой каждому городу назначается преемник – i для π_i города. Таким образом, тур можно записать как $(1, \pi(1), \pi(\pi(1)), \dots, 1)$. Если число городов в туре равно n , тогда перестановку называют циклической. Задача о назначениях ставит перед собой цель найти циклические перестановки, а задача коммивояжера преследует ту же цель, но с ограничением, что у этих перестановок должна быть минимальная стоимость. Метод ветвей и границ в первую очередь находит решение задачи о назначениях, стоимость которой для n городов довольно большая и асимптотически равна $O(n^3)$. [8]

Если был найден полный тур, то полученное значение также является решением задачи коммивояжера. В противном случае проблема разделяется на несколько подобластей, каждая из которых исключает некоторые дуги подтура, таким образом исключая сам подтур. Метод, с помощью которого высчитывается, какую дугу следует удалить, называют правилом ветвления. Важное замечание – не должно существовать дублированных подзадач, их общее количество должно быть минимизировано.

Будем использовать критерий, гарантирующий независимость подзадач – рассматривается включенный набор дуги и выбирается минимальное число дуг, которые не принадлежат набору. Обозначим E как множество исключенных дуг и I как множество включенных. Разложим I . Выберем t дуги подобластей, x_1, x_2, \dots, x_n , которые не принадлежат I . Задача разделена на t потомков так, чтобы $y_{j^{th}}$ потомка были E_j исключенных дуг и I_j включенных дуг. Запишем критерий в виде формулы (1).

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

22

$$\begin{cases} E_j = E \cup \{x_j\} \\ I_j = I \cup \{x_1, x_2, \dots, x_{j-1}\}, k = 1, 2, \dots, j \end{cases} \quad (1)$$

Но x - исключенная дуга j подзадач и включенная дуга в $(j+1)$ области. Это значит тур, полученный решением $(j+1)_{st}$: задачи, может иметь x_j дугу, но тур, полученный решением $(j+1)_{st}$, не содержит эту дугу. Это гарантирует отсутствие дублирующихся маршрутов.

Количество возможных решений равно $(n-1)!/2$, для $n=50$ это приблизительно 3×10^{62} . Этот метод наиболее часто используется при количестве узлов от 40 до 60. [9]

Необходимость целиком решать задачи линейного программирования во всей области допустимых решений можно считать главным недостатком вышеописанного метода. Для задач с большим объемом данных метод ветвей и границ является неоправданно трудоемким, в то же время алгоритм является надежным методом решения целочисленных задач.

- Алгоритм Гомори (The Cutting Plane)

В 1954 году была представлена работа Данцига, Фалкерсона и Джонсон, описывающая новый метод решения задачи коммивояжера, который также может быть использован для решения любой проблемы

$$\text{minimize } c^T x \text{ subject to } x \in S \quad (2)$$

где $c \neq 0$, S – конечное подмножество некоторого R^m , и таким образом мы сможем найти точки S . Это итерационный алгоритм – каждое повторение начинается с линейной программной релаксации. Запишем в виде формулы:

$$\text{minimize } c^T x \text{ subject to } Ax \leq b \quad (3)$$

где многогранник P , определенный как $\{x : Ax \leq b\}$, содержит S и ограничен. Так как P ограничен, мы можем найти оптимальное решение x^* как экстремальную точку P . Если x^* принадлежит S , то оптимальное решение найдено (2.2); в противном случае некоторое линейное неравенство удовлетворяет все точки

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв.	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

23

S и нарушает x^* . Такое неравенство называют алгоритмом Гомори, который подробно описал его 1958, методом отсекающих плоскостей или просто отсечений. [10]

Данный метод используется для построения точных или приближенных задач, особенно часто встречается в сочетании с методом ветвей и границ и тогда называется методом ветвей и отсечений. Оба метода основаны на решении последовательности релаксированных подзадач линейного программирования. В алгоритме Гомори релаксированные подзадачи постепенно улучшают аппроксимацию целочисленной задачи, уменьшая окрестность рационального решения. Если оптимальность не удалось получить, тогда ищется приближенное решение с погрешностью.

У метода отсечений есть преимущество над методом ветвей и границ – первые более удобны для аппаратного вычисления, так как для их решения не требуется большой объем оперативной памяти для хранения дерева решений. [11]

- Метод динамического программирования (Dynamic Programming)

Рассмотрим задачу с n городами и расстояниями между любыми двумя городами, путь начинается и заканчивается в городе n . TSP может быть решена за время $O(n!)$ методом полного перебора, но алгоритм динамического программирования позволяет сократить то время до $O(n^2 2^n)$.

Обозначим подзадачу: пусть S будет подмножеством городов, содержащим 1 и как минимум еще один город, j будет городом отличным от 1, обозначим через $C(S, j)$ самый короткий путь, начинающийся в 1, проходящий все города S и заканчивающийся в 1. Запишем алгоритм в виде псевдо-кода:

```

for all  $j$  do  $C(\{1, j\}, j) := d_{1,j}$ 
for  $s := 3$  to  $n$  do (the size of the subsets considered this round)
    for all subsets  $S$  of  $\{1, \dots, n\}$  of size  $n$  and containing 1 do
        for all  $j \in S, j \neq 1$  do
             $C(S; j) := \min_{i \neq j, i \in S} [C(S - \{j\}, i) + d_{i,j}]$ 

```

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

$$opt := \min [C(\{1,2,\dots,n\},j) + d_{1,j}] [12]$$

Данный метод используется для повышения эффективности вычислительных повторений, храня промежуточные результаты и снова используя их при необходимости.

2.3.2 Неточные алгоритмы

В целом алгоритмы данной группы предлагают потенциально неоптимальные, но быстрые решения. В свою очередь приближенные алгоритмы можно разделить на две категории: эвристические и метаэвристики.

- Алгоритм Кристофида (Christofides' Algorithm)

Алгоритм Кристофида используется для решения метрических ЗК.

Большая часть эвристических алгоритмов относятся к 2-приближенному классу. Профессор Никос Кристофайдс в 1976 году доработал один из существующих алгоритмов (метод двойного минимального оствового дерева $O(n^2 \log_2(n))$) так, что время решения задачи не превышает оптимальное время более чем на $3/2$. [13]

Решение оригинального алгоритма можно записать так: найти минимальное дерево из множества всех городов, продублировать все ребра и построить эйлеров граф, построить гамильтонов цикл, пройдя каждый узел только один раз и выбирая наикратчайших путь, ведущий из каждого узла.

Алгоритм Кристофида состоит из последовательности следующих действий:

1. найти минимальное дерево из множества всех городов;
2. найти паросочетание с минимальным весом множества вершин нечетной степени и построить эйлеров граф;
3. найти эйлеров обход и построить гамильтонов цикл, избегая посещаемых узлов. [13]

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв.	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

25

Основное различие – дополнительное вычисление паросочетания с минимальным весом. Эта часть также самая трудоемкая, поэтому время выполнения алгоритма возрастает до $O(n^3)$. [14]

- Алгоритм ближайшего соседа (NearestNeighbour)

Один из самых простых эвристических методов решения ЗК. Главное правило алгоритма – всегда выбирать близлежащий город (соседа). Решение задачи складывается из следующих шагов:

1. выбрать любой город;
2. найти близлежащий город, не включенный в маршрут, и перейти в него;
3. проверить остались ли города, не включенные в маршрут, если ответ положительный – повторить второй шаг;
4. чтобы завершить тур, добавить ребро между последним выбранным городом и первым.

В общем случае трудоемкость решения задачи равна $O(n^2)$. [15]

- Жадный алгоритм (Greedy)

Чтобы решить ЗК с использованием жадного алгоритма, мы исследуем все ребра, выходящие из города-узла, и выбираем n самых коротких дуг. Если те n самых коротких дуг формируют гамильтонов цикл, тогда мы нашли оптимальное решение. [16]

Трудоемкость решения задачи жадным алгоритмом равна $O(n^2)$. [14]

- Алгоритм Кернигана – Лина (Lin-Kernighan)

Алгоритм Кернигана – Лина считается один из самых эффективных методов поиска оптимальных или почти оптимальных решений задачи коммивояжера. Однако разработка и реализация алгоритма не проста, так как алгоритм состоит из множества шагов, большинство из которых сильно влияет на работу алгоритма. [17] Создание алгоритма Кернигана было вдохновлено наблюдением, что статическое K в -оптимальном методе не дает наилучшее решение. Появилась идея использовать различные стадии рационального метода в выполнении

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

26

эвристического алгоритма. На практике было показано, что практически невозможно заранее предугадать какое K следует использовать, чтобы достигнуть лучшего компромисса между трудоемкостью и качеством решения. Лин и Керниган убрали этот недостаток, введя оптимальную переменную, таким образом значение K меняется во время выполнения алгоритма. [18] Трудоемкость при этом равна $O(n^{2.2})$. [14]

- Алгоритм поиска с запретами (TabuSearch)

Главная проблема алгоритма ближайшего соседа состоит в частом застревании в точке локального оптимума. Этого можно избежать, применив алгоритм поиска с запретами, в 1977 году предложенный Ф. Гловером. Данный метод позволяет переходить от одного локального оптимума к другому в поиске глобального оптимума, после перехода ребро попадает в список запретов и повторно не используется, кроме случаев, когда оно может улучшить построенный путь. На практическом уровне запрещенный набор сохраняется как комбинация ранее посещаемых шагов, который позволяет построить дальнейший путь относительно текущего решения и соседних узлов. [19]

Главным недостатком этого метода служит его время выполнения – трудоемкость алгоритма оценивается как $O(n^3)$. [14]

- Муравьиный алгоритм (Ant Colony Optimization)

Муравьиный алгоритм – эффективный полиномиальных алгоритм, вдохновленный вдохновлена поведением настоящих муравьев. Впервые его принципы были описаны в 1991 Марко Дориго. Муравьям свойственно сотрудничать в поисках пищевые ресурсы, поэтому они оставляют след химического вещества, феромонов, на их пути от гнезда до источника пищи. [18] Этот тип неверbalной коммуникации называют стигмергия – стимуляция, основанная на опыте предыдущих муравьев и направленная на повышение производительности. [20]

Для решения задачи коммивояжера как правило используют около 20 муравьёв. Их размещают в случайные города и отправляют в другие города. Им не

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

27

позволяют дважды посещать один и тот же город, только если они не завершают маршрут. Тот муравей, который выбрал самый короткий тур, будет оставлять след феромонов обратно пропорциональный длине маршрута. Этот след феромонов будет считываться следующим муравьем при выборе города, и с большой вероятностью он пойдет тем же путем, еще сильнее укрепив след. Этот процесс будет многократно повторен пока не будет найден маршрут, достаточно короткий, чтобы быть оптимальным.

Среди недостатков алгоритма хочется выделить, что первое полученное решение может оказаться одним из худших в плане оптимизации, однако при повторном решении метод выдает достаточно точный результат. [21]

- Генетический алгоритм (Genetic Algorithm)

Генетические алгоритмы принадлежат к методам оптимизации, в основу которых легли биологические процессы, протекающие в природе. Чарльз Дарвин в своей эволюционной теории ввел определение естественного отбора, согласно которой особи, более приспособленные к условиям окружающей среды, имеют больше шансов на выживание и продолжение рода, и наоборот – неприспособленные особи подвергаются избирательному уничтожению. Основой отбора являются мутации генов и их комбинации, формирующиеся при размножении и передающиеся потомству. В ходе естественного отбора выживают экземпляры с наибольшей функцией приспособленности. Это численная характеристика, которая может изменяться в зависимости от условий конкретной задачи. Приспособленные особи скрещиваются (кроссовер) и производят потомство. Случайные мутации также могут оказывать влияние на развитие популяции.[22]

По состоянию на 2015 год алгоритм Керинагана-Лина считается самым эффективных методом для решения задачи коммивояжера на общих метрических пространствах, хотя известны лучшие приближения для частных случаев. [14]

В результате обзора и анализа существующих методов решения задачи коммивояжера для реализации было выбрано решение с помощью генетического алгоритма, так как данный алгоритм зарекомендовал себя с лучшей стороны.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

28

3 Математическое, информационное и методическое обеспечение

3.1 Математическая модель задачи коммивояжера

Прежде чем раскрывать математическую модель задачи коммивояжера, обусловимся описанные в содержательной постановке задачи адреса пункта производства товара и пунктов доставки называть городами в связи с общепринятой постановкой задачи коммивояжера.

Имеется N городов, по которым должен пройти коммивояжер с наименьшими затратами. При этом на его маршрут накладываются следующие ограничения:

- маршрут должен быть замкнутым, то есть коммивояжер должен вернуться в тот город, из которого он начал движение;
- в каждом из городов коммивояжер должен побывать не менее одного раза.

Для расчета затрат существует матрица условий, содержащая затраты на переход из каждого города в каждый, при этом считается, что можно перейти из любого города в любой, кроме того же самого (в матрице диагональ заполнена нулями). Целью решения является нахождения маршрута, удовлетворяющего всем условиям и при этом имеющего минимальную сумму затрат.

Коммивояжер должен обездить n городов. Для того чтобы сократить расходы, он хочет построить такой маршрут, чтобы обездить все города точно по одному разу и вернуться в исходный с минимумом затрат.

В терминах теории графов задачу можно сформулировать следующим образом.

Пусть $G(V,E)$ - полный ориентированный граф, где $V = \{1, \dots, N\}$ – множество вершин – городов, E - множество ребер (участков дорог). Каждому ребру $(i,j) \in E$ сопоставлено число $c_{i,j}$ - расстояние (или стоимость) между парой городов.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

29

Введем переменные $x_{i,j} = \begin{cases} 1, & \text{если из города } i \text{ движемся в город } j \\ 0, & \text{в противном случае} \end{cases}, i = \overline{1, N}, j = \overline{1, N}$.

Требуется найти замкнутый маршрут (цикл) виде матрицы $X = (x_{ij}), i = \overline{1, N}, j = \overline{1, N}$, удовлетворяющий следующим ограничениям: [23]

1°. Маршрут должен проходить через каждый город ровно один раз:

$$\sum_{i=1}^N x_{ij} = 1, j = \overline{1, N} \quad (4)$$

$$\sum_{i=1}^N x_{ij} = 1, i = \overline{1, N}, \quad (5)$$

$$x_{ij} = \begin{cases} 1, & \text{если из города } i \text{ едем в город } j \\ 0, & \text{в противном случае} \end{cases}, i = \overline{1, N}, j = \overline{1, N}$$

2°. Маршрут должен иметь минимальное суммарное пройденное расстояние или затраченное время:

$$L = \sum_{i=1}^N \sum_{j=1}^N c_{ij} x_{ij} \rightarrow \min \quad (6)$$

3.2 Методы решения задачи коммивояжера генетическим алгоритмом

Применение ГА для решения задачи коммивояжера можно представить следующей схемой: [24]

1. Предварительный этап:

а. выбор представления пространства решения (способ кодирования);

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист
30

- b. задание целевой функции – функции оценки допустимости решения;
 - c. выбор начального подмножества решений.
2. Этап конструирования решений с использованием операторов кроссовера.
 3. Этап случайного изменения решения с использованием оператором мутации.
 4. Этап отбора полученных решений в следующую выборку-поколение с использованием оператора селекции.

Конструирование конкретного типа ГА осуществляется выбором одного из возможных вариантов соответствующего оператора на каждом шаге выполнения ГА.

3.2.1 Выбор представления пространства решения (способ кодирования)

Для решения данной задачи возможны несколько способов кодирования генов:

1) Двоичное кодирование (binary representation)

При использовании данного способа кодирования ген (номер города) представляется в двоичной системе счисления, причем он задается минимальным количеством символов, которое определяется округлением в большую сторону следующей величины l :

$$l = \log_2 N, \quad (7)$$

где N – общее число городов в маршруте.

Пример кодирования маршрута данным способом отображен на рисунке ниже.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист
31

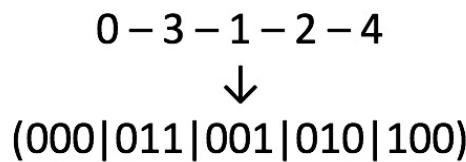


Рисунок 4 - Пример двоичного кодирования

Достоинства:

- универсальность;
- существует множество готовых алгоритмов.

Недостатки:

- при мутации и скрещивании хромосом возможно появление маршрутов, не удовлетворяющих ограничениям задачи (например, один город может появиться дважды);
- необходимо модифицировать существующие алгоритмы.[25]

2) Представление в виде пути обхода – последовательности номеров городов (Path representation)

При использовании данного способа кодирования ген представляется десятичным числом, равным номеру города. Поэтому сама хромосома задается последовательностью чисел – номеров городов, по которым проходит маршрут.

Пример кодирования маршрута данным способом отображен на рисунке ниже.

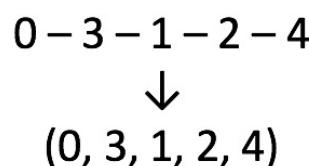


Рисунок 5 - Пример представления в виде пути обхода

Достоинства:

- Данный способ кодирования является наиболее естественным.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

1303.102002.001 ПЗ

Лист

32

Изм.	Лист	№ докум.	Подпись	Дата

Недостатки:

- Необходимость придумывать новые операторы кроссовера и мутации. [25]

3) Список смежности (adjacency representation)

При использовании данного способа кодирования ген представляется десятичным числом и зависит от позиции гена в хромосоме следующим образом: если ген находится на позиции n , а в маршруте обхода существует путь из n в m , то значение данного гена равно m .

Пример кодирования маршрута данным способом отображен ниже.

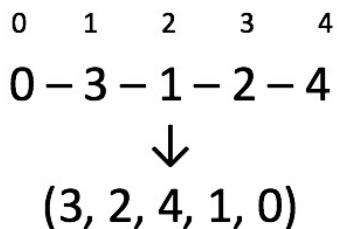


Рисунок 6 - Пример кодирования при помощи списка смежности

Достоинства:

- удобство применения на графах.

Недостатки:

- некоторые хромосомы могут не удовлетворять условиям задачи;
- необходимы разрабатывать новые операторы кроссовера. [25]

В качестве способа кодирования был выбран способ представления в виде пути обхода, так как при его использовании невозможно появление хромосом, не удовлетворяющих условиям задачи, а, следовательно, не придется нагружать алгоритм дополнительными проверками.

После выбора способа кодирования можно описать основные понятия, используемые в генетическом алгоритме, применительно к задаче коммивояжера:

- ген – это номер города;
- хромосома – порядок обхода городов;

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

33

- особь – маршрут обхода;
- популяция – конечное множество возможных маршрутов.

3.2.2 Функция оценки допустимости решения (целевая функция)

Так как целевая функция асимметричной ЗК является минимизация общей стоимости полученного маршрута, то целесообразно в качестве функции оценки допустимости решения, полученного генетическим алгоритмом, взять функцию нахождения суммарной стоимости обхода по маршруту.

3.2.3 Выбор начального подмножества решений

1) Решение, полученное случайным образом

Так как в качестве способа кодирования было выбран наиболее естественное представление пути как последовательности номеров городов, то случайное решение может быть получено одной из перестановок чисел от 1 до N , где N – число городов.

2) Решение, полученное алгоритмом ближайшего соседа

Подробное описание данного алгоритма описано в п.2.3.2. Данный алгоритм, как и другие «жадные» алгоритмы, может выдавать неоптимальные решения, однако из-за простоты его реализации решение, полученное данным алгоритмом, можно использовать в качестве начального.

3) Решение, полученное обходом местоположений по кругу относительно начального места

В условиях поставленной задачи в качестве начального города целесообразно взять пункт производства товара, так как именно из него

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

34

должен начинаться маршрут. Для получения решения данным способом необходимо выполнить следующие действия:

- a) координаты всех местоположений перенести в новую прямоугольную систему координат с центром в начальном городе;
- b) преобразовать прямоугольную систему координат в полярную;
- c) упорядочить координаты по мере увеличения полярного угла

Полученный порядок и будет обходом местоположений по кругу относительно начального места.

4) Решение, полученное обходом городов по мере увеличения расстояния относительно начального города

По тем же соображениям, что описаны в п.3) в качестве начального города целесообразно взять пункт производства товара. Для получения решения необходимо упорядочить все города по мере возрастания расстояния от них до начального пункта. Полученная отсортированная последовательность городов и будет искомым обходом городов.

Для того, чтобы обеспечить широкое разнообразие особей и изначально приблизить их к искомому решению, для создания начальной популяции были выбраны все вышеописанные способы, причем последние три способа генерируют по одной особи, а первый способ – до тех пор, пока размер популяции не достигнет своей верхней границы.

3.2.4 Выбор операторов скрещивания

1) Частичное отображение генов (Partially matched)

Частично отображающий кроссовер (partially-mapped crossover) или кроссовер PMX часть генотипа 1-го «родителя» s^1_p строго отображает в генотип «потомка» генотип потомка s_n , а остальная информация о генотипе 2-го «родителя» s^2_p изменяется после соответствующих

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

35

операций обмена так, чтобы сохранить порядок и позиции как можно большего числа городов из этого «родителя» в генотипе потомка и s_n . Используя допустимые генотипы двух «родительских» особей, кроссовер PMX сохраняет допустимость «потомства». [26] На рисунке 7 представлен пример работы частично отображающего кроссовера.

2) Рекомбинация дуг (Edge Recombination)

Кроссовер перестановки дуг (ER) основывается на перемещении общих дуг родителей к потомку. Для каждой из родительских особей составляется список смежности, а затем эти списки объединяются. В итоге выбирается та дуга, которая чаще всего встречалась в родительских особях. [25]

Данный способ удобен тем, что в качестве родителей для новой особи может выступать любое количество особей.

3) Циклический сдвиг (Cycle)

Циклический кроссовер (cycle crossover) или кроссовер CX формирует генотип потомка s_n таким образом, чтобы каждый город и его позиции переходили от генотипа одного из родителей. При этом он сохраняет в потомке абсолютные позиции городов из «родительских» генотипов, заданных перестановками s_p^1 и s_p^2 . Поэтому кроссовер CX по своей идее относится к классу позиционных кроссоверов. Для обеспечения с помощью циклического кроссовера допустимого генотипа s_n должны удовлетворяться следующие условия:

1. Каждая позиция генотипа потомка обязана сохранить имя города, найденного в соответствующей позиции одного из родителей.
2. Генотип потомка является перестановкой без пропусков и дублирования города. [26]

Пример работы циклического кроссовера можно рассмотреть на рисунке 8.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

36

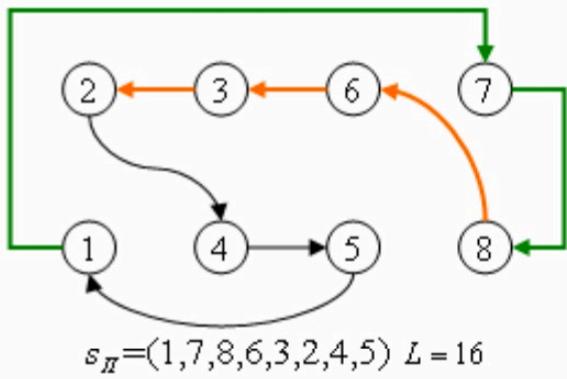
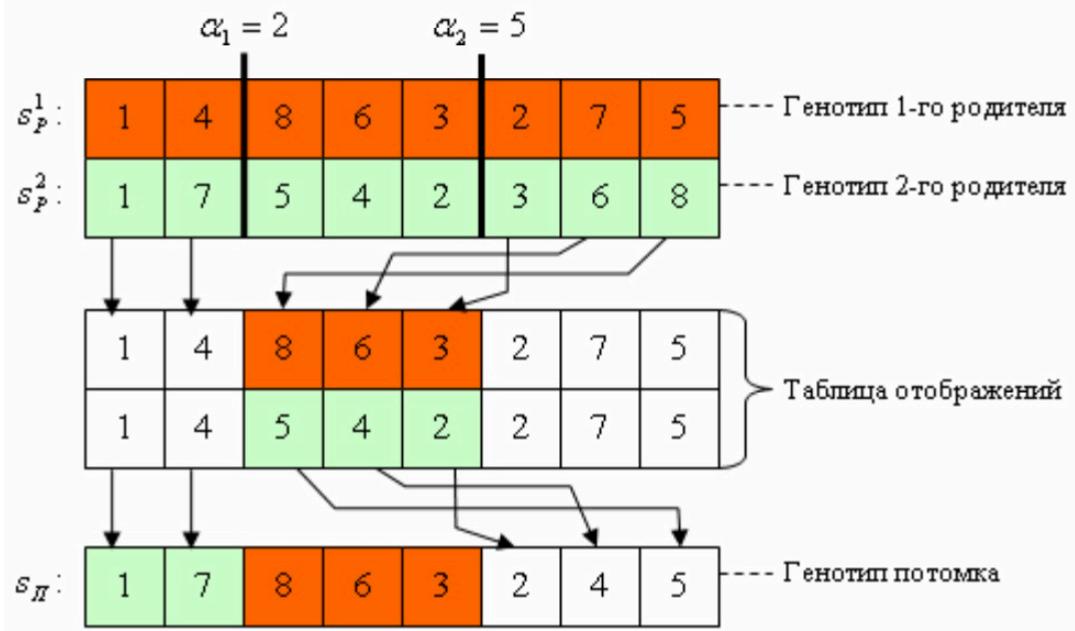


Рисунок 7 - Пример работы частично отображающего кроссовера [26]

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата
--------------	--------------	--------------	--------------	--------------

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

1303.102002.001 ПЗ

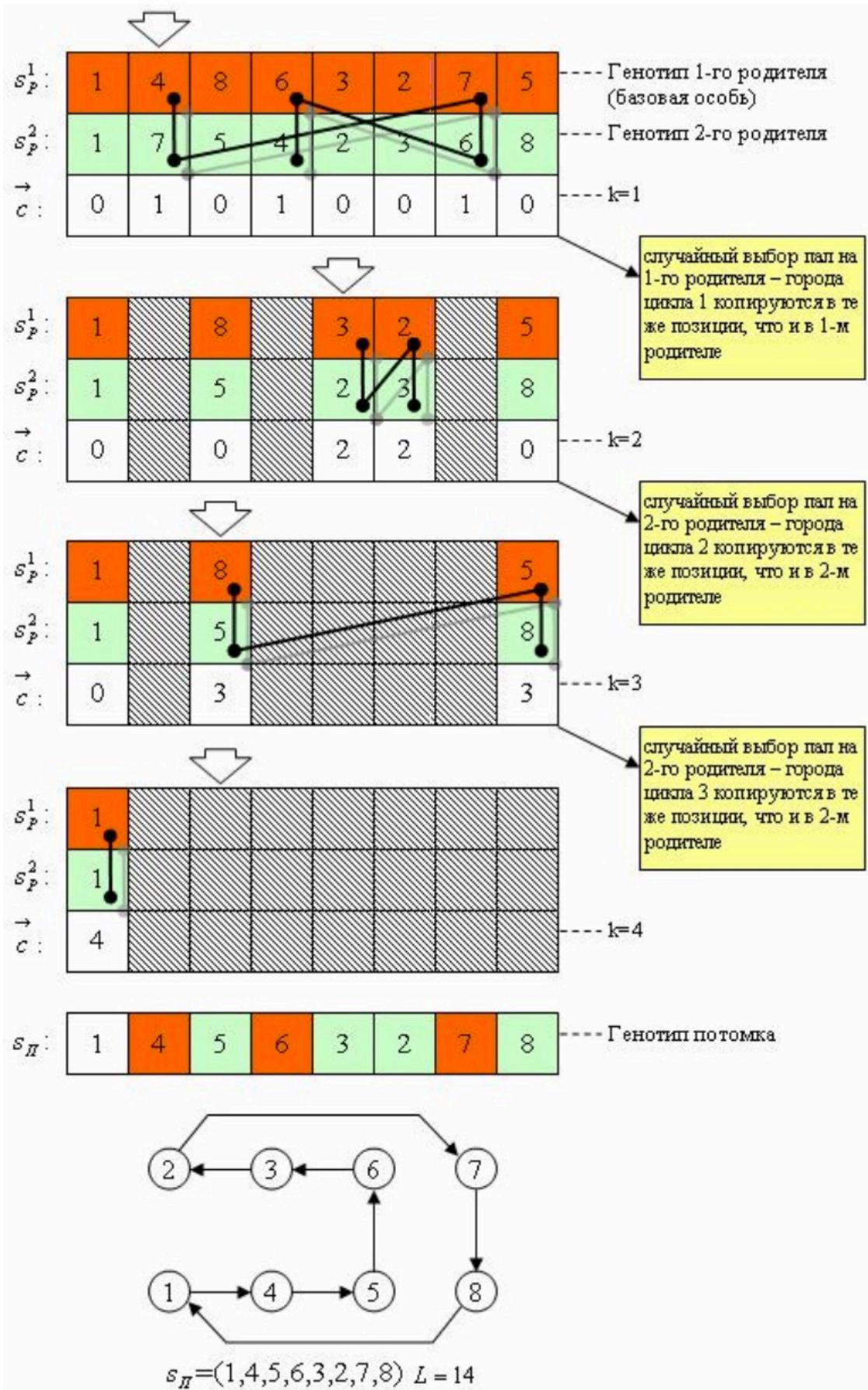


Рисунок 8 - Пример работы циклического кроссовера [26]

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №

1303.102002.001 ПЗ

3.2.5 Выбор оператора мутации

1) Перемещение гена

Данный оператор мутации основан на простом перемещении одного гена на другую случайную позицию. [25] Пример показан на рисунке 8.

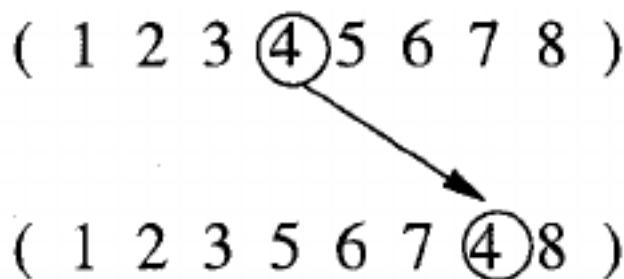


Рисунок 9 - Оператор мутации перемещения гена

2) Перемещение последовательности генов

При использовании данного оператора мутации случайным образом выбранная подпоследовательность генов хромосомы сдвигается на случайное число позиций. [25]

Пример работы данного оператора мутации описан на рисунке 10.

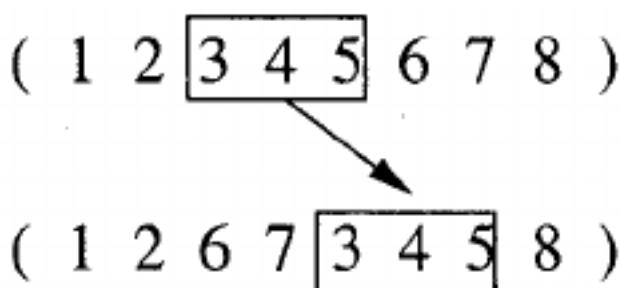


Рисунок 10 - Пример работы оператора мутации "Перемещение последовательности генов"

3) Перестановка двух генов

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист
39

При использовании данного оператора мутации случайным образом выбираются два гена, которые меняются местами. [25]

Пример работы данного оператора мутации приведен ниже.

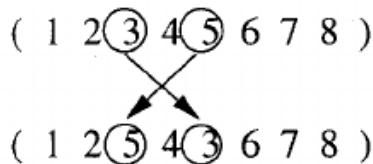


Рисунок 11 - Пример работы оператора мутации перестановки двух генов

4) Разворот последовательности генов

В данном операторе мутации случайно выбранная подпоследовательность генов хромосомы разворачивается. [25] Пример показан на рисунке 12.

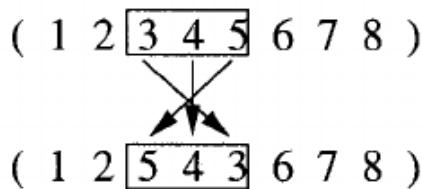


Рисунок 12 - Пример работы оператора разворота последовательности генов

5) Случайная перестановка части генов

В данном операторе мутации случайно выбранная подпоследовательность генов хромосомы перемешивается. [25] Пример отображен ниже.

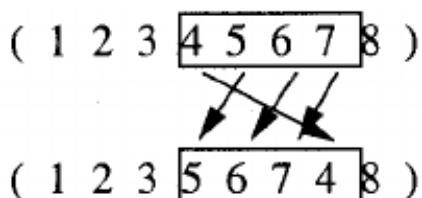


Рисунок 13 - Пример работы оператора мутации случайной перестановки последовательности генов

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

3.2.6 Выбор оператора селекции

1) Пропорциональная селекция

Родительские особи выбираются пропорционально значениям их функций приспособленности: каждой хромосоме сопоставлен сектор колеса рулетки, величина которого устанавливается пропорциональной значению функции приспособленности данной хромосомы, поэтому, чем больше значение функции приспособленности, тем больше сектор на колесе рулетки. Поэтому чем больше сектор на колесе рулетки, тем выше шанс, что будет выбрана именно эта хромосома.

Пример распределения вероятностей перехода особей в новую популяцию в соответствии с оператором селекции «колесо рулетки» показан на рисунке 7.

Недостаток данного метода в том, что он имеет случайный характер, а также особи с очень малым значением функции приспособленности слишком быстро исключаются из популяции, что может привести к преждевременной сходимости генетического алгоритма. [26]

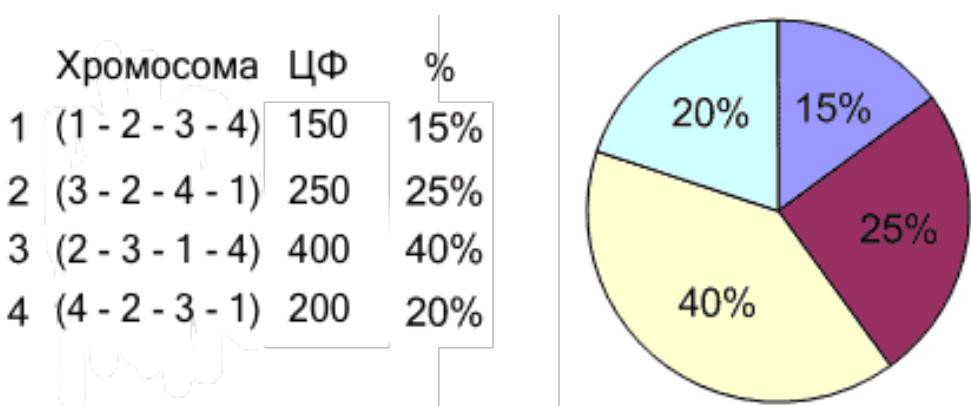


Рисунок 14 - Пример оператора пропорциональной селекции

2) Турнирная селекция

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв.	Подп. и дата

При турнирной селекции все особи популяции разбиваются на подгруппы с последующим выбором в каждой из них особи с наилучшей приспособленностью. Различаются два способа такого выбора: детерминированный выбор и случайный выбор. Детерминированный выбор осуществляется с вероятностью, равной 1, а случайный выбор – с вероятностью, меньшей 1. Подгруппы могут иметь произвольный размер, но чаще всего популяция разделяется на подгруппы по 2-3 особи в каждой. [26]

На рисунке ниже представлена схема, которая иллюстрирует метод турнирной селекции для подгрупп, состоящих из двух особей. Такую схему легко обобщить на подгруппы большего размера.

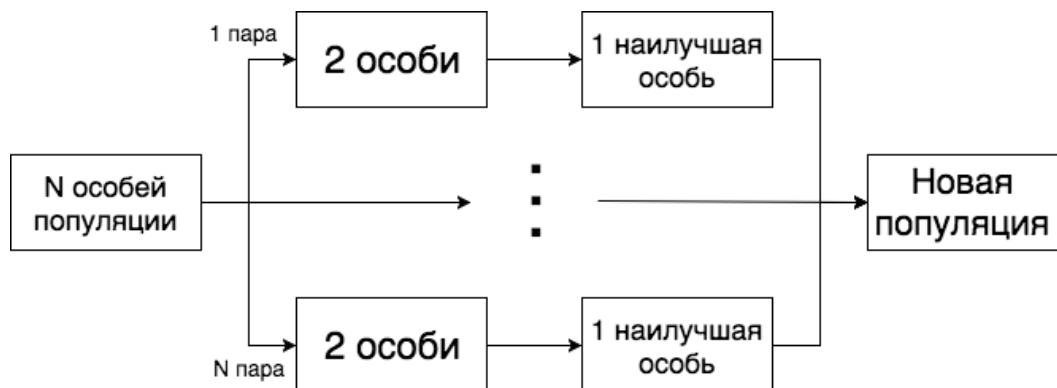


Рисунок 15 - Турнирная селекция

Недостаток данной селекции в том, что худший индивид никогда не выбирается.

3) Ранговая селекция

При ранговой селекции особи популяции ранжируются по значениям их функции приспособленности. Это можно представить себе как отсортированный список особей, упорядоченных по направлению от наиболее приспособленных к наименее приспособленным, в котором каждой особи приписывается число, определяющее ее место в списке и называемое рангом. В самом простом случае применительно к задаче

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

коммивояжера особи упорядочиваются по мере убывания значений целевой функции.

Ранговая селекция устраняет недостатки пропорциональной: нет стагнации, т.к. даже к концу работы алгоритма нет преждевременной сходимости, т.к. нет индивидов с вероятностью отбора близкой к единице. [26]

Также стоит отметить, что существуют особая процедура селекции, называемая элитарной стратегией.

Элитарная стратегия заключается в защите наилучших хромосом на последующих итерациях. В классическом генетическом алгоритме самые приспособленные особи не всегда переходят в следующее поколение. Это означает, что новая популяция не всегда содержит хромосому с наибольшим значением функции приспособленности из предыдущей популяции. Элитарная стратегия применяется для предотвращения потери такой особи. Эта особь гарантированно включается в новую популяцию. Преимущества: гарантия сходимости. Недостатки: большой риск захвата локальным оптимумом. [27]

В итоге для решения задачи коммивояжера были выбраны следующие операторы генетического алгоритма:

- Оператор рекомбинации дуг в качестве кроссовера;
- Перемещение и разворот последовательности генов в качестве операторов мутации;
- Турнирная селекция с применением элитарной стратегии

3.3 Алгоритмы и структуры данных

Общая схема генетического алгоритма выглядит следующим образом:

1. Выбрать начальную популяцию Р из N особей.
2. Пока не выполнен критерий остановки делать следующее:
 - 2.1. Выбрать всевозможные сочетания “родителей” S₁,...,S_n из популяции;

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв.	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

43

- 2.2. Для каждого сочетания родителей S_1, \dots, S_n выполнить:
- 2.2.1. Применить к S_1, \dots, S_n оператор скрещивания и получить новое решение S' ;
 - 2.2.2. Сгенерировать случайное число k с равномерным законом распределения в интервале $[0; 1]$. Если $k \leq PM$, то применить к S' оператор мутации и получить новое решение S'' ;
 - 2.2.3. Применить к S'' оператор локального улучшения и получить новое решение S''' ;
 - 2.2.4. Добавить S''' к популяции;
- 2.3. Отсортировать особи по мере увеличения значения целевой функции и оставить в популяции первые N особей. [28]

В виде блок-схемы данный алгоритм показан на рисунке 17.

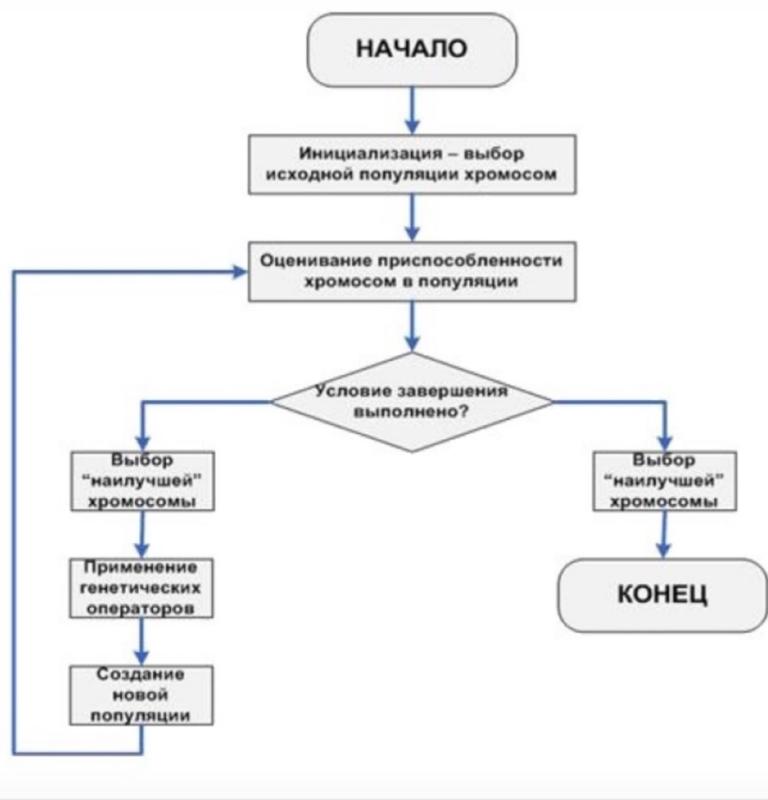


Рисунок 16 - Общая схема генетического алгоритма

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

- Обобщенный алгоритм работы оператора кроссовера рекомбинации дуг представлен ниже:
 0. Пусть P – список родительских особей, M – списки смежности.
 1. K – пустой список;
 2. N – случайный элемент из P ;
 3. Пока $\text{Длина}(K) < \text{Длина}(P)$:
 - 3.1. Добавить в конец списка K элемент N ;
 - 3.2. Удалить N из всех списков смежности;
 - 3.3. Если $M[N]$ не пустой:
 - 3.3.1. то $N^* :=$ случайный элемент списка $M[N]$;
 - 3.3.2. иначе N^* выбирается случайным образом так, чтобы не принадлежал K
 - 3.4. $N := N^*$ [29]
 - Алгоритм работы оператора мутации разворота:
 0. Пусть H – исходный массив переходов между городами, H' – получаемый после мутации массив, их длина = N ;
 1. Сгенерировать случайным образом 2 целых числа L и R из диапазона $[0, N-1]$;
 2. Если $R \leq L$, то поменять значения L и R между собой.
 3. В цикле для $i = [0, R-1]$ делать:
 - $H'[i] := H[i];$
 4. В цикле для $i = [0, (R-L)/2]$ делать:
 - $H'[L+i] := H[R-i];$
 5. В цикле для $i = [R+1, N-1]$ делать:
 - $H'[i] := H[i];$
 6. Вернуть H' .
 - Алгоритм работы оператора мутации сдвига:
 0. Пусть H – исходный массив переходов между городами, H' – получаемый после мутации массив, их длина = N ;

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

45

1. Сгенерировать случайным образом 2 целых числа L и R из диапазона [0, N-1];
 2. Если R <= L, то поменять значения L и R между собой.
 3. Сгенерировать случайным образом целое число M из диапазона [-L, N-R-1] – на сколько позиций будет сдвиг;
 4. В цикле для $i = [0, L+M-1]$ делать:
 - a. $H'[i] := H[i];$
 5. В цикле для $i = [L+M, R+M]$ делать:
 - a. $H'[i] := H[i-M];$
 6. В цикле для $i = [R+M+1, N-1]$ делать:
 - a. $H'[i] := H[i];$
 7. Вернуть H' .
- Обобщенный алгоритм работы оператора селекции:

Размер популяции ограничивается каким-либо натуральным числом N.

1. Взять наилучшую особь из исходной популяции и поместить в новую популяцию;
2. Упорядочить полученные после скрещивания и мутации особи по мере увеличения значений целевой функции;
3. Поместить в новую популяцию первые (N-1) особей.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

46

4 Программное обеспечение

4.1 Язык программирования и инструментальные средства разработки

4.1.1 Выбор языка программирования

Для разработки программы необходимо выбрать кроссплатформенный язык программирования с низким порогом вхождения и достаточным функционалом.

В современном мире кроссплатформенным языком можно назвать практически любой высокоуровневый язык к примеру C/C++ является кроссплатформенным на уровне компиляции. Для нас эти языки не подходят, во время переноса программы с одной серверной базы на другую необходимо заново собрать проект и провести полное тестирование.

Есть еще один тип кроссплатформенных языков на уровне среды исполнения. Для разработки программы этот вариант максимально оптимальный. Основными представителями данного класса языков программирования являются PHP и Java. Давайте подробнее рассмотрим их.

• Язык программирования PHP

PHP – В области веб-программирования, в частности серверной части, является мощнейший инструмент и поддерживает основной функционал для разработки полноценного адаптера.

Плюсами являются:

- Удобная и быстрая работа с POST и GET запросами;
- Легкое взаимодействие с множеством системами управления баз данных;
- Работа с локальными и удаленными файлами, а также с сокетами;
- Входит в распространенный набор программного обеспечения для создания и хостинга веб-сайтов;
- Синтаксис подобен синтаксису языка C;

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв.	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

47

- Является объектно-ориентированным.

Минусы:

- Отсутствие обратной совместимости версий;
- Больше подходит под написание скриптов.

- **Язык программирования Java**

Java - объектно-ориентированный язык программирования широко используемый в разработке.

Плюсы:

- Высокая производительность серверной части, написанной на Java;
- В наличии огромное количество разнообразнейших фреймворков;
- Развитое сообщество разработчиков;
- Каноничная реализация ООП;
- Удобная работа с XML файлами при использовании библиотеки dom4j;
- Приложение транслируется в байт-код и может воспроизводиться на любой операционной системе, где установлена Java Virtual Machine.

Минусы:

- Методика очищения памяти во время процесса работы программы.

Для разработки выбран язык Java потому что он переводится в специальный байт-код и может воспроизводиться на любой операционной системе где возможно поставить JVM. [30]

4.1.2 Выбор системы управления базой данных

При выборе СУБД основным критерием было так же кроссплатформенность легкость поддержания и разработки базы данных.

Такие СУБД, как SQLite моментально откладываются из-за их зависимости от платформы развертывания, хоть они и портированы на самые распространённые.

Рассмотрим бесплатно распространяемые СУБД MySQL и PostgreSQL. [31]

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

48

1. MySQL - свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей.

механизм репликации. Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей.

2. PostgreSQL- это мощная объектно-реляционная система управления базами данных с открытыми исходными текстами. Она разрабатывается на протяжении более 15 лет и улучшает архитектуру, чем завоевала репутацию надежной, интегрированной и масштабируемой СУБД. Она запускается на всех основных платформах, включая Linux, UNIX, и Windows. Она полностью соответствует ACID, имеет полную поддержку ключей, объединений, представлений, триггеров, и хранимых процедур (на разных языках).

В качестве используемой СУБД разработчиком выбрана MySQL. База данных PostgreSQL не обладает какими-либо недостатками по отношению к MySQL, ее использование обоснованно при реализации больших бизнес-систем, с огромным потоком данных, так как данная БД более «строгая» к запросам и пользователю, что позволяет усилить контроль над запросами.

4.1.3 Технология ORM

Чтобы запросы в базу данных осуществлялись не с помощью SQL-запросов, а с помощью обычных Java-методов, необходимо использовать ORM.

ORM (Object/Relational Mapping) - это способ сохранения объектов в реляционную базу данных. Другими словами ORM освобождает от работы с SQL

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв. №	Подп. и дата
Изм.	Лист	№ докум.	Подпись	Дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

49

и позволяет сосредоточиться на ООП. Основой ORM есть конфигурационный файл, в котором описывается как объект будет записываться в БД. Если с примитивными полями все так же примитивно: строковое поле записывается в колонку с типом varchar, целочисленное - в int, то со связями между объектами все интересней.

Например, у существуют классы Книга и Автор, для обоих созданы конфигурационные файлы, в которых описаны их поля. Но Книга, кроме всего прочего, имеет ссылку на Автора. В таком случае таблицы в БД будут связаны по внешнему ключу, что так же описывается в одном из конфигурационных файлов.

После выбора системы управления базы данных необходимо выбрать оболочку для работы с базой данных. Давайте сравним Фреймворк Yii и Hibernate.

Фреймворк Yii

Фреймворк Yii использует объекты доступа к данным DAO предоставляют API для доступа к данным, находящихся в различных СУБД. Это позволяет без сложностей сменить СУБД на любую другую без изменения исходного кода программы.

Основным минусом данного фреймворка является то что он написан на PHP так как мы выбрали основным языком программирования Java то Yii нам не подходит.

Фреймворк Hibernate

С помощью этого фреймворка работа с базой данных происходила не с помощью SQL-скриптов, а с помощью обычных Java-методов.

Hibernate - это один из самых популярных на сегодняшний день ORM-фреймворков. Само собой, что работа с БД посредством грамотно написанного SQL-кода будет более производительной, нежели с использованием Hibernate. Однако SQL нужно еще уметь грамотно написать, в то время, как Hibernate создает запросы оптимизированные. Так же у разработчиков уходит уйма времени на написание запросов и хранимых процедур, работа же с Hibernate ускоряет процесс разработки. Одной из привлекательных сторон Hibernate есть то, что он

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата
--------------	--------------	--------------	--------------	--------------

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

1303.102002.001 ПЗ

Лист

50

поддерживает диалекты ко всем популярным СУБД. Наши приложения становятся еще и СУБД-независимыми, т.к. единственное, что придется поменять если мы хотим перейти на другую СУБД, - это диалект в конфигурации Hibernate. [32]

Для функционирования Hibernate нужны:

- Главный конфигурационный файл, в котором описываются параметры соединения к БД.
- Описание связи между классом и таблицей. Включает в себя связь между полем класса и колонкой таблицы. Также здесь описывается связь классов друг с другом.

В качестве используемого ORM-фреймворка был выбран Hibernate, так как Yii не подходит для использования с ЯП Java.

4.1.4 Стек используемых технологий

Для реализации клиентской части приложения было решено использовать распространенную связку веб-технологий HTML+CSS+JavaScript.

Выбран JavaScript, потому что на данный момент этот язык развивается стремительно создается множество полезных библиотек для работы с разными сферами веб-разработки.

В качестве используемой среды разработки была выбрана NetBeans IDE, так как она распространяется бесплатно и содержит все необходимые инструменты.

В качестве сервера веб-приложений выбран Apache Tomcat, так как он имеет высокую скорость развертывания приложений, а также легок в настройке.

В итоге общий стек используемых технологий в разработке программного обеспечения выглядит следующим образом:

- Язык программирования Java;
- СУБД MySQL;
- Веб технологии HTML+CSS+JavaScript;

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв.	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

51

- Сборщик проекта Apache Maven;
- ORM технология Hibernate;
- Spring Framework.

4.2 Описание структуры программного продукта

Описание структуры программного продукта приводится в приложении 3 «Руководство программиста».

4.3 Описание интерфейса пользователя

Подробное описание интерфейса пользователя находится в Приложении 1 «Руководство пользователя».

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

52

5 Оценка качества решения

5.1 Тестирование ПО

5.1.1 Тестирование в нормальных условиях

Под нормальными условиями понимаются:

- Сервер приложений и сервер БД находятся в одной сети и имеют доступ в сеть интернет;
- Пользователь использует для подключения веб-браузер Chrome версии 28 и выше;
- Пользователь вводит корректные данные.

Ввод данных и их просмотр пользователем группы «Администратор»:

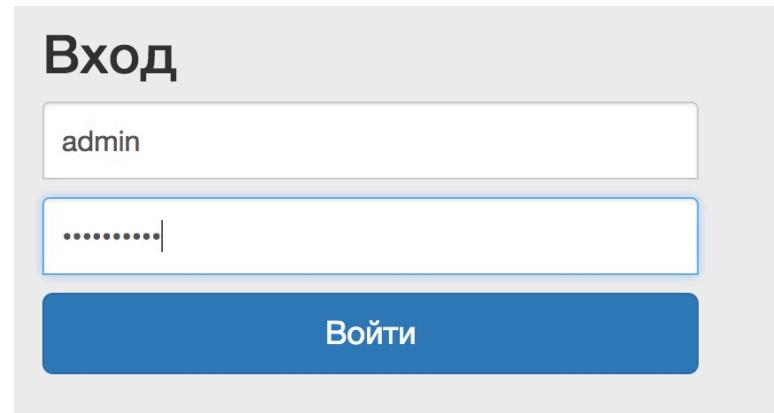


Рисунок 17 - Вход в систему администратора

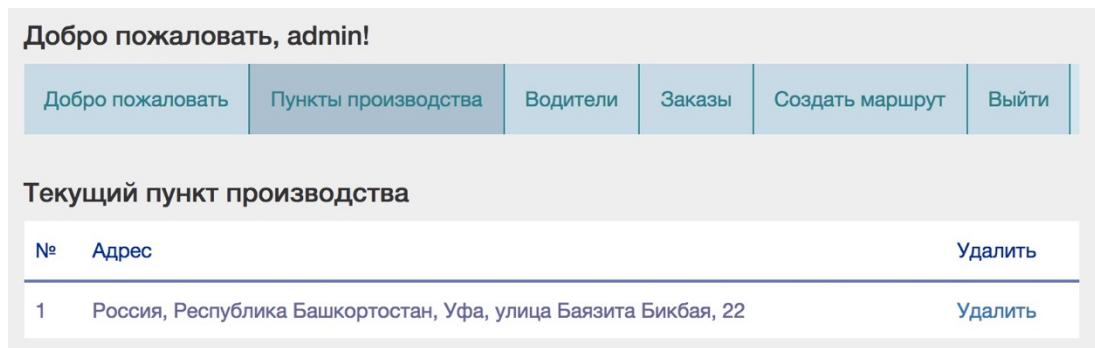


Рисунок 18 - Добавление пункта производства в систему

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата	Лист
					1303.102002.001 ПЗ
Изм.	Лист	№ докум.	Подпись	Дата	53

Создание аккаунта

Хаматханов Ирек Винерович

irek

••••

••••

Россия, Республика Башкортостан, Уфа, Российская улица, 25

Создать

Рисунок 19 - Регистрация водителей в системе

Добро пожаловать, admin!					
Добро пожаловать	Пункты производства	Водители	Заказы	Создать маршрут	Выйти
Добавить нового водителя					
Текущие водители:					
№	ФИО	Пункт производства, на котором работает	Просмотр маршрутов водителя	Уволить	
1	driver1	Россия, Республика Башкортостан, Уфа, улица Баязита Бикбая, 22	Посмотреть	Уволить	
2	Хаматханов Ирек Винерович	Россия, Республика Башкортостан, Уфа, улица Баязита Бикбая, 22	Посмотреть	Уволить	

Рисунок 20 - Просмотр списка водителей

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Добро пожаловать, admin!

Добро пожаловать	Пункты производства	Водители	Заказы	Создать маршрут	Выйти
------------------	---------------------	----------	--------	-----------------	-------

Добавить заказ

Полный адрес (город, улица, дом)

Дата доставки заказа: 12.06.2017

Описание заказа

Добавить заказ

Заказы :

№	Дата	Адрес	Статус	Описание	Отменить заказ
1	2017-06-12	Россия, Республика Башкортостан, Уфа, Заливной переулок, 24	Не выполнен	-	Отменить
2	2017-06-12	Россия, Республика Башкортостан, Уфа, Иксский переулок, 9	Не выполнен	-	Отменить
3	2017-06-12	Россия, Республика Башкортостан, Уфа, микрорайон Максимовка, Совхозная улица, 5	Не выполнен	-	Отменить
4	2017-06-12	Россия, Республика Башкортостан, Уфа, Посадская улица, 20	Не выполнен	-	Отменить

Рисунок 21 - Добавление заказов в систему

Добро пожаловать, admin!

Добро пожаловать	Пункты производства	Водители	Заказы	Создать маршрут	Выйти
------------------	---------------------	----------	--------	-----------------	-------

Создать маршрут

Выберите водителя: Хаматханов Ирек Винерович Выберите дату: 12.06.2017

Создать маршрут

Рисунок 22 - Создание маршрута для водителя

После создания маршрута в меню «Водители» - «Просмотр маршрутов водителя» будет отображен новый маршрут, как показано ниже.

Добро пожаловать, admin!

Добро пожаловать	Пункты производства	Водители	Заказы	Создать маршрут	Выйти
------------------	---------------------	----------	--------	-----------------	-------

Маршруты irek:

№	Дата	Статус	Посмотреть маршрут
1	2017-06-12	Не выполнен	Посмотреть Отметить выполненным

Рисунок 23 - Просмотр списка маршрутов водителя

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв. №	Подп. и дата
--------------	--------------	--------------	----------------	--------------

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					55

1303.102002.001 ПЗ

После этого в личном кабинете данного водителя отобразится построенный маршрут.

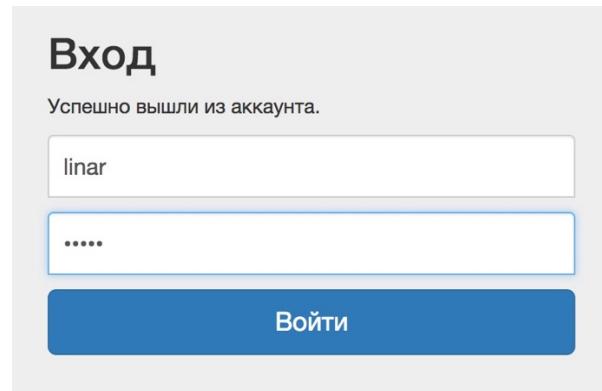


Рисунок 24 - Вход в систему водителя

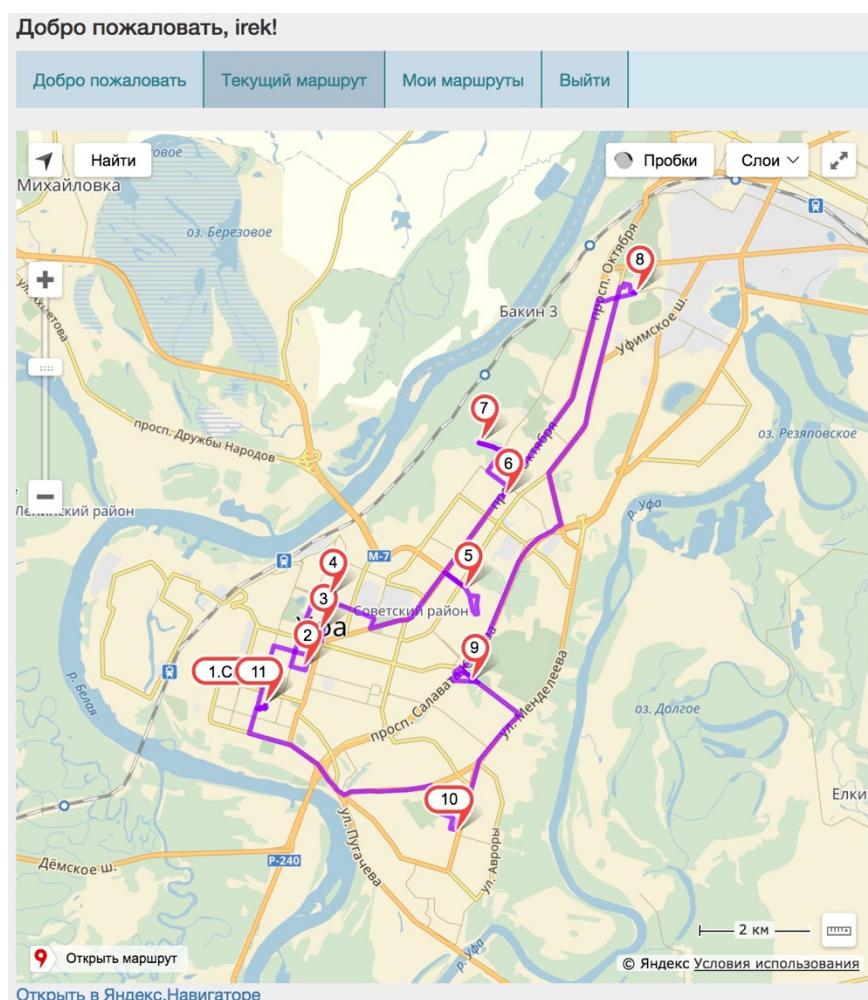


Рисунок 25 – Текущий маршрут водителя на карте

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

1303.102002.001 ПЗ

Лист

56

Ниже описан данный маршрут в текстовом виде.

Трогаемся,
Едем прямо на Майская улица, проезжаем 105.98 м.,
Едем налево на Волгоградская улица, проезжаем 297.78 м.,
Едем направо на Байконурская улица, проезжаем 396.75 м.,
Едем налево на Гвардейская улица, проезжаем 628.45 м.,
Едем направо на Шоссейная улица, проезжаем 274.81 м.,
Едем направо на Шоссейная улица, проезжаем 103.87 м.,
Едем прямо, проезжаем 873.48 м.,
Едем прямо на Перспективная улица, проезжаем 6878.42 м.,
Едем налево на Вологодская улица, проезжаем 152.12 м.,
Едем прямо, проезжаем 174.59 м.,
Едем прямо на Сельская Богородская улица, проезжаем 5339.39 м.,
Едем налево на улица Маршала Жукова, проезжаем 4180.42 м.,
Едем прямо, проезжаем 208.98 м.,
Едем прямо на Сипайлловская улица, проезжаем 1145.9 м.,
Едем прямо на проспект Салавата Юлаева, проезжаем 6956.02 м.,
Едем прямо, проезжаем 11.9 м.,
Едем съезд направо на улица Заки Валиди, проезжаем 91.74 м.,
Едем направо, проезжаем 139.89 м.,
Едем прямо на улица Заки Валиди, проезжаем 35.26 м.,
Едем прямо на Сочинская улица, проезжаем 523.07 м.,
Едем правее, проезжаем 219.72 м.,

Рисунок 26 – Текущий маршрут водителя в текстовом виде

5.1.2 Тестирование в экстремальных условиях

При вводе некорректного логина или пароля, система выдаст соответствующее сообщение об ошибке.

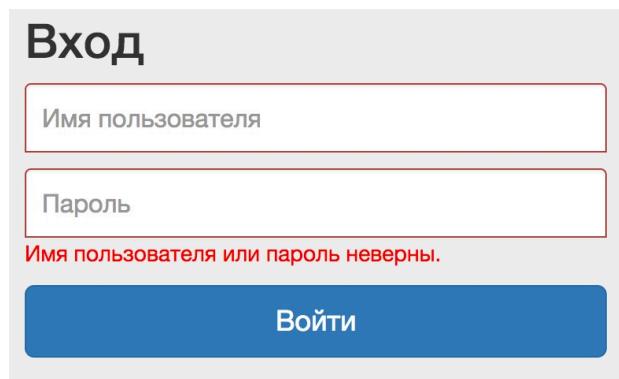


Рисунок 27 - Ввод некорректного логина / пароля

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист
57

При вводе некорректных данных при регистрации водителей в системе также будут выданы подсказки.

Создание аккаунта

Полное имя

aa

Имя пользователя должно быть не короче 4 и не длиннее 32 символов.

..

Пароль должен быть длиннее 4 символов.

.

Пароли не совпадают.

Россия, Республика Башкортостан, Уфа, улица Цюрупы, 149

Создать

Рисунок 28 - Ввод некорректных данных при регистрации

При вводе некорректного адреса при добавлении пункта производства или заказа в систему, данный адрес не получится геокодировать, и поэтому он не будет добавлен в систему.

Добро пожаловать, admin!

Добро пожаловать Склады Водители Заказы Выйти

mfghblkjgfhngfjklhdh

Добавить склад

Текущие склады:

№	Адрес склада	Удалить склад
1	Россия, Республика Башкортостан, Уфа, улица Цюрупы, 149	Удалить
2	Россия, Республика Башкортостан, Уфа, Российская улица, 25	Удалить

Рисунок 29 - Ввод некорректного адреса

Инв. № подп.	Подп. и дата
Изм.	Лист

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

58

5.1.3 Тестирование в исключительных условиях

- Если отсутствует подключение к веб-серверу приложений, то в таком случае страница веб-страница данного приложения в браузере не загрузится.
- Если отсутствует подключение к БД, но при этом веб-сервер приложений имеет доступ к сети, то в данном случае будет невозможен вход в систему, так как все логины и пароли хранятся в БД. При этом будет выдано сообщение о неправильном вводе логина и пароля.

5.1.4 Анализ тестирования

Тестирование показало, что программный продукт корректно выполняет свои функции. В исключительных ситуациях программа выдает сообщение об ошибке и продолжает работу в штатном режиме.

5.2 Оценка качества программного обеспечения

Оценка качества программного продукта осуществляется согласно ГОСТ 28195-89. Выбор номенклатуры показателей качества для данного программного продукта осуществляется с учетом его требований и области применения. Данный программный продукт относится к подклассу 509- «Прочие ПС».

Полные результаты вычисления показателей качества разработанного программного обеспечения представлены в приложении 1.

В результате проведенного анализа получили следующие оценки факторов качества программного средства, которые представлены в таблице 3.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

59

Таблица 3 - Оценки факторов качества программного средства

Фактор	Оценка
Надежность (ФН)	0,822
Сопровождение (ФС)	0,73
Удобство применения (ФУП)	0,97
Эффективность (ФЭ)	0,983
Универсальность (ФУн)	0,81
Корректность (ФКт)	0,924

5.3 Вычислительный эксперимент и анализ результатов

Перед проведением вычислительного эксперимента составлено 18 модификаций генетического алгоритма комбинацией следующих параметров:

- Оператор скрещивания рекомбинацией дуг:
 1. С 2-мя родителями;
 2. С 3-мя родителями;
- Оператор мутации:
 1. Разворот последовательности генов (Reverse);
 2. Сдвиг последовательности генов (Move);
 3. Разворот и сдвиг (Reverse + Move).
- Вероятность мутации:
 1. 0,1
 2. 0,3;
 3. 0,5.

Для удобства будем обозначать алгоритмы по следующему правилу:

«{2;3} {R;M;RM} {0,1;0,3;0,5}» , где

- {2,3} – кол-во родителей, участвующих в операторе кроссовера;

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата	1303.102002.001 ПЗ	Лист
						60

- {R;M;RM} – оператор мутации (Reverse, Move, Reverse+Move соответственно);
- {0,1;0,3;0,5} – значение вероятности мутации.

Например, записи «2M0,5» соответствует генетический алгоритм использованием оператора скрещивания рекомбинацией дуг с 2-мя родителями, оператора мутации разворота последовательности генов с вероятностью мутации = 0,5.

Вычислительный эксперимент проводился в 2 этапа: в первый раз – на основе существующих решенных задач коммивояжера, во второй раз – на реальных случайных данных.

- **Вычислительный эксперимент на основе существующих решенных задач коммивояжера**

Для проведения данного этапа эксперимента составлена следующая методика:

- I. В качестве исходных данных взяты матрицы расстояний из проекта TSPLIB, который предоставляет частные случаи задач коммивояжера разной размерности и их **оптимальные** решения.
- II. Написан модуль программы,читывающий исходные данные из файла и проверяющий решение на оптимальность.
- III. Каждая задача решается 10 раз каждой из модификаций генетического алгоритма.
- IV. Полученные значения целевой функции для каждого из 10-ти повторных решений усредняются, сравниваются с оптимальным значением и вычисляется точность решения по формуле

$$T = \frac{F - OPT}{F} * 100\%, \quad (8)$$

где Т – точность в процентах,

F – усредненное значение целевой функции алгоритма,

OPT – оптимальное решение задачи.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист	61
------	----

V. На основе полученных результатов делается предположение о точности модификаций алгоритма. [32]

Результаты проведения первого этапа вычислительного эксперимента представлены в таблице 4.

Таблица 4 - Результаты первого этапа вычислительного эксперимента (точность)

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата	Подп. и дата	Кол-во городов / Алгоритм	17	26	43	53	70	100
						2R0,1	99,5%	91,0%	99,5%	77,7%	89,7%	71,8%
						2M0,1	99,7%	92,0%	99,6%	78,7%	89,9%	73,2%
						2RM0,1	100,0%	93,9%	99,5%	78,9%	89,8%	72,5%
						3R0,1	100,0%	96,7%	99,7%	80,2%	90,3%	74,8%
						3M0,1	100,0%	96,1%	99,5%	79,2%	90,5%	74,7%
						3RM0,1	100,0%	96,2%	99,6%	79,8%	90,3%	73,5%
						2R0,3	99,7%	94,8%	99,6%	78,4%	89,9%	72,1%
						2M0,3	99,2%	94,3%	99,6%	78,3%	89,8%	71,6%
						2RM0,3	100,0%	93,7%	99,5%	78,9%	89,8%	70,9%
						3R0,3	100,0%	95,9%	99,7%	80,0%	90,3%	74,2%
						3M0,3	100,0%	96,2%	99,7%	80,1%	90,3%	74,8%
						3RM0,3	100,0%	95,2%	99,7%	80,1%	90,6%	74,9%
						2R0,5	100,0%	93,8%	99,6%	78,1%	89,7%	72,9%
						2M0,5	99,2%	94,0%	99,6%	78,6%	89,9%	70,9%
						2RM0,5	99,7%	93,7%	99,5%	78,5%	89,7%	71,9%
						3R0,5	100,0%	94,9%	99,8%	80,2%	90,6%	74,4%
						3M0,5	100,0%	95,4%	99,7%	80,1%	90,5%	74,4%
						3RM0,5	100,0%	97,0%	99,7%	79,7%	90,4%	74,3%

Жирным в таблице выделены наибольшие значения в каждом столбце.

Изм.	Лист	№ докум.	Подпись	Дата	1303.102002.001 ПЗ	Лист
						62

Жирным в таблице 4 выделено наибольшее значение точности в каждом столбце.

Также на рисунке 31 отображен график зависимости значений точности модификации алгоритма от количества городов.

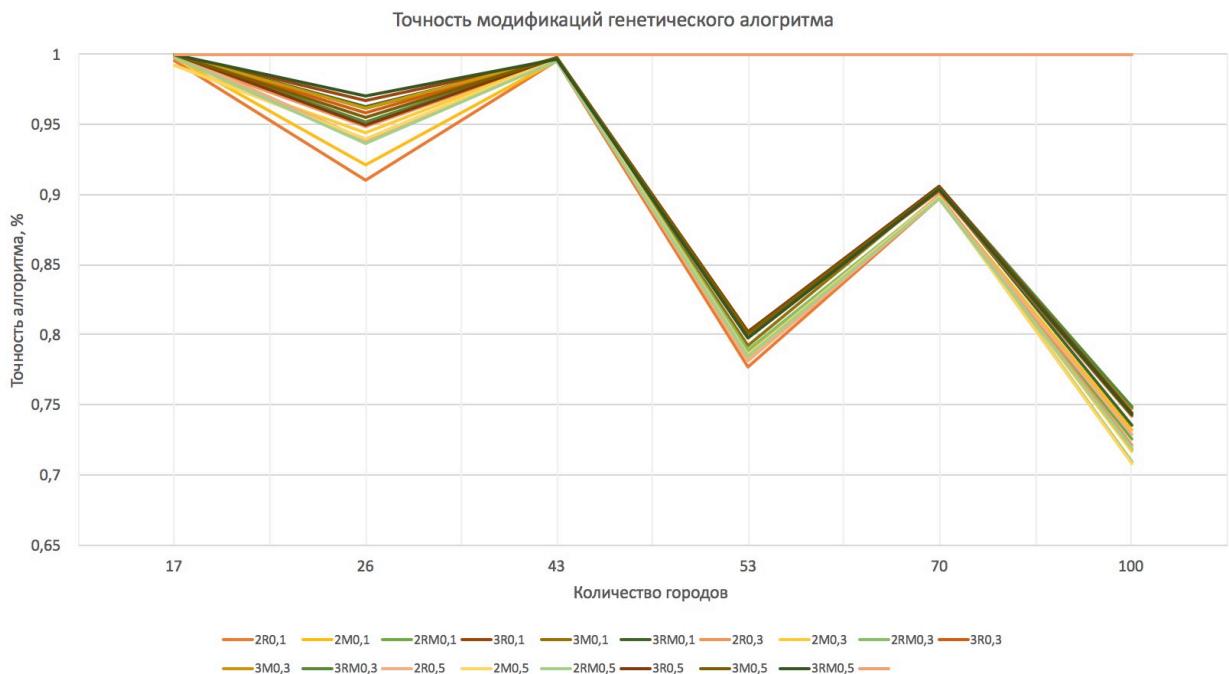


Рисунок 30 - Результаты первого этапа вычислительного эксперимента (точность)

Исходя из полученных результатов можно сделать вывод о том, что точность алгоритма не зависит от количества городов в условии задачи и варьируется в диапазоне [70%, 100%]. Также заметно, что точность алгоритма практически не изменяется в зависимости от модификации алгоритма. Наиболее точными алгоритмами в целом являются 3R0,5 и 3RM0,3. Можно предположить, что на задачах меньшей размерности (< 50 городов) алгоритм ищет решения лучше, чем на задачах большой размерности (> 50 городов).

- Вычислительный эксперимент на основе реальных данных**

Для проведения данного этапа эксперимента составлена следующая методика:

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

- I. В качестве исходных адресов для проведения эксперимента были взяты ~15000 адресов г. Уфы и Уфимского района из Федеральной информационной адресной системы (ФИАС);
- II. Написан модуль программы, генерирующий случайную выборку от 10 до 100 адресов.
- III. Для каждой выборки решается задача построения рационального маршрута 10 раз каждой из модификаций генетического алгоритма.
- IV. Полученные значения целевой функции для каждого 10-ти повторных решений усредняются и сравниваются между собой.
- V. На основе полученных результатов делается вывод об эффективности модификаций алгоритма и выбирается наилучшая.[33]

Результаты проведения второго этапа вычислительного эксперимента представлены в приложении .

Также на рисунке 32 отображен график зависимости значений целевой функции от количества заявок на доставку для каждой модификации алгоритма.

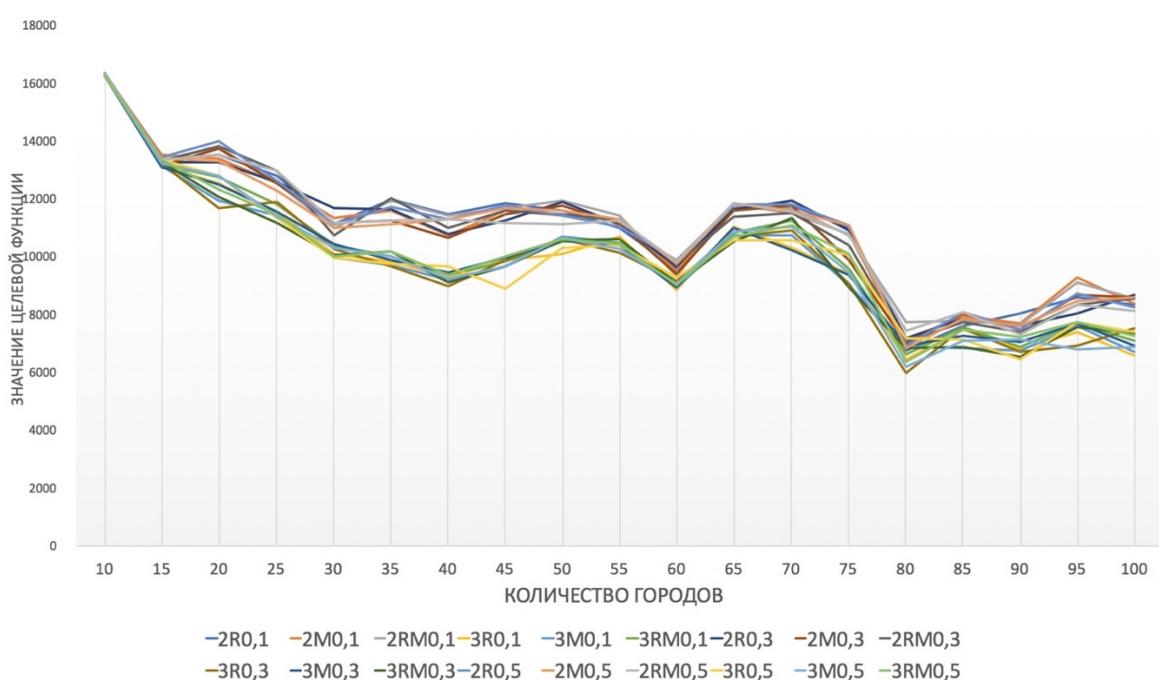


Рисунок 31 – Результаты второго этапа вычислительного эксперимента

Изм.	Лист	№ докум.	Подпись	Дата	Лист	1303.102002.001 ПЗ	64

На данном графике видно, что линии «разделились» на 2 группы – верхняя и нижня. Группе, которая находится ниже, соответствуют алгоритмы, в которых применялся оператор кроссовера с 3-мя родителями, а верхней группе – с 2-мя родителями.

Вывод: на основе полученных результатов можно предположить, что алгоритмы с использованием оператора кроссовера с 3-мя родителями лучше справляется с решением задачи, чем с 2-мя. Среди используемых операторов мутации лучше всего проявил себя REVERSE. Трудно сделать вывод о том, при какой вероятности мутации алгоритм ведет себя лучше, однако в большинстве случаев лучшее решение находилось при вероятности мутации = 0,3.

Поэтому для разработки приложения использовался алгоритм 3R0,3.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

65

ЗАКЛЮЧЕНИЕ

В ходе проделанной работы были получены следующие результаты выпускной квалификационной работы:

- 1) Проведен обзор существующего ПО для поиска оптимальных маршрутов.
Выяснено, что не существует бесплатных программ, которые строят и визуализируют на карте рациональные маршруты из большого количества городов.
- 2) Выполнен аналитический обзор методов решения задачи. На основе анализа выбран генетический алгоритм в силу его большой гибкости при настройке параметров.
- 3) Разработаны и программно реализованы различные модификации генетического алгоритма.
- 4) Произведен вычислительный эксперимент, который показал:
 - a. Точность решения задачи не зависит от количества городов;
 - b. В операторе кроссовера использование трех родительских особей улучшает результат;
 - c. В качестве оператора мутации лучше всего себя проявил оператор разворота последовательности генов.
- 5) Написано программное обеспечение для построения рационального маршрута и визуализации его на карте.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

66

СПИСОК ЛИТЕРАТУРЫ

1. Хемди А. Таха Введение в исследование операций, 7-е издание. С.-Петербург: Вильямс, 2005. – 901 с.
2. Колесников А.В., Кириков И.А., Листопад С.В., Румовская С.Б., Доманицкий А.А. Решение сложных задач коммивояжера методами функциональных гибридных интеллектуальных систем / Под ред. А.В. Колесникова. — М.: ИПИ РАН, 2011. — 295 с.
3. Губко М.В. Введение в исследование операций: Курс лекций. – [Электронный ресурс]. – Режим доступа: http://mipt.ipu.ru/sites/default/files/page_file/1. Введение в исследование операций.pdf.
4. Ли К. Основы САПР (CAD/CAM/CAE) / Кунву Ли. – СПБ.: Питер, 2004.
5. Википедия – свободная энциклопедия. Задача коммивояжёра. – [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Задача_коммивояжёра.
6. Левитин А.В. Алгоритмы. Введение в разработку и анализ: пер. с англ. М.; СПб.; Киев: Вильямс, 2006.
7. Kona H., Burde A., Dr. Zanwar D. R. A Review of Traveling Salesman Problem with Time Window Constraint // IJIRST – International Journal for Innovative Research in Science & Technology, 2015. Vol. 2, Issue 1. P. 253– 254.
8. Clausen J. Branch and Bound Algorithms – Principles and Examples. University of Copenhagen, 1999. P. 5-6.
9. Tollis I. G. Algorithms and Complexity. University of Crete, 2000. P. 140–146.
10. Applegate D., Bixby R., Chvatal V., Cook W. TSP cuts outside the template paradigm. Donet, 2000. P. 1–10.
11. Захарова Е.М., Минашина И.К. Обзор методов многомерной оптимизации // Информационные процессы, 2014. Том 14, № 3. стр. 265–266.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв.	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

67

- | | |
|--------------|--------------|
| Инв. № подп. | Подп. и дата |
| Инв. № дубл. | Взам. Инв. № |
| Подп. и дата | |
| Подп. и дата | |
- 12.Papadimitriou C., Vazirani U. Efficient Algorithms and Intractable Problems. Berkeley EECS, 1999. Chapter 10.
- 13.Goodrich M., Roberto T. Algorithm Design and Applications, Wiley, 2015. P. 513–514.
- 14.Nilsson C. Heuristics for the Traveling Salesman Problem. Linkoping University, 2011. P. 1–6.
- 15.Alsalibi B.A., Jelodar M.B., Venkat I. A Comparative Study between the Nearest Neighbor and Genetic Algorithms: A revisit to the Traveling Salesman Problem // International Journal of Computer Science and Electronics Engineering (IJCSEE), 2013. Vol. 1, Issue 1.
- 16.Gutin G.,Yeo A. The Greedy Algorithm for the Symmetric TSP. University of London, 2002. P. 1–2.
- 17.Gutin G., Karapetyan D. Lin-Kernighan Heuristic Adaptations for the Generalized Traveling Salesman Problem. Royal Holloway London University, 2010. P. 1–5.
- 18.Gupta R., Chauhan C., Pathak K. Survey of Methods of Solving TSP along with its Implementation using Dynamic Programming Approach. // International Journal of Computer Applications, 2012. Vol. 52, No.4. P. 1–6.
- 19.Basu S. Tabu Search Implementation on Traveling Salesman Problem and Its Variations: A Literature Survey. Indian Institute of Management Calcutta, 2012. P. 1–8.
- 20.Dorigo M., Caro G. D. Ant Algorithms for Discrete Optimization // Artificial Life, 1999. Vol. 5, No 2. P. 139–140.
- 21.Dorigo M., Gambardella L. M. Ant colonies for the traveling salesman problem. Université Libre de Bruxelles, 1996. P. 1–4.
- 22.Батищев Д.И., Неймарк Е.А., Старостин Н.В. Применение генетических алгоритмов к решению задач дискретной оптимизации: Учебное пособие. – Н.Новгород, изд-во ННГУ им. Н.И.Лобачевского, 2006. – 136с.

Инв. № подп.	Подп. и дата
Инв. № дубл.	Взам. Инв. №
Подп. и дата	
Подп. и дата	

1303.102002.001 ПЗ

Лист

68

Изм.	Лист	№ докум.	Подпись	Дата

- | | |
|--------------|--------------|
| Инв. № подп. | Подп. и дата |
| Инв. № дубл. | Взам. Инв. № |
| Подп. и дата | |
| Подп. и дата | |
23. Алексеева Е.В. Лекция 4. Задача коммивояжера. – [Электронный ресурс]. – НГУ, 2012. – Режим доступа: http://math.nsc.ru/~alekseeva/DP-MMF-2013/L4_mmf_2012_2013.pdf
24. Орехов Ю.В. Исследование эффективности операторов генетического алгоритма при поиске экстремальных решений в бинарных и вещественных пространствах / Ю.В.Орехов, Э.Ю.Орехов, А.Р.Заминова // Информационные технологии. – 2008. - № 9. – С. 65 – 72.
25. Example: The Travelling Salesman Problem. Handouts. – [Электронный ресурс]. – Istanbul Technical University, 2015. Р. 1-4. – Режим доступа: http://web.itu.edu.tr/etaner/courses/NIC/handouts/example_TSP_handouts.pdf
26. Батищев Д.И., Неймарк Е.А., Старостин Н.В. Применение генетических алгоритмов к решению задач дискретной оптимизации: Учебное пособие. — Нижний Новгород: Изд-во ННГУ им. Н.И. Лобачевского, 2007. – 181с.
27. Батищев Д.И., Костюков В.Е., Неймарк Е.А., Старостин Н.В. Решение дискретных задач с помощью эволюционно-генетических алгоритмов: Учебное пособие. — Нижний Новгород: Изд-во ННГУ им. Н.И. Лобачевского, 2011. — 199 с.
28. Edge recombination operator. – [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Edge_recombination_operator.
29. Эккель Б. Философия Java: учеб. пособие по ООП. 4-е изд. – СПб.: Питер, 2011. – 640с.
30. Сравнение реализаций диалектов SQL. – [Электронный ресурс]. – Режим доступа: <https://programming-ray.blogspot.com>
31. P.Fisher, B.D. Murphy. Spring Persistence with Hibernate. 1st edition – Apress, 2010. Р. 164
32. Мухачёва А.С., Усманова А.Р. Методика тестирования и анализа результатов вероятностных алгоритмов локального поиска оптимума. Методические указания. – Уфимск. гос. авиац. техн. ун-т. Уфа, 2003. -27с.

Инв. № подп.	Подп. и дата
Инв. № дубл.	Взам. Инв. №
Подп. и дата	
Подп. и дата	

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

69

ПРИЛОЖЕНИЕ А
(обязательное)
ТЕХНИЧЕСКОЕ ЗАДАНИЕ
ВВЕДЕНИЕ

1 Наименование программы

Наименование - «Автоматизированная система учета заказов и нахождения пути минимальной длины для службы доставки».

2 Краткая характеристика области применения программы

Программа предназначена к применению в информационных системах организаций, предоставляющих услуги доставки товаров.

1 Основание для разработки

Основанием для проведения разработки является задание на выполнение выпускной квалификационной работы.

2 Назначение разработки

2.1 Функциональное назначение программы

Функциональным назначением программы является регистрация заказов и построение маршрута минимальной длины, проходящего по всем местоположениям, соответствующим заказам, ровно один раз.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

70

2.2 Эксплуатационное назначение программы

Ограничений на эксплуатацию программы не вводится.

3 Требования к программе

3.1 Требования к функциональным характеристикам

3.1.1 Требования к составу выполняемых функций

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

- 1) вход пользователей в систему / выход из системы;
- 2) регистрация и удаление водителей как пользователей системы;
- 3) добавление и удаление складов;
- 4) регистрация заказов на доставку товара с указанием даты и местоположения доставки;
- 5) построение маршрута минимальной длины, проходящего по всем местоположениям, соответствующим заказам, ровно один раз;
- 6) отображение маршрута на карте, отображение помощи в навигации по маршруту в текстовом виде (или интеграция со сторонними приложениями для навигации).

3.1.2 Требования к организации входных данных

Основную входную информацию программа получает из базы данных, имеющую следующие таблицы с указанными полями:

- Географическое местоположение
 - Долгота

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв.	Подп. и дата
Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

71

- Широта
 - Почтовый адрес
- Склады
 - Идентификатор местоположения
- Водители
 - Идентификатор
 - ФИО
 - Логин
 - Пароль
 - Идентификатор склада, к которому он привязан
- Заказы
 - Идентификатор местоположения
 - Дата
 - Описание
 - Статус
- Маршруты
 - Идентификатор
 - Статус
 - Дата
- Таблица, связывающая маршруты и заказы
 - Идентификатор маршрута
 - Идентификатор заказа
- Таблица, связывающая склады и заказы

Также программа должна иметь возможность получать всю необходимую входную информацию в текстовом виде через визуальный интерфейс системы, а именно:

- адреса пунктов производства товара;
- ФИО, имя пользователя и пароль для учетных записей водителей;
- адрес, дата и описание заказов.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

72

3.1.3 Требования к организации выходных данных

Программа должна предоставлять выходные данные пользователям посредством визуального интерфейса, а именно:

- списки складов, водителей, заказов для группы пользователей «Администратор»;
- построенные маршруты как в графическом виде (на карте), так и в текстовом виде для групп пользователей «Администратор» и «Водитель».

3.1.4 Требования к временным характеристикам

Программа должна работать круглосуточно и непрерывно.

3.2 Требования к надежности

3.2.1 Требования к обеспечению надежного (устойчивого) функционирования программы

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением совокупности организационно-технических мероприятий, перечень которых приведен ниже:

- 1) организацией бесперебойного питания технических средств;
- 2) регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

73

3.2.2 Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать времени, необходимого на перезагрузку операционной системы и запуск программы, при условии соблюдения условий эксплуатации технических и программных средств.

3.2.3 Отказы из-за некорректных действий оператора

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему административных привилегий.

3.3 Условия эксплуатации

3.3.1 Климатические условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

3.3.2 Требования к видам обслуживания

См. Требования к обеспечению надежного (устойчивого) функционирования программы.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

74

3.3.3 Требования к численности и квалификации персонала

Требования к численности и квалификации персонала не предъявляются.

3.4 Требования к составу и параметрам технических средств

В состав технических средств серверной части приложения должны входить:

- 1) сервер БД с оперативной памятью объемом не менее 3Гб и жестким диском объемом не менее 10 Гб;
- 2) сервер приложений с оперативной памятью объемом не менее 2Гб;
- 3) наличие подключения к локальной вычислительной сети и сети Интернет у сервера приложений и сервера БД.

В состав технических средств клиентской части приложения должно входить компьютерное устройство с оперативной памятью не менее 256Мб и доступом в сеть Интернет.

3.5 Требования к информационной и программной совместимости

3.5.1 Требования к информационным структурам и методам решения

Требования к информационным структурам (файлов) на входе и выходе, а также к методам решения не предъявляются.

3.5.2 Требования к исходным кодам и языкам программирования

Исходные коды программы должны быть реализованы на языке Java версии не ниже 8. В качестве интегрированной среды разработки программы должна быть использована среда NetBeans версии не ниже 8 или IntelliJ IDEA версии не ниже

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв.	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

75

2016. В качестве используемой СУБД должна быть использована MySQL версии не ниже 5, а в качестве сервера приложений – Apache Tomcat версии не ниже 7.

3.5.3 Требования к программным средствам, используемым программой

Требования к серверной части:

- базовая ОС – операционная система на базе Linux с версией ядра не ниже 2.6 / Windows 7 или выше / MacOS 10.9 или выше;
- система управления базами данных – MySQL версия не ниже 5;
- JRE/JDK версии 8.0 и выше;
- сервер приложений Apache Tomcat 7 или выше

Требования к клиентской части:

- базовая ОС – Windows XP/Vista/7/8/10; Linux; Android; iOS, MacOS;
- браузеры – Chrome Desktop 28 и выше, Safari Desktop 5 и выше, Firefox Desktop 25 и выше.

3.5.4 Требования к защите информации и программ

Пароли пользователей системы должны храниться в зашифрованном виде.

3.6 Специальные требования

Специальные требования к программе не предъявляются.

4 Требования к программной документации

4.1 Предварительный состав программной документации

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв.	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

76

Состав программной документации должен включать в себя:

- 1) техническое задание;
- 2) руководство программиста;
- 3) руководство пользователя;

5.2 Специальные требования к программной документации

Специальные требования к программной документации не предъявляются.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

77

ПРИЛОЖЕНИЕ Б
(обязательное)
РУКОВОДСТВО ПРОГРАММИСТА

1 Назначение и условия применения программы

1.1 Назначение программы

Программа предназначена для регистрации заказов и построения маршрута минимальной длины, проходящего по всем местоположениям, соответствующим заказам, ровно один раз.

1.2 Функции, выполняемые программой

Общие функции:

- 1) вход пользователей в систему / выход из системы.

Функции, доступные группе пользователей «Администратор»:

- 4) регистрация заказов на доставку товара с указанием даты и местоположения доставки;
- 5) регистрация и удаление водителей;
- 6) регистрация и удаление складов.

Функции, доступные группе пользователей «Водитель»:

- 7) построение маршрута минимальной длины, проходящего по всем местоположениям, соответствующим заказам, ровно один раз;

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист
78

1.3 Условия, необходимые для выполнения программы

1.3.1 Объем оперативной памяти

Для выполнения своих функций, программе достаточно:

- 256 Мб оперативной памяти на рабочей станции пользователя;
- 2048 Мб оперативной памяти на сервере приложений;
- 3 Гб оперативной памяти на сервере БД.

Однако, поскольку часть ОЗУ неизбежно занимает операционная система, то рекомендуется использовать программу на оборудовании, имеющем следующие параметры:

- более 512 Мб оперативной памяти на рабочей станции пользователя;
- более 3 Гб оперативной памяти на сервере приложений;
- более 4 Гб оперативной памяти на сервере БД.

1.3.2 Требования к составу периферийных устройств

Для работы необходимо наличие следующих периферийных устройств:

- выделенного объема дискового пространства не менее 10 Гб на сервере БД;
- наличие подключения к локальной вычислительной сети и сети Интернет у сервера приложений и сервера БД;
- наличие подключения к сети Интернет на рабочей станции пользователя.

1.3.3 Требования к параметрам периферийных устройств

Периферийные устройства, требуемые для работы программы, должны иметь следующие параметры:

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

79

- локальная вычислительная сеть должна иметь пропускную способность способностью не ниже 100Мб/сек;
- подключение к сети Интернет на рабочей станции должно обеспечивать пропускную способность не менее 1 Мб/сек.

1.3.4 Требования к программному обеспечению

Требования к серверной части:

- базовая ОС – операционная система на базе Linux с версией ядра не ниже 2.6 / Windows 7 или выше / MacOS 10.9 или выше;
- система управления базами данных – MySQL версия не ниже 5;
- JRE/JDK версии 8.0 и выше;
- сервер приложений Apache Tomcat 7 или выше

Требования к клиентской части:

- базовая ОС – Windows XP/Vista/7/8/10; Linux; Android; iOS, MacOS;
- браузеры – Chrome Desktop 28 и выше, Safari Desktop 5 и выше, Firefox Desktop 25 и выше.

1.3.5 Требования к персоналу (программисту)

Программист должен иметь минимум среднее техническое образование и обладать всеми необходимыми навыками для решения следующих задач:

- задача поддержания работоспособности программы;
- доработка программы по требованиям пользователей системы.

В минимальный набор знаний и навыков программиста должны входить следующий набор технологий:

- знание языков программирования: JAVA, JavaScript,
- знание основ работы с реляционными базами данных
- навыки работы со Spring Framework;

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата
Изм.	Лист	№ докум.	Подпись	Дата

- знание JPA или Hibernate

2 Характеристика программы

2.1 Описание основных характеристик программы

2.1.1 Режим работы программы

Программа должна работать круглосуточно и непрерывно.

2.1.2 Средства контроля правильности выполнения программы

Контроль правильности выполнения программы осуществляется через файлы логов программы и компонентов системы, обеспечивающих функционирование программного обеспечения на стороне сервера. В случае возникновения исключительных ситуаций, которые не могут быть обработаны средствами системы, информация о них выводится в файлы логов (расположение файлов логов задается в конфигурационных файлах программы и зависит от версии используемой операционной системы и настроек конкретного сервера).

2.2 Самовосстанавливаемость программы

Серверные компоненты программы имеют возможность работы в режиме, когда не требуется их ручное восстановление после сбоев или после перезагрузки компьютера. Такой режим функционирования системы обеспечивается конфигурацией операционной системы в целом, и всех ее компонентов в частности (автостарт сервисов после перезагрузки и/или сбоев).

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

81

3 Обращение к программе

3.1 Загрузка и запуск программы

Поскольку на сервере, содержащем код программы, настроен и протестирован автозапуск всех технологических компонент: веб-сервер, сервер БД, сервер приложений и др., то для запуска программы необходимо лишь включить сервер (виртуальную машину) и дождаться полного запуска операционной системы.

В случае, если требуется осуществить запуск компонент в ручном режиме, необходимо пользоваться следующими командами:

```
/opt/apache-tomcat-7.0.56/bin/startup.sh  
#запуск сервера приложений *  
service mysql start  
#запуск сервера БД *
```

* механизм запуска службы может отличаться в зависимости от версии операционной системы, версии приложения и настроек конкретного сервера.

3.2 Выполнение программы

Выполнение всех функций программы основано на шаблоне проектирования «Модель-Представление-Контроллер» при помощи Spring Web MVC. Описание данного паттерна приведено на рисунке 1.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

82

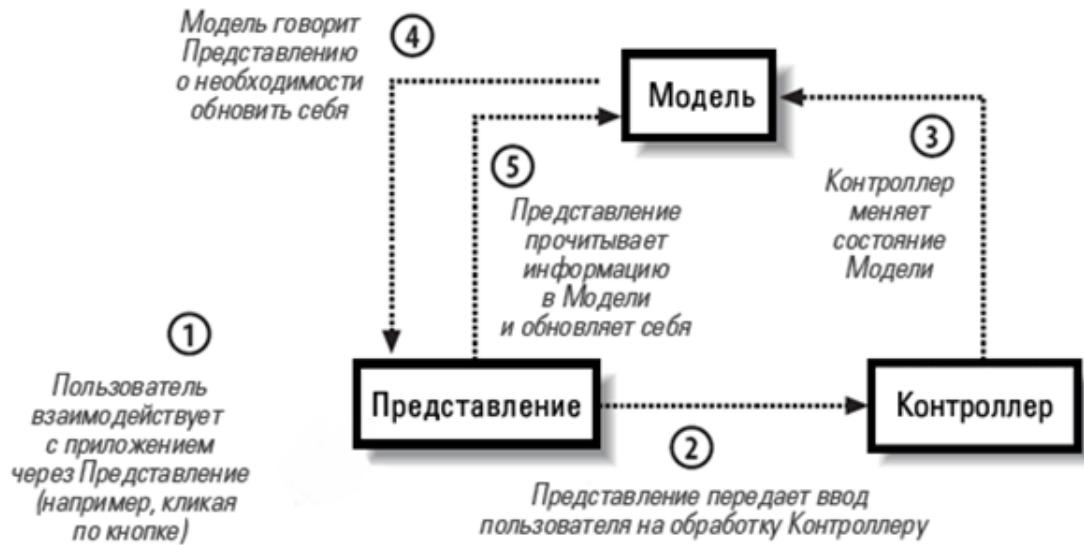


Рисунок 32. Схема работы приложения, использующего MVC.

3.2.1 Выполнение функции входа пользователей в систему / выхода из системы

- 1) Пользователь обращается на страницу ввода логина и пароля /login
- 2) Класс FilterChainProxy, встроенный в Spring Security, перехватывает обращение к данной странице и проверяет введенные логин/пароль.
- 3) Если данные введены неверно, то метод login(...) контроллера UserController возвратит страницу ввода логина и пароля с полученной ошибкой;
- 4) Если же данные введены верно, то метод loadUserByUsername класса UserDetailsServiceImpl по данным из БД возвращает новый объект класса org.springframework.security.core.userdetails.UserDetails.
- 5) Происходит перенаправление на страницу /welcome, которое обрабатывается в методе welcome() контроллера UserController, и возвращает представление welcome.jsp.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв. №	Подп. и дата

3.2.2 Выполнение функции регистрации заказов на доставку товара

- | | | | | |
|--------------|--------------|--------------|----------------|--------------|
| Инв. № подп. | Подп. и дата | Инв. № дубл. | Взам. № инв. № | Подп. и дата |
|--------------|--------------|--------------|----------------|--------------|
- 1) Администратор вводит данные о заказе: местоположение для доставки, дату, описание заказа, нажимает кнопку «Добавить»;
 - 2) Происходит отправка POST-запроса по адресу /admin/orders/add
 - 8) Метод addOrder() контроллера AdminController перехватывает посылку запроса на вышеуказанный адрес и вызывает метод addNewOrder(...) класса OrderDao, который выполняет следующее:
 - a. Сохраняет местоположение заказа при помощи метода addNewPlace(String address) класса PlaceDao, который:
 - i. Пытается геокодировать введенный адрес и получить долготу и широту местоположения в классе Geocoder;
 - ii. Создает новый объект класса Place с полученными от геокодера данными, если они корректны; иначе зануляет объект.
 - b. Если на предыдущем шаге создан ненулевой объект, то создается новый объект класса Order и сохраняется в БД;
 - c. Находится ближайший от местоположения заказа склад при помощи формулы Гаверсинуса, и к данному складу привязывается данный заказ.
 - 9) Происходит перенаправление на страницу /admin/orders, которое обрабатывается в методе orders() контроллера AdminController, и возвращает представление orders.jsp.

3.2.3 Выполнение функции регистрации и удаления водителей

Регистрация водителей:

- 1) Пользователь вводит данные и нажимает кнопку «Зарегистрировать»;
- 2) Происходит отправка POST-запроса по адресу /driver/registration;

Изм.	Лист	№ докум.	Подпись	Дата	1303.102002.001 ПЗ	Лист 84
------	------	----------	---------	------	--------------------	------------

- 3) Метод registration() контроллера UserController перехватывает посылку запроса на вышеуказанный адрес и вызывает метод validate(...) класса валидатора UserValidator, который проверяет корректность введенных данных;
- 4) Если валидатор найдет ошибки во введенных данных, то пользователю вернется та же самая страница с разъяснением ошибок;
- 5) Иначе – вызовется метод save(Driver) класса DriverService, который зашифрует введенный пароль и сохранит водителя в базу данных;
- 6) Пользователь будет перенаправлен на страницу /admin/drivers со списком всех водителей.

Удаление водителей:

- 1) Для удаления водителя необходимо в строке, соответствующей ему, нажать кнопку «Уволить», при этом на сервер отправится GET-запрос с ID данного водителя по адресу /admin/drivers/remove;
- 2) Метод removeDriver(id) контроллера AdminController перехватывает обращение по этому адресу и выполняет следующее:
- 3) Вызывает метод removeById(id) класса DriverDao, который удаляет из БД запись с id, соответствующую удаляемому водителю;
- 4) Перенаправляет на страницу /admin/drivers с обновленным списком всех водителей.

3.2.4 Выполнение функции добавления и удаления склада

Добавление склада:

- 1) Пользователь вводит адрес склада и нажимает кнопку «Добавить»;
- 2) Происходит отправка POST-запроса по адресу /admin/stocks/add;
- 3) Метод addStock() контроллера AdminController перехватывает посылку запроса на вышеуказанный адрес и вызывает метод addNewStock(address) класса StockDao, который:

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата
Изм.	Лист	№ докум.	Подпись	Дата

- a. Сохраняет местоположение заказа при помощи метода addNewPlace(String address) класса PlaceDao аналогично тому, как описано в п. 3) а. раздела 3.2.2;
- b. Если на предыдущем шаге создан ненулевой объект, то создается новый объект класса Stock и сохраняется в БД;
- 4) Затем пользователь будет перенаправлен на страницу /admin/stocks со списком всех складов.

Удаление склада:

- 1) Для удаления склада необходимо в строке, соответствующей ему, нажать кнопку «Удалить», при этом на сервер отправится GET-запрос с ID данного склада по адресу /admin/stocks/remove;
- 2) Метод removeStock(id) контроллера AdminController перехватывает обращение по этому адресу и выполняет следующее:
 - а. Вызывает метод removeById(id) класса StockDao, который удаляет из БД запись с id, соответствующую удаляемому водителю;
 - б. Перенаправляет на страницу /admin/stocks с обновленным списком всех складов.

3.2.5 Выполнение функции нахождения маршрута минимальной длины

- 1) Водитель переходит по адресу /driver/initroute, происходит отправка GET-запроса по данному адресу;
- 2) Метод initroute() контроллера DriverController перехватывает посылку запроса на вышеуказанный адрес, получая в качестве параметра имя пользователя, от которого произошел вызов метода, и выполняет следующие действия:
 - а. вызывает метод findUntakenOrdersByDateForDriver (Date, Driver) класса OrderDao, который находит заказы на заданную дату, которые прикреплены к складу, на котором работает данный

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв. №	Подп. и дата
Изм.	Лист	№ докум.	Подпись	Дата

- водитель;
- При помощи Yandex Maps API находятся расстояния между всеми местоположениями заказов и склада в виде двумерной матрицы;
 - Вызывается статический метод `getOrdersAsRoute()` класса `GAService`, который запускает генетический алгоритм нахождения кратчайшего пути и возвращает список заказов в порядке, соответствующем найденному пути;
 - Создается новый объект класса `Route` и при помощи метода `save(Route)` класса `RouteDao` записывается в БД
- 3) Затем пользователь будет перенаправлен на страницу `/driver/currentRoute`, на которой будет отображен маршрут на Яндекс.Карте, ссылка для переноса маршрута в мобильное приложение Яндекс.Навигатор, а также сам маршрут в текстовом виде.

3.3 Завершение работы программы

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв. №	Подп. и дата
--------------	--------------	--------------	----------------	--------------

Завершение работы серверной части программного обеспечения осуществляется остановкой сервера приложений (Apache Tomcat). Остановка со стороны сервера только приложения Базы данных недопустимо (т.к. непосредственно приложение продолжит свою работу, но не сможет корректно функционировать ввиду отсутствия доступа к массиву данных приложения). Завершение работы серверной части программного обеспечения приводит к невозможности доступа всех пользователей в разработанной АИС до момента восстановления полной работоспособности приложения.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв. №	Подп. и дата
--------------	--------------	--------------	----------------	--------------

Закрытие окна браузера пользователем приводит к завершению работы клиентской части приложения на компьютере конкретного пользователя. В таком случае серверная часть приложения остается функционировать в непрерывном режиме и продолжает обеспечивать доступ всем остальным пользователям к данным и функциям АИС в полном объеме.

Изм.	Лист	№ докум.	Подпись	Дата

4 Входные и выходные данные

4.1 Организация используемой входной информации

Основную входную информацию программа получает из базы данных, имеющую следующие таблицы с указанными полями:

- Географическое местоположение
 - Долгота
 - Широта
 - Почтовый адрес
- Склады
 - Идентификатор местоположения
- Водители
 - Идентификатор
 - ФИО
 - Логин
 - Пароль
 - Идентификатор склада, к которому он привязан
- Заказы
 - Идентификатор местоположения
 - Дата
 - Описание
 - Статус
- Маршруты
 - Идентификатор
 - Статус
 - Дата
- Таблица, связывающая маршруты и заказы
 - Идентификатор маршрута

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв. №	Подп. и дата
Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

88

- Идентификатор заказа
- Таблица, связывающая склады и заказы
 - Идентификатор склада
 - Идентификатор заказа

Также программа получает входную информацию в текстовом виде через визуальный интерфейс системы.

В качестве входных параметров используются:

- адреса пунктов производства товара;
- ФИО, имя пользователя и пароль для учетных записей водителей;
- адрес, дата и описание заказов.

4.2 Организация используемой выходной информации

Программа предоставляет выходные данные пользователям посредством визуального интерфейса.

Выходными параметрами являются:

- списки складов, водителей, заказов для группы пользователей «Администратор»;
- построенные маршруты как в графическом виде (на карте), так и в текстовом виде для групп пользователей «Администратор» и «Водитель».

5 Сообщения

5.1 Сообщение об ошибке

При возникновении ошибки (необработанного исключения) в программе, например, при ошибочной работе сервера БД, в логах системы будет следующее сообщение:

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

89

SEVERE: Servlet.service() for servlet [dispatcher] in context with path [/TSP] threw exception [Request processing failed; nested exception is java.lang.NullPointerException] with root cause
java.lang.NullPointerException
at
ais.spring.web.utils.DomainVOConverter.convertToFormVO(DomainVOConverter.java:60)
at
ais.spring.web.utils.DomainVOConverter.convertToVO(DomainVOConverter.java:35)
at
ais.spring.web.service.impl.FormServiceImpl.getId(FormServiceImpl.java:62)
at ais.spring.web.controller.BaseController.getForm(BaseController.java:79)
at sun.reflect.GeneratedMethodAccessor26.invoke(Unknown Source)

В тексте сообщения об ошибке присутствует строка «Exception», после которой идет текст ее описания.

При получении данного сообщения, необходимо по стеку вызова, приведенному в сообщении, найти в исходном коде строку, на которой возникла исключительная ситуация, проанализировать проблему и устранить ее изменением исходного кода программы либо данных, хранящихся в БД программы.

5.2 Сообщение о неуспешной авторизации

При неуспешной авторизации пользователя в логах системы появится сообщение следующего вида:

2017-06-01 00:42:04 - login failed

5.3 Сообщение об успешной авторизации

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата
--------------	--------------	--------------	--------------	--------------

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

1303.102002.001 ПЗ

Лист

90

При успешной авторизации пользователя в логах системы появится сообщение следующего вида:

2017-06-01 00:41:50 – “username” logged in

5.4 Сообщение о регистрации пользователя в системе

При успешной регистрации пользователя в логах системы появится сообщение следующего вида:

2017-06-01 00:49:45 - Driver “username” successfully registered

5.5 Сообщение о новом составленном маршруте

При успешной составлении нового маршрута для водителя пользователя в логах системы появится сообщение следующего вида:

2017-06-01 01:01:19 - New route created for “username” with id=1

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист
91

ПРИЛОЖЕНИЕ В
(обязательное)
РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

1 Назначение программы

1.1 Функциональное назначение программы

Программа предназначена для регистрации заказов и построения маршрута минимальной длины, проходящего по всем местоположениям, соответствующим заказам, ровно один раз.

1.2 Эксплуатационное назначение программы

Ограничений на эксплуатацию программы не вводится.

1.3 Состав функций

Общие функции:

- 1) вход пользователей в систему / выход из системы.

Функции, доступные группе пользователей «Администратор»:

- 2) регистрация заказов на доставку товара с указанием даты и местоположения доставки;
- 3) регистрация и удаление водителей;
- 4) регистрация и удаление складов.
- 5) создание маршрута для водителя

Функции, доступные группе пользователей «Водитель»:

- 6) просмотр списка всех прошлых маршрутов
- 7) просмотр маршрута на текущий день, если присутствуют заказы

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата
Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

92

2 Условия выполнения программы

2.1 Минимальный состав аппаратных средств

- 1) Оперативная память объемом не менее 256 Мб;
- 2) Наличие подключения к сети Интернет с пропускной способностью не менее 1Мб/сек;
- 3) Наличие устройств ввода (клавиатура, мышь, тачпад).

2.2 Минимальный состав программных средств

- 1) Операционная система - Windows XP/Vista/7/8/10; Linux; Android; iOS, MacOS;
- 2) Веб-браузер - Chrome Desktop 28 и выше, Safari Desktop 5 и выше, Firefox Desktop 25 и выше.

2.3 Требования к персоналу (пользователю)

Требований к персоналу (пользователю) не предъявляются.

3 Выполнение программы

3.1 Загрузка и запуск программы

Доступ к автоматизированной информационной системе осуществляется интерактивно через сеть Интернет посредством обычного web–браузера. При входе на сайт пользователь попадает на страницу ввода логина и пароля.

На сайте различаются следующие группы пользователей:

1. Водитель

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата	Лист	93
					1303.102002.001 ПЗ	

2. Администратор

3.2 Выполнение программы

Для входа в систему необходимо ввести логин и пароль, как показано на Рисунок 33.

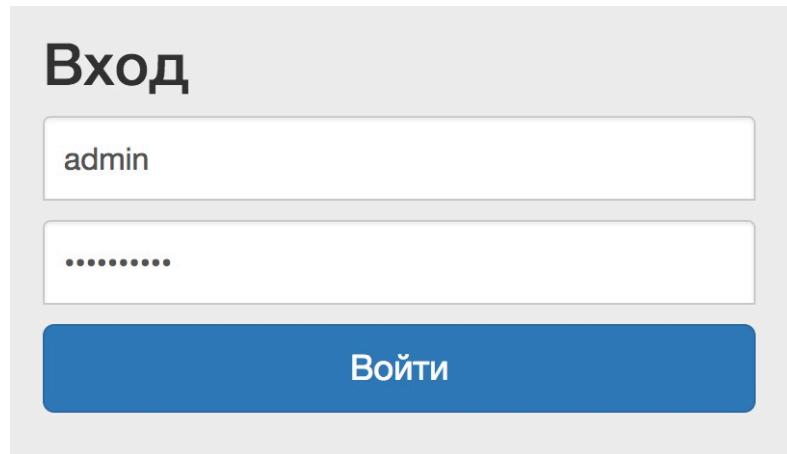


Рисунок 33. Вход в систему.

3.2.1 Функции, доступные группе пользователей «Администратор»

Меню администратора представлено на Рисунок 34.

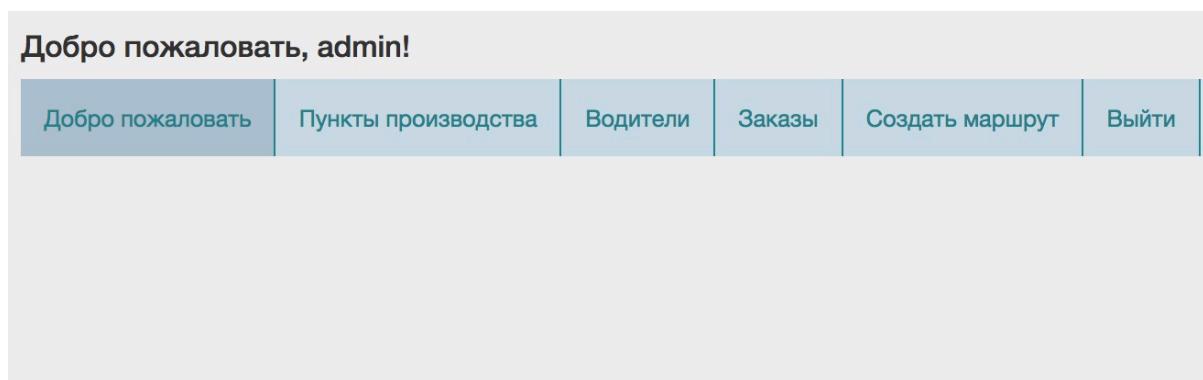


Рисунок 34. Меню администратора.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата
Изм.	Лист	№ докум.	Подпись	Дата

Для просмотра пункта производства, необходимо перейти к «Пункт производства» (Рисунок 35).

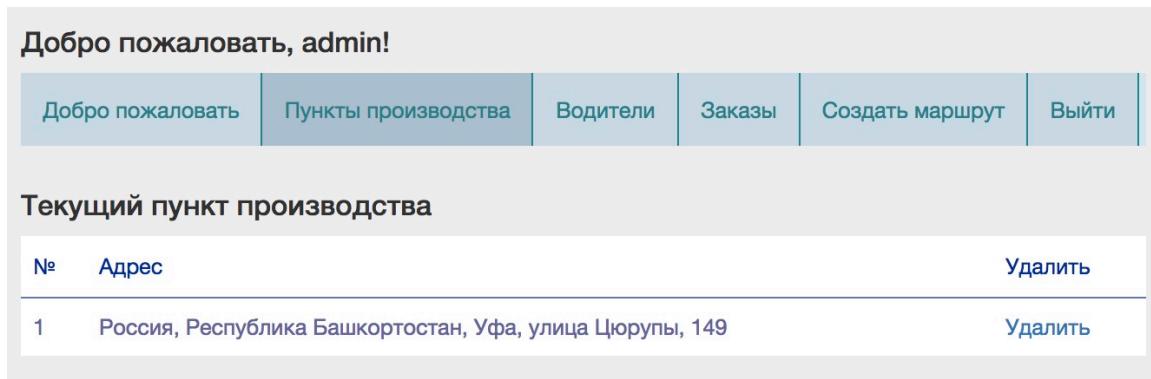


Рисунок 35. Пункт производства.

Для того, чтобы добавить пункт производства в систему, необходимо ввести полный адрес его местонахождения, и нажать кнопку «Добавить».

Для удаления склада из системы необходимо в строке, соответствующей данному складу, нажать на «Удалить».

Для просмотра списка водителей, необходимо перейти к пункту «Водители» (Рисунок 36).

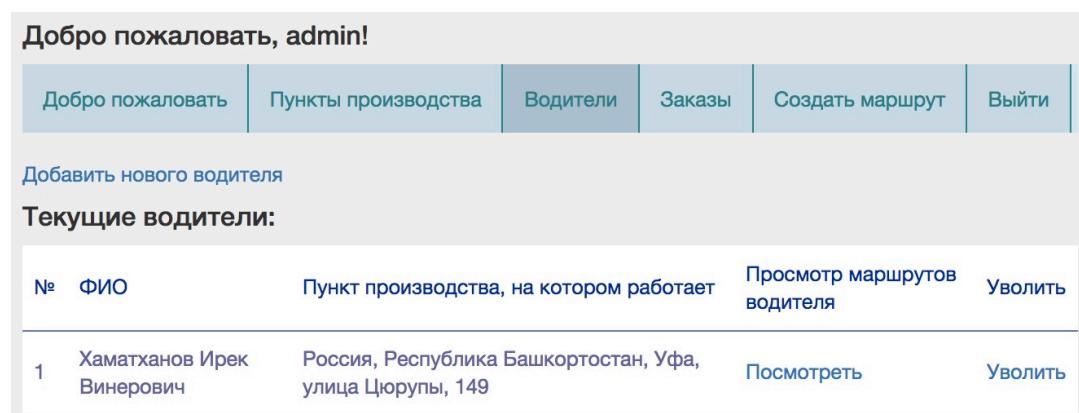


Рисунок 36. Список водителей.

Для того, чтобы добавить нового водителя в систему, необходимо нажать кнопку «Добавить водителя». Откроется форма регистрации водителя, в которой необходимо ввести ФИО, логин и пароль, а также выбрать склад, к которому привязан водитель, как показано на Рисунок 37.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв.	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата	1303.102002.001 ПЗ	Лист 95

Создание аккаунта

Хаматханов Ирек Винерович
irek
....
....
Россия, Республика Башкортостан, Уфа, Российская улица, 25
Создать

Рисунок 37. Регистрация водителя в системе.

Для завершения регистрации необходимо нажать кнопку «Создать».

Для удаления водителя из системы необходимо в строке, соответствующей данному водителю, нажать на «Уволить».

Для того, чтобы отметить какой-либо маршрут водителя выполненным, необходимо сначала нажать на «Посмотреть» в строке, соответствующей нужному водителю, а затем в списке маршрутов в строке, соответствующей не выполненному маршруту, нажать кнопку «Отметить выполненным» (Рисунок 38).

Добро пожаловать, admin!

Добро пожаловать	Склады	Водители	Заказы	Выйти
------------------	--------	----------	--------	-------

Маршруты linear:

№	Дата	Статус	Посмотреть маршрут	Отметить выполненным
1	2017-06-02	Не выполнен	Посмотреть	Отметить выполненным

Рисунок 38. Отметка маршрута как выполненного.

Инв. № подп.	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист
96

Для просмотра списка заказов, необходимо перейти к пункту «Заказы» (Рисунок 39).

Добро пожаловать, admin!

Добавить заказ

Полный адрес (город, улица, дом)

Дата доставки заказа: 02.06.2017

Описание заказа

Добавить заказ

Заказы :

№	Дата	Адрес	Статус	Описание	Отменить заказ
1	2017-06-02	Россия, Республика Башкортостан, Уфа, улица Карла Маркса, 12	Не выполнен	Огурцы 20кг	Отменить
2	2017-06-02	Россия, Республика Башкортостан, Уфа, улица Шота Руставели, 25/1	Не выполнен	Помидоры 40кг	Отменить

Рисунок 39. Список заказов.

Для того, чтобы добавить новый заказ в систему, необходимо ввести полный адрес для его доставки, дату доставки и описание, после чего нажать кнопку «Добавить заказ».

Для отмены заказа необходимо в строке, соответствующей данному заказу, нажать на «Отменить».

Для того, чтобы создать маршрут и назначить его водителю, необходимо перейти в пункт меню «Создать маршрут», затем выбрать водителя из списка, и выбрать дату маршрута. При нажатии на кнопку «Создать маршрут» все заказы на выбранную дату будут объединены в маршрут. (Рисунок 40).

Добро пожаловать, admin!

Создать маршрут

Выберите водителя: Хаматханов Ирек Винерович Выберите дату: 18.06.2017

Создать маршрут

Рисунок 40 - Создание маршрута

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. № инв. №	Подп. и дата
--------------	--------------	--------------	----------------	--------------

Изм.	Лист	№ докум.	Подпись	Дата	1303.102002.001 ПЗ	Лист 97
------	------	----------	---------	------	--------------------	------------

3.2.2 Функции, доступные группе пользователей «Водитель»

Меню водителя представлено на рисунке 41.



Рисунок 41. Меню водителя.

При нажатии на пункт «Текущий маршрут» может отображаться следующее:

- Если водитель не имеет незавершенных маршрутов, то будет выведено соответствующее сообщение (Рисунок 42).

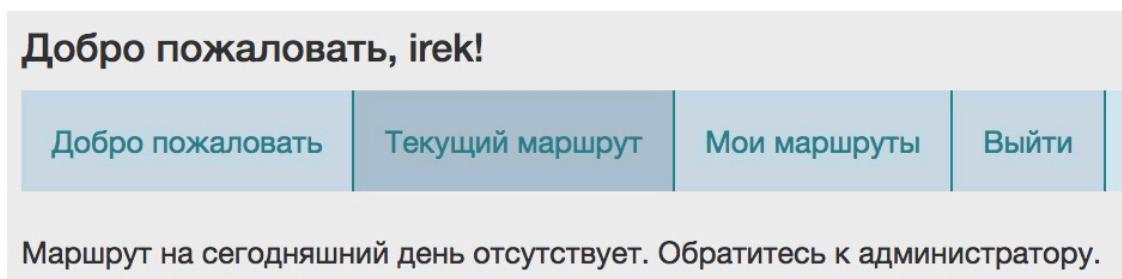


Рисунок 42. Предложение взяться за доставку заказов.

- Если же водитель имеет незавершенный маршрут, то он будет отображен в виде карты (при помощи Яндекс.Карты) и в текстовом виде (Рисунок 43).

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата
Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

98

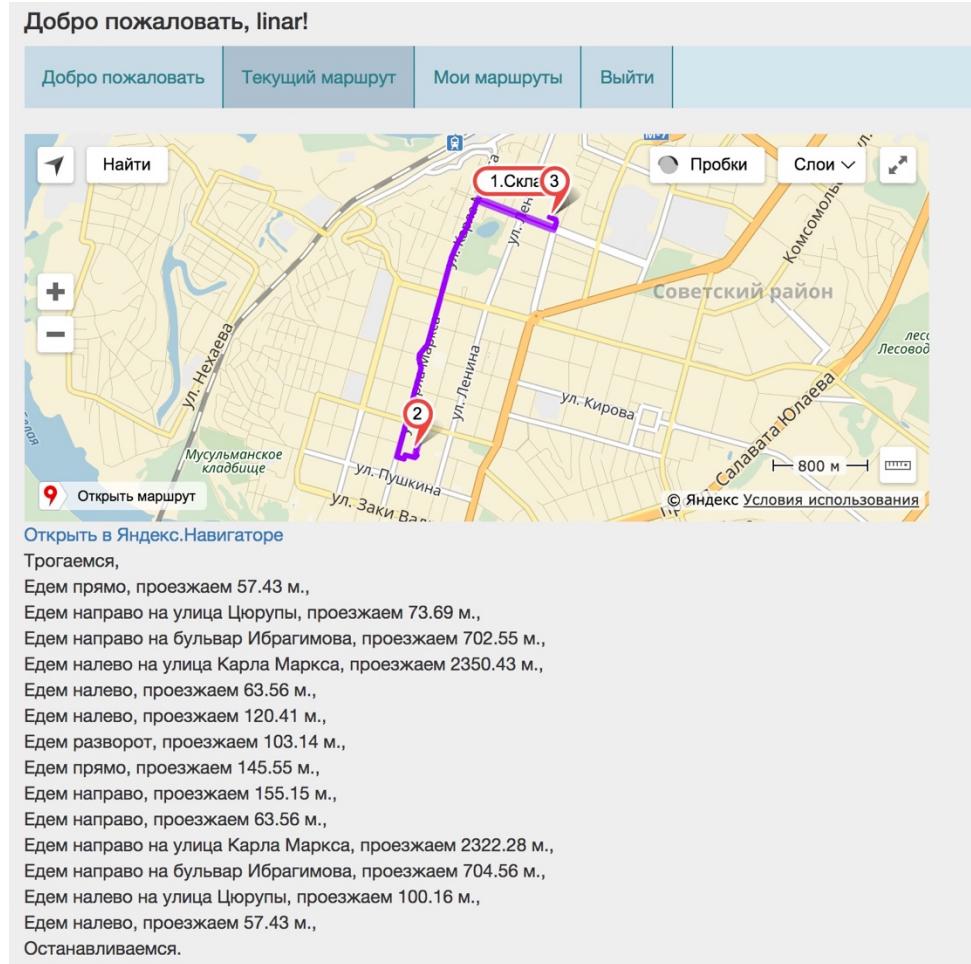


Рисунок 43. Отображение маршрута.

Для просмотра списка маршрутов, необходимо перейти к пункту «Мои маршруты» (Рисунок 10).

Маршруты :			
№	Дата	Статус	Посмотреть маршрут
1	2017-06-02	Не выполнен	Посмотреть

Рисунок 44. Список маршрутов.

При нажатии на «Посмотреть» будет показан выбранный маршрут аналогично тому, как показано на рисунке 9.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

1303.102002.001 ПЗ

Лист

99

3.3 Завершение работы программы

Закрытие окна браузера пользователем приводит к завершению работы клиентской части приложения на компьютере конкретного пользователя.

4 Сообщения пользователю

4.1 Сообщение при регистрации

При регистрации водителя в системе могут быть получены сообщения об ограничениях на длину вводимых логина и пароля, а также о несовпадении паролей (Рисунок 45).

Создание аккаунта

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата
--------------	--------------	--------------	--------------	--------------

Полное имя

Логин

Данное поле необходимо заполнить.
Имя пользователя должно быть не короче 4 и не длиннее 32 символов.

Пароль

Данное поле необходимо заполнить.
Пароль должен быть длиннее 4 символов.

•

Пароли не совпадают.

Россия, Республика Башкортостан, Уфа, улица Цюрупы, 149

Создать

Рисунок 45. Сообщения при регистрации.

Изм.	Лист	№ докум.	Подпись	Дата	Лист	100
					1303.102002.001 ПЗ	

4.2 Сообщение при авторизации

При вводе некорректной пары логин/пароль пользователь получает предупреждение (Рисунок 46).

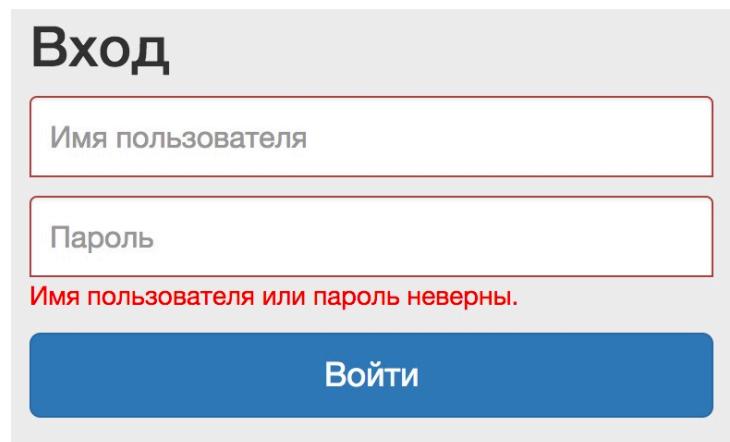


Рисунок 46. Сообщение при авторизации.

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист
101

ПРИЛОЖЕНИЕ Г
ОЦЕНКА КАЧЕСТВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Таблица 5 - Оценочные элементы фактора "Надежность ПС"

Код элемент а	Наименование	Метод оценки	Оцен- ка
Средства восстановления при ошибках на входе			
H0101	Наличие требований к программе по устойчивости функционирования при наличии ошибок во входных данных	Экспертный	1
H0102	Возможность обработки ошибочных ситуаций	Экспертный	1
H0103	Полнота обработки ошибочных ситуаций	Экспертный	0,8
H0104	Наличие тестов для проверки допустимых значений входных данных	Экспертный	0
H0105	Наличие системы контроля полноты входных данных	Экспертный	1
H0106	Наличие средств контроля корректности входных данных	Экспертный	1
H0107	Наличие средств контроля непротиворечивости входных данных	Экспертный	1
H0108	Наличие проверки параметров и адресов по диапазону их значений	Экспертный	1
H0109	Наличие обработки граничных результатов	Экспертный	0,5
H0110	Наличие обработки неопределенностей (деление на 0, квадратный корень из отрицательного числа и т. д.)	Экспертный	1
			0,83
Средства восстановления при сбоях оборудования			
H0201	Наличие требований к программе по восстановлению процесса выполнения в случае сбоя операционной системы, процессора, внешних устройств	Экспертный	0
H0202	Наличие требований к программе по восстановлению результатов при отказах процессора, ОС	Экспертный	0
H0203	Наличие средств восстановления процесса в случае сбоев оборудования	Экспертный	0

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

H0204	Наличие возможности разделения по времени выполнения отдельных функций программ	Экспертный	1
H0205	Наличие возможности повторного старта с точки останова	Экспертный	1
			0,4
Реализация управления средствами восстановления			
H0301	Наличие централизованного управления процессами, конкурирующими из-за ресурсов	Экспертный	1
H0302	Наличие возможности автоматически обходить ошибочные ситуации в процессе вычисления	Экспертный	1
H0303	Наличие средств, обеспечивающих завершение процесса решения в случае помех	Экспертный	1
H0304	Наличие средств, обеспечивающих выполнение программы в сокращенном объеме в случае ошибок или помех	Экспертный	1
H0305	Показатель устойчивости к искажающим воздействиям	Расчетный	0,9
			0,98

Функционирование в заданных режимах

H0401	Вероятность безотказной работы	То же	0,9
Обеспечение обработки заданного объема информации			
H0501	Оценка по среднему времени восстановления	»	1
H0502	Оценка по продолжительности преобразования входного набора данных в выходной	»	1
			1

Таблица 6 - Оценочные элементы фактора «Сопровождаемость ПС»

Код элемента	Наименование	Метод оценки	Оценка
Простота архитектуры проекта			
C0101	Наличие модульной схемы программы	Экспертный	1
C0102	Оценка программы по числу уникальных модулей	»	1
			1
Сложность архитектуры проекта			

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата
--------------	--------------	--------------	--------------	--------------

C0201	Наличие ограничений на размеры модуля	»	0
Межмодульные связи			
C030	Наличие требований к независимости модулей программы от типов и форматов выходных данных	»	0
C0301	Наличие проверки корректности передаваемых данных	»	1
C0302	Оценка простоты программы по числу точек входа и выхода	Расчетный	0,0036
C0303	Осуществляется ли передача результатов работы модуля через вызывающий его модуль	Экспертный	1
C0304	Осуществляется ли контроль за правильностью данных, поступающих в вызывающий модуль от вызываемого	»	1
			0,6
Экспертиза принятой системы идентификации			
C0601	Использование при построении программ метода структурного программирования	»	1
C0602	Соблюдение принципа разработки программы сверху вниз	»	1
C0603	Оценка программы по числу циклов с одним входом и одним выходом	»	1
C0604	Оценка программы по числу циклов	»	1
			1
Комментарии логики программ проекта			
C0801	Наличие комментариев ко всем машинозависимым частям программы	»	1
C0802	Наличие комментариев к машинозависимым операторам программы	»	0,2
C0803	Наличие комментариев в точках входа и выхода программы	»	0,5
			0,57
Оформление текста программ			
C0901	Соответствие комментариев принятым соглашениям	»	1
C0902	Наличие комментариев-заголовков программы с указанием ее структурных и функциональных характеристик	»	1
C0903	Оценка ясности и точности описания последовательности функционирования всех элементов программы	»	1

Изм.	Лист	№ докум.	Подпись	Дата	Лист 104

			1
Простота кодирования			
C1001	Используется ли язык высокого уровня	»	1
C1002	Оценка простоты программы по числу переходов по условию	Расчетный	0,91
			0,95

Таблица 7 - Оценочные элементы фактора "Удобство применения ПС"

Код элемента	Наименование	Метод оценки	Оценка
Освоение работы ПС			
У0101	Возможность освоения программных средств по документации	Экспертный	1
У0102	Возможность освоения ПС на контрольном примере при помощи ЭВМ	Экспертный	1
У0103	Возможность поэтапного освоения ПС	»	1
			1
Документация для освоения			
У0201	Полнота и понятность документации для освоения	»	1
У0202	Точность документации для освоения	»	1
У0203	Техническое исполнение документации	»	1
			1
Полнота пользовательской документации			
У0301	Наличие краткой аннотации	»	1
У0302	Наличие описания решаемых задач	»	1
У0303	Наличие описания структуры функций ПС	»	1
У0304	Наличие описания основных функций ПС	»	1
У0306	Наличие описания частных функций	»	0,8
У0307	Наличие описания алгоритмов	»	1
У0308	Наличие описания межмодульных интерфейсов	»	1
У0309	Наличие описания пользовательских интерфейсов	»	1
У0310	Наличие описания входных и выходных данных	»	1
У0311	Наличие описания диагностических сообщений	»	1
У0312	Наличие описания основных характеристик ПС	»	1

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата	Лист 105

У0314	Наличие описания программной среды функционирования ПС	»	1
У0315	Достаточность документации для ввода ПС в эксплуатацию	»	1
У0316	Наличие информации технологии переноса для мобильных программ	»	1
			0,99

Точность пользовательской документации

У0401	Соответствие оглавления содержанию документации	»	1
У0402	Оценка оформления документации	»	1
У0403	Грамматическая правильность изложения документации	»	1
У0404	Отсутствие противоречий	»	1
У0405	Отсутствие неправильных ссылок	»	1
У0406	Ясность формулировок и описаний	»	1
У0407	Отсутствие неоднозначных формулировок и описаний	»	1
У0408	Правильность использования терминов	»	1
У0409	Краткость, отсутствие лишней детализации	»	1
У0410	Единство формулировок	»	1
У0411	Единство обозначений	»	1
У0412	Отсутствие ненужных повторений	»	1
У0413	Наличие нужных объяснений	»	1
			1

Понятность пользовательской документации

У0501	Оценка стиля изложения	»	1
У0502	Дидактическая разделенность	»	1
У0503	Формальная разделенность	»	1
У0504	Ясность логической структуры	»	1
У0505	Соблюдение стандартов и правил изложения в документации	»	1
У0506	Оценка по числу ссылок вперед в тексте документов	»	1
			1

Техническое исполнение пользовательской документации

У0601	Наличие оглавления	»	1
У0602	Наличие предметного указателя	»	0
У0603	Наличие перекрестных ссылок	»	1
У0604	Наличие всех требуемых разделов	»	1

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата
--------------	--------------	--------------	--------------	--------------

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

У0605	Соблюдение непрерывности нумерации страниц документов	»	1
У0606	Отсутствие незаконченных разделов абзацев, предложений	»	1
У0607	Наличие всех рисунков, чертежей, формул, таблиц	»	1
У0608	Наличие всех строк и примечаний	»	1
У0609	Логический порядок частей внутри главы	»	1
			0,89

Прослеживание вариантов пользовательской документации

У0701	Наличие полного перечня документации	»	1
-------	--------------------------------------	---	---

Эксплуатация

У0801	Уровень языка общения пользователя с программой	»	1
У0802	Легкость и быстрота загрузки и запуска программы	»	1
У0803	Легкость и быстрота завершения работы программы	»	1
У0804	Возможность распечатки содержимого программы	»	1
У0805	Возможность приостанова и повторного запуска работы без потерь информации	»	0
			0,8

Управление меню

У0901	Соответствие меню требованиям пользователя	»	1
У0902	Возможность прямого перехода вверх и вниз по многоуровневому меню (пропуск уровней)	»	1
			1

Функция Help

У1001	Возможность управления подробностью получаемых выходных данных	»	1
У1002	Достаточность полученной информации для продолжения работы	»	1
			1

Управление данными

У1101	Обеспечение удобства ввода данных	»	1
У1102	Легкость восприятия	»	1
			1

Рабочие процедуры

Инв. № подп. Подп. и дата
 Изв. № дубл. Взам. изв. №
 Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

1303.102002.001 ПЗ

Лист
107

У1201	Обеспечение программой выполнения предусмотренных рабочих процедур	»	1
У1202	Достаточность информации, выдаваемой программой для составления дополнительных процедур	»	1
			1

Таблица 8 - Оценочные элементы фактора "Эффективность ПС"

Код элемента	Наименование	Метод оценки	Оценка
Уровень автоматизации			
Э0101	Проблемно-ориентированные функции	Экспертный	1
Э0102	Машинно-ориентированные функции	Экспертный	1
Э0103	Функции ведения и управления	Экспертный	1
Э0104	Функции ввода/вывода	Экспертный	1
Э0105	Функции защиты и проверки данных	Экспертный	1
Э0106	Функции защиты от несанкционированного доступа	Экспертный	1
Э0107	Функции контроля доступа	Экспертный	1
Э0108	Функции защиты от внесения изменений	Экспертный	1
Э0109	Наличие соответствующих границ функциональных областей	Экспертный	1
Э0110	Число знаков после запятой в результатах вычислений	Экспертный	1
			1
Временная эффективность			
Э0201	Время выполнения программ	»	1
Э0202	Время реакции и ответов	»	1
Э0203	Время подготовки	»	1
Э0205	Затраты времени на защиту данных	»	1
Э0206	Время компиляции	»	1
			1
Ресурсоемкость			
Э0301	Требуемый объем внутренней памяти	»	0,8
Э0302	Требуемый объем внешней памяти	»	1
Э0303	Требуемые периферийные устройства	»	1
Э0304	Требуемое базовое программное обеспечение	»	1
			0,95

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

108

Таблица 9 - Оценочные элементы фактора "Универсальность ПС"

Код элемента	Наименование	Метод оценки	Оценка
Зависимость от используемого комплекса технических средств			
Г0701	Оценка зависимости программ от ёмкости оперативной памяти ЭВМ	Экспертный	1
Г0702	Оценка зависимости временных характеристик программы от скорости вычислений ЭВМ	Экспертный	1
Г0703	Оценка зависимости функционирования программы от числа внешних запоминающих устройств и их общей ёмкости	Экспертный	0
Г0704	Оценка зависимости функционирования программы от специальных устройств ввода-вывода	Экспертный	0
			0,5
Зависимость от базового программного обеспечения			
Г0801	Применение специальных языков программирования	»	1
Г0802	Оценка зависимости программы от программ операционной системы	»	0
Г0803	Зависимость от других программных средств	»	1
			0,67
Изоляция немобильности			
Г0901	Оценка локализации непереносимой части программы	»	1
Простота кодирования			
Г1001	Оценка использования отрицательных или булевых выражений	»	1
Г1002	Оценка программы по использованию условных переходов	»	1
Г1003	Оценка программы по использованию безусловных переходов	»	1
Г1004	Оформление процедур входа и выхода из циклов	»	1
Г1005	Ограничения на модификацию переменной индексации в цикле	»	1
Г1007	Оценка программы по использованию локальных переменных	»	1

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Г1006	Оценка модулей по направлению потока управления	»	1
			1

Число комментариев

Г1101	Оценка программы по числу комментариев	»	0,5
-------	--	---	-----

Качество комментариев

Г1201	Наличие заголовка в программе	»	1
Г1202	Комментарии к точкам ветвлений	»	0
Г1203	Комментарии к машинозависимым частям программы	»	1
Г1204	Комментарии к машинозависимым операторам программы	»	0
Г1205	Комментарии к операторам объявления переменных	»	1
Г1206	Оценка семантики операторов	»	1
Г1207	Наличие соглашений по форме представления комментариев	»	1
Г1208	Наличие общих комментариев к программам	»	1
			0,8

Использование описательных средств языка

Г1301	Использование языков высокого уровня	»	1
Г1302	Семантика имен используемых переменных	»	1
Г1303	Использование отступов, сдвигов и пропусков при формировании текста	»	1
Г1304	Размещение операторов по строкам	»	1
			1

Независимость модулей

Г1401	Передача информации для управления по параметрам	»	1
Г1402	Наличие передачи результатов работы между модулями	»	1
Г1403	Наличие проверки правильности данных, получаемых модулями от вызываемого модуля	»	1
Г1404	Использование общих областей памяти	»	1
Г1405	Параметрическая передача входных данных	»	1
			1

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата
--------------	--------------	--------------	--------------	--------------

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

1303.102002.001 ПЗ

Лист

110

Таблица 10 - Оценочные элементы фактора "Корректность ПС"

Код элемента	Наименование	Метод оценки	Оценка
Требования, предъявляемые к полноте документации разработчика			
K0101	Наличие всех необходимых документов для понимания и использования ПС	Экспертный	1
K0102	Наличие описания и схемы иерархии модулей программы	»	1
K0103	Наличие описания основных функций	»	1
K0104	Наличие описания частных функций	»	0
K0105	Наличие описания данных	»	1
K0106	Наличие описания алгоритмов	»	1
K0107	Наличие описания интерфейсов между модулями	»	1
K0108	Наличие описания интерфейсов с пользователями	»	1
K0109	Наличие описания используемых числовых методов	»	1
K0110	Указаны ли все численные методы	»	0
K0111	Наличие описания всех параметров	»	0
K0112	Наличие описания методов настройки системы	»	1
K0113	Наличие описания всех диагностических сообщений	»	0
K0114	Наличие описания способов проверки работоспособности программы	»	1
			0,8
Полнота программной документации			
K0201	Реализация всех исходных модулей	»	1
K0202	Реализация всех основных функций	»	1
K0203	Реализация всех частных функций	»	1
K0204	Реализация всех алгоритмов	»	1
K0205	Реализация всех взаимосвязей в системе	»	1
K0206	Реализация всех интерфейсов между модулями	»	1
K0207	Реализация возможности настройки системы	»	1
K0208	Реализация диагностики всех граничных и аварийных ситуаций	»	1
K0209	Наличие определения всех данных (переменные, индексы, массивы и проч.)	»	1
K0210	Наличие интерфейсов с пользователем	»	1

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата
--------------	--------------	--------------	--------------	--------------

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

1303.102002.001 ПЗ

Лист

111

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата	1							
					Непротиворечивость документации разработчика							
					K0301	Отсутствие противоречий в описании частных функций	»	1				
					K0302	Отсутствие противоречий в описании основных функций в разных документах	»	1				
					K0303	Отсутствие противоречий в описании алгоритмов	»	1				
					K0304	Отсутствие противоречий в описании взаимосвязей в системе	»	1				
					K0305	Отсутствие противоречий в описании интерфейсов между модулями	»	1				
					K0306	Отсутствие противоречий в описании интерфейсов с пользователем	»	1				
					K0307	Отсутствие противоречий в описании настройки системы	»	1				
					K0309	Отсутствие противоречий в описании иерархической структуры сообщений	»	1				
					K0310	Отсутствие противоречий в описании диагностических сообщений	»	1				
					K0311	Отсутствие противоречий в описании данных	»	1				
					1							
Непротиворечивость программы												
Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата	K0401	Отсутствие противоречий в выполнении основных функций	»	1				
					K0402	Отсутствие противоречий в выполнении частных функций	»	1				
					K0403	Отсутствие противоречий в выполнении алгоритмов	»	1				
					K0404	Правильность взаимосвязей	»	1				
					K0405	Правильность реализации интерфейса между модулями	»	1				
					K0406	Правильность реализации интерфейса с пользователем	»	1				
					K0407	Отсутствие противоречий в настройке системы	»	1				
					K0408	Отсутствие противоречий в диагностике системы	»	1				
					K0409	Отсутствие противоречий в общих переменных	»	1				
					1							
Единообразие интерфейсов между модулями и пользователями												
Изм.	Лист	№ докум.	Подпись	Дата	1303.102002.001 ПЗ							
					Лист 112							

K0501	Единообразие способов вызова модулей	»	1
K0502	Единообразие процедур возврата управления из модулей	»	1
K0503	Единообразие способов сохранения информации для возврата	»	1
K0504	Единообразие способов восстановления информации для возврата	»	0
K0505	Единообразие организации списков передаваемых параметров	»	1
			0,8

Единообразие кодирования и определения переменных

K0601	Единообразие наименования каждой переменной и константы	»	1
K0602	Все ли одинаковые константы встречаются во всех программах под одинаковыми именами	»	1
K0603	Единообразие определения внешних данных во всех программах	»	1
K0604	Используются ли разные идентификаторы для разных переменных	»	1
K0605	Все ли общие переменные объявлены как общие переменные	»	1
K0606	Наличие определений одинаковых атрибутов	»	1
			1

Соответствие документации стандартам

K0701	Комплектность документации в соответствии со стандартами	»	1
K0702	Правильное оформление частей документов	»	1
K0703	Правильное оформление титульных и заглавных листов документов	»	1
K0704	Наличие в документах всех разделов в соответствии со стандартами	»	1
K0705	Полнота содержания разделов в соответствии со стандартами	»	1
K0706	Деление документов на структурные элементы: разделы, подразделы, пункты, подпункты	»	1
			1

Соответствие ПС стандартам программирования

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист
113

K0801	Соответствие организации и вычислительного процесса эксплуатационной документации	»	1
K0802	Правильность заданий на выполнение программы, правильность написания управляющих и операторов (отсутствие ошибок)	»	1
K0803	Отсутствие ошибок в описании действий пользователя	»	1
K0804	Отсутствие ошибок в описании запуска	»	1
K0805	Отсутствие ошибок в описании генерации	»	1
K0806	Отсутствие ошибок в описании настройки	»	1
			1

Полнота тестирования проекта

K1001	Наличие требований к тестированию программ	»	0
K1002	Достаточность требований к тестированию программ	»	1
K1003	Отношение числа модулей, отработавших в процессе тестирования и отладки (Q_t^M) к общему числу модулей (Q_o^M)	Расчетный	1
K1004	Отношение числа логических блоков, отработавших в процессе тестирования и отладки (Q_t^B), к общему числу логических блоков в программе (Q_o^B)	То же	0,88
			0,72

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

114

ПРИЛОЖЕНИЕ Д
РЕЗУЛЬТАТЫ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА
(ВТОРОЙ ЭТАП)

Таблица 11 - Результаты второго этапа вычислительного эксперимента

Инв. № подп.		Подп. и дата		Инв. № дубл.		Взам. инв. №		Подп. и дата		A
										2
	2		2		3		3		2	2
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	2
1	1	1	1	1	1	1	1	1	1	2
1	1	1	1	1	1	1	1	1	1	3
1	1	1	1	1	1	1	1	1	1	3
1	1	1	1	1	1	1	1	1	1	4
1	1	1	1	1	1	1	1	1	1	4
1	1	1	1	1	1	1	1	1	1	5
1	1	1	1	1	1	1	1	1	1	5
1	1	1	1	1	1	1	1	1	1	6
1	1	1	1	1	1	1	1	1	1	7
9	1	9	9	9	8	8	9	9	9	7
6	7	6	6	6	6	6	7	7	7	8
8	7	7	6	7	6	7	7	7	7	8
7	7	6	6	6	6	6	7	7	7	9
8	8	7	7	7	7	7	9	9	8	9
8	8	7	6	6	6	8	8	8	8	1

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист

115

Инв. № подп.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

3	3	3	2	2	2	3	3	3	2
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
9	9	9	9	9	9	9	9	9	8
9	9	9	8	1	1	9	9	9	1
8	9	9	9	9	9	9	9	8	9
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	9	1	1	1	1	8	9	9	1
6	6	7	7	6	6	6	6	6	5
7	7	7	8	7	8	6	7	7	7
7	6	7	8	8	8	7	7	6	8
7	6	7	8	8	8	7	6	7	8

Изм.	Лист	№ докум.	Подпись	Дата

1303.102002.001 ПЗ

Лист
116