Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Инженерно-экономический факультет

Кафедра экономической информатики

Дисциплина: Системный анализ и проектирование информационных систем

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ПРИНЯТИЯ

РЕШЕНИЙ МЕТОДОМ РАНГА

(НА ПРИМЕРЕ ЛОГИСТИЧЕСКОЙ КОМПАНИИ)

Студент: гр. 672303 Коленда В. А.

Руководитель: Хомяков П. В.

Минск 2018

**СОДЕРЖАНИЕ**

Введение 3

1 Постановка задачи 5

2 Описание предметной области 6

3 Построение базовой аналитической модели 10

4 Моделирование системы 11

4.1 Диаграмма вариантов использования 11

4.2 Моделирование базы данных 11

5 Алгоритм работы системы 13

5.1 Блок-схема работы программы 13

5.2 Блок-схема алгоритма выбора наиболее подходящего варианта транспортировки методом предпочтения 13

6 Обоснование выбора программных средств 15

7 Программная реализация 16

8 Руководство пользователя 18

9 Контрольный пример 22

Заключение 24

Список использованных источников 25

Приложение А Модели системы 26

Приложение Б Блок-схема работы программы 31

Приложение В Листинг кода программы 32

# **ВВЕДЕНИЕ**

Многие зарубежные компании в течение уже длительного периода времени нанимают для оказания услуг по выполнению непрофильных для них функций сторонние организации. Подобная схема носит название "аутсорсинг". Она означает привлечение на возмездной основе третьей стороны с целью выполнения задач, которые стоят перед компанией. Аутсорсинг помогает бизнесу быть более гибким, что позволяет ему получать хорошую прибыль.

На сегодняшний день существуют организации, которые предлагают выполнение широкого спектра операций, связанных с хранением, приобретением и перевозкой товаров. Среди них:

1. Складская логистика, производящая обработку, хранение и дистрибуцию грузов;
2. Логистика перевозок, призванная удовлетворить потребности компаний, у которых отсутствует собственный парк машин, в перевозке грузов;
3. Таможенная логистика, представляющая собой помощь в сертификации и таможенной очистке грузов.

Современная логистическая компания - это организация, находящаяся на 3PL-уровне. То есть она придерживается схемы, в которой выступает третьей стороной в сделках между покупателями и поставщиками, предоставляя полнейший комплекс сервисов, позволяющих обработать груз.

Логистика является наукой о контроле, управлении и планировании перемещения ресурсов и товаров. Одной из основных целей данного направления являются оптимизация и рационализация процесса перемещения грузов, а также производства и сбыта готовой продукции.

Логистическая компания – это надежный помощник бизнеса, который позволит предприятию совершенно легально сократить затраты на производство товаров благодаря оптимизации материальных потоков. Это происходит благодаря грамотному подбору транспорта для перевозки того или иного груза и выбору оптимальных маршрутов, которые наиболее эффективны с точки зрения сроков и тарифов, а также в связи со снижением складских и таможенных издержек.

Выбор вида транспорта решается во взаимной связи с такими задачами логистики как: создание и поддержание оптимального уровня запасов на складе; выбор вида транспортной тары; составление маршрутов движения транспортных средств и др.

На выбор транспортных средств будут влиять такие условия, как характер груза (вес, объем, консистенция), количество отправляемых партий, срочность доставки груза заказчику, местонахождение пунктов отправки и назначения с учетом погодных и климатических условий, ценность груза, близость расположения точек отправки и доставки к транспортным коммуникациям и др.

Основой при выборе вида транспорта, оптимального для конкретной перевозки, служит информация о характерных особенностях различных видов транспорта (автомобильный, железнодорожный, морской, внутренний водный, воздушный и трубопроводный).

Каждый вид транспорта имеет свои преимущества и недостатки, зная которые можно оптимизировать расходы и скорость перевозки грузов. Исходя из типа и параметров груза, подбирается наиболее подходящий транспорт.

Так как подбор вида транспорта для перевозки – достаточно сложное решение, от которого зависит как сохранность груза, так и прибыль самой компании, то необходимо создание такого программного приложения, которое позволило бы облегчить это процесс и в то же время обеспечить его правильность.

Исходя из этого, *целью* разработки данного программного приложения является оптимизация части работы логистической компании. Таким образом, в результате должно получиться функционирующее приложение, имеющее возможность взять на себя роль в принятии решения о выборе транспортных средств для перевозки грузов.

Достижение поставленной цели обеспечивается решением следующих *задач*:

1. Исследовать основные аспекты деятельности логистической компании.
2. Выделить аспекты для моделирования и последующей автоматизации.
3. Выполнить функциональное и информационное моделирование процессов по организации транспортировки груза.
4. Спроектировать архитектуру веб-сайта.
5. Разработать интерфейс веб-сайта.
6. Организовать корректную работу с базой данных.
7. Разработать систему, которая позволит принимать решения по подбору транспортных средств.
8. Провести тестирование разработанного программного средства, выявить возможные неполадки и своевременно их исправить.

**1 ПОСТАНОВКА ЗАДАЧИ**

В соотвествии с назначением разарбатываемой программы – решением проблемы о выборе средства транспорта для перевозки того или иного груза, требуется достигнуть поставленную цель и задачи, которые включены в её реализацию.

Для этого требуется разработать простое и удобное в использовании программное приложение с несложным интерфейсом, при помощи которого сотрудники логистической компании смогут принимать грамотные и взвешенные решения.

Следовательно, в приложении должны быть реализованы:

1. Система регистрации для пользователей программного приложения, предназначенная для получения доступа к использованию ресурсов приложения.
2. Механизм авторизации уже зарегестрированных в системе пользователей, который необходим для того, чтобы разделить права администратора и пользователя.
3. Простой, интуитивно понятный интерфейс для широкой аудитории пользователей, в том числе неподготовленных.
4. Меню с набором пользовательских функций.
5. Меню с набором фцнкций для администратора.
6. Продуманный механизм обработки всевозможных исключительных ситуаций, которые могут возникать при некорректно введённой информации или же в случае неправильного использования программного приложения.

Для реализации главной задачи приложения – подбора вида траснпорта, будет использован метод ранга.

Полученный после решения поставленных в работе задач программный продукт, разрабатываемый в рамках данного курсового проекта, даст возможность оптимизировать процесс работы компании и не допустить ошибки в принятии решения по выбору транспорта для перевозки груза.

**2 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**

В данном курсовом проекте рассматривается деятельность логистической компании. Для примера взята компания под названием «LogistOn».

Для проведения детального анализа деятельности компании была смоделирована организационная структура (см. в приложении А на рисунке А.1) и стратегическая карта (см. в приложении А на рисунке А.2).

Стратегическая карта – это элемент системы сбалансированных показателей. Она представляет собой диаграмму, на которой обозначены основные цели существования организации. Для организации логистической компании «LogistOn» выделено 4 основные перспективы стратегической карты: Финансы, Внутренние бизнес-процессы, Обучение и развитие, Клиенты. Перспективы предназначены для понимания того, к какому виду внутренней деятельности относится та или иная цель. Главной целью на стратегической карте компании является увеличения прибыли. Остальные являются средствами достижения этой главной цели.

Также была смоделирована функциональная модель IDEF0. Цель моделирования – показать функционирование компании в процессе организации перевозки груза. Главным процессом модели является «Осуществление транспортировки материалов» (рисунок 2.1).

title

Рисунок 2.1 – Контекстная диаграмма верхнего уровня

На входе в диаграмме запрос заказчика, исходя из которого и формируется дальнейшая деятельность компании. Так же на входе имеются средства оплаты и материалы для транспортировки. На выходе, соответственно, получаем отчёт выполненном запросе, чек о проведении транзакции и осуществленную перевозку материалов. Механизмами данного процесса выступает персонал, оборудование и техника, заказчик и транспортные средства.

Процесс осуществления транспортировки материалов разбивается на 6 процессов: «Подписание договора», «Запись характеристик материалов в форму», «Просмотр и анализ критериев», «Запуск поиска наилучшего вида транспорта», «Транспортировка материалов» и «Оплата услуг компании» (рисунок 2.2).

title

Рисунок 2.2 – Декомпозиция первого уровня

Далее остановимся на основном процессе области деятельности: «Запуске поиска наилучшего вида транспорта» (рисунок 2.3). В данном процессе участвует персонал вместе с оборудованием и техникой. Управление данным процессом осуществляют критерии выбора и транспорт компании. На вход для данного процесса поступают проанализированные критерии, на выходе имеем наилучший вариант транспорта.

Данный процесс состоит из следующих подпроцессов: «открытие формы подбора вида транспорта», «Ввод характеристик для транспортируемых материалов», «Подбор вида транспорта» и «Вывод результат подбора».

title

Рисунок 2.3 – Декомпозиция блока «Запуск поиска наилучшего вида транспорта»

Блок «Подбор вида транспорта» так же разбивается на составляющие, а именно: «Получение характеристик материалов», «Подбор самого подходящего вида транспорта» и «Получение наилучшего варианта» (рисунок 2.4).

Более полное представление модели IDEF0 можно найти в приложение А на рисунках А.3, А.4, А.5, А.6 и А.7.

title

Рисунок 2.4 – Декомпозиция блока «Подбор вида транспорта»

**3 ПОСТРОЕНИЕ БАЗОВОЙ АНАЛИТИЧЕСКОЙ**

**МОДЕЛИ**

В курсовом проекте для решения задачи о подборе наиболее подходящего транспортного средства для перевозки материалов используется метод ранга.

Данный метод основан на балльных оценках альтернатив, указываемых несколькими экспертами. Каждый из экспертов (независимо от других) оценивает альтернативы по некоторой шкале (обычно — 10-балльной). Чем более предпочтительной (по мнению эксперта) является альтернатива, тем более высокий балл для нее указывается. При этом оценки могут быть как целыми, так и дробными. В этих условиях веса целей определяются следующим образом:

1. Составляется матрица оценок экспертов:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Эj/Zi** | **Z1** | **Z2** | **...** | **Zn** |
| **Э1** | S11 | S12 | ... | S1n |
| **Э2** | S21 | S22 | ... | S2n |
| **...** | ... | ... | ... | ... |
| **Эm** | Sm1 | Sm2 | ... | Smn |

1. 0≤pji≤10 (j = 1,m, i = 1,n)
2. Составляется матрица нормированных оценок:

ω = pji/∑pji (j = 1,m, i = 1,n)

1. Вычисляются искомые веса целей:

ωi = ∑ωij/∑∑ωij (i = 1,n) ∑ωi = 1

**4 МОДЕЛИРОВАНИЕ СИСТЕМЫ**

# **4.1 Диаграмма вариантов использования**

На рисунке 4.1 представлена диаграмма вариантов использования (use case diagram) данного программного приложения. На диаграмме отображается отношение между пользователем и администратором и вариантами использования проектируемой системы.

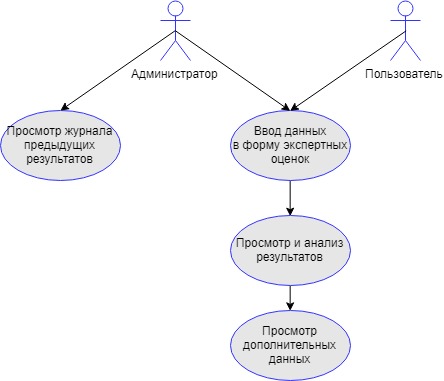
****

Рисунок 4.1 – Диаграмма вариантов использования

# **4.2 Моделирование базы данных**

В данном курсовом проекте разрабатывается программное приложение с организацией взаимодействия с базой данных на объектно-ориентированном языке Java. В качестве базы данных была выбрана СУБД MySQL.

СУБД MySQL является универсальной и обладает всеми необходимыми средствами необходимыми для реализации базы данных. MySQL является наиболее приспособленной для применения в среде web СУБД.

MySQL является надежным стандартом в области СУБД для web. Она идёт в ногу с СУБД таких производителей, как Oracle, IBM, Microsoft и Sybase.

Название базы данных в моём курсовом проекте – «course» (рисунок 4.2). В базе данных course присутствует таблица «users», в которой хранятся данные о зарегистрированных пользователях, а именно:

* login (логин);
* password (пароль);
* idusers (идентификационный номер).

В таблице «weight» хранятся искомые веса целей:

* first (первый коэффициент);
* second (второй коэффициент);
* third (третий коэффициент);
* fourth (четвертый коэффициент);
* idweight (идентификационный номер).

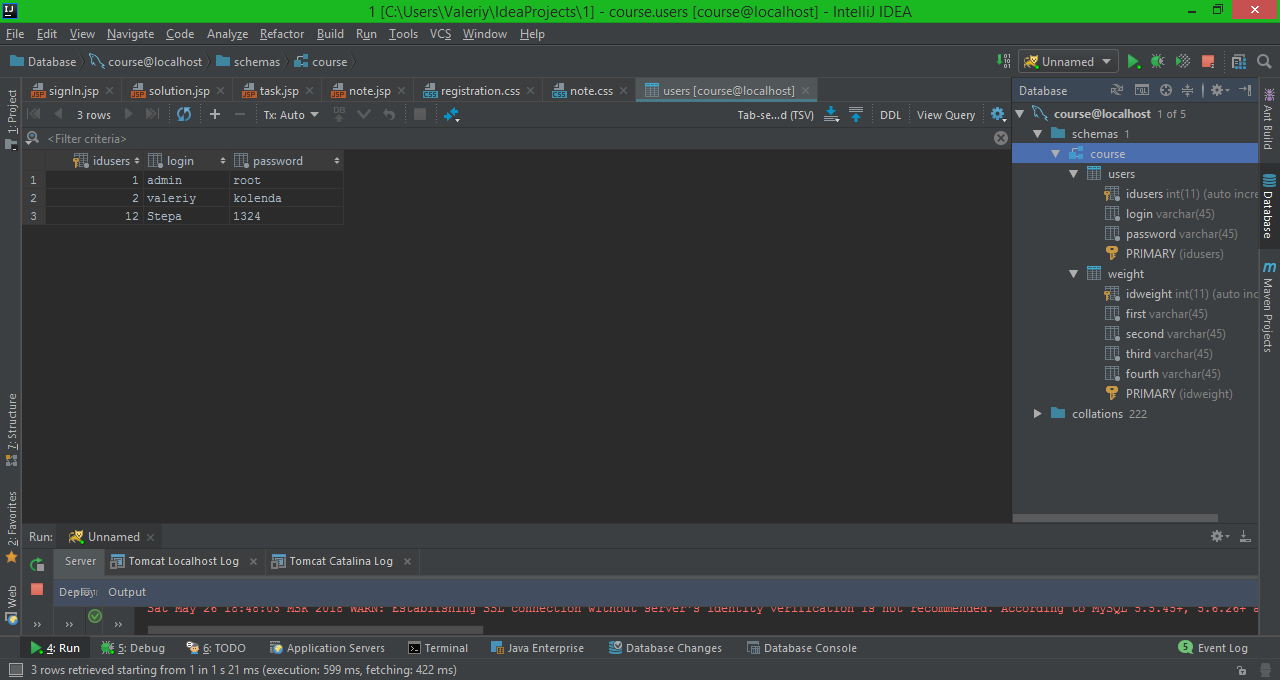


Рисунок 4.2 – База данных «course» и её таблицы

**5 АЛГОРИТМ РАБОТЫ СИСТЕМЫ**

# **5.1 Блок-схема работы программы**

Блок-схема работы программы представлена в приложении Б на рисунке Б.1.

Вход в программу требует установление соединения с сервером. Если подключение отсутствует, то программа работать не будет. Если подключение к серверу произошло успешно, то работа программы будет продолжаться до того момента, пока не будет осуществлён выход из нее.

После происходит вход в учетную запись и следует либо регистрация пользователя, если она еще не была выполнена, либо вход пользователем в программу, где будет предложен весь её функционал.

**5.2 Блок-схема алгоритма выбора наиболее подходящего**

**варианта транспортировки методом предпочтения**

При входе в раздел решения задачи вводятся оценки экспертов, после чего приводится в действие внутреннее функционирование программы по поиску наилучшего вида транспорта для перевозки методом ранга. Блок-схема алгоритма решения задачи методом предпочтения представлена на рисунке 5.2.



Рисунок 5.2 – Блок-схема алгоритма поиска наиболее

подходящего варианта методом предпочтения

**6 ОБОСНОВАНИЕ ВЫБОРА ПРОГРАММНЫХ**

**СРЕДСТВ РАЗРАБОТКИ**

Языком для разработки приложения является Java. Java — объектно-ориентированный язык программирования, разрабатываемый компанией Sun Microsystems с 1991 года и официально выпущенный 23 мая 1995 года. Изначально новый язык программирования назывался Oak (James Gosling) и разрабатывался для бытовой электроники, но впоследствии был переименован в Java и стал использоваться для написания апплетов, приложений и серверного программного обеспечения.

Одно из основных преимуществ языка Java — независимость от платформы, на которой выполняются программы*:* один и тот же код можно запускать под управлением операционных систем Windows, Solaris, Linux, Machintosh и др.

Кроме того, Java — полностью объектно-ориентированный язык, даже в большей степени, чем C++*.* Все сущности в языке Java являются объектами, за исключением немногих основных типов (primitive types).

Для разработки клиент-серверного приложения в курсовом проекте была использована технология Java Server Pages (JSP), которая является составной частью единой технологии создания бизнес-приложений J2EE.

JSP — это альтернативная методика разработки приложений, динамически генерирующих ответ на те или иные запросы клиента.

Страница JSP содержит текст двух типов: статические исходные данные, которые могут быть оформлены в одном из текстовых форматов [HTML](https://ru.wikipedia.org/wiki/HTML), [SVG](https://ru.wikipedia.org/wiki/SVG), [WML](https://ru.wikipedia.org/wiki/WML), или [XML](https://ru.wikipedia.org/wiki/XML), и JSP- элементы, которые конструируют динамическое содержимое. Кроме этого могут использоваться библиотеки JSP-тегов, а также [EL](https://ru.wikipedia.org/wiki/Expression_Language) ([Expression Language](https://ru.wikipedia.org/wiki/Expression_Language" \o "Expression Language)), для внедрения Java-кода в статичное содержимое JSP-страниц.

**7 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ**

В данной главе всё внимание будет уделено основным классам и их методам, которые на всём протяжении работы программы выполняют определенные функции.

Так, одним из таких классов является класс DatabaseHandler, который содержит следующие методы:

* getInstance(), который создает новый объект, если он еще не был создан, или возвращает уже существующий.
* signUp(), который позволяет еще незарегистрированным пользователям создать свой аккаунт и тем самым обеспечит им возможность воспользоваться функционалом данного приложения. В методе предусмотрены разного рода проверки. Так, пользователь не сможет ввести «нулевой» логин или пароль, а также логин, который уже занят другим пользователем. Для завершения регистрации вам будет предложено повторить пароль, в случае ошибки вам придется вводить данные снова.
* getNotes(), который тесно связывает классы DatabaseHandler и Note. Данный метод позволяет пользователю, который вошел в систему как администратор, увидеть все записи решенных задач.

Не менее важным классом в данной программе является класс UserInfo, который включает в себя следующие методы:

* setters and getters, которые отвечают за ввод и предоставление информации соответственно.
* Compare(), благодаря данному методу происходит сравнение логина и пароля, которые ввел пользователь с теми, которые уже находятся в базе данных. В первую очередь происходит сравнение логинов, при наличии совпадения сравниваются пароли и в случае успешного ввода данных вы войдете в свой личный кабинет.
* Entering(), данный метод сверяет логины, из базы данных с введенным логином при регистрации пользователя в системе и обеспечивая тем самым надежную помощь методу signUp() из класса DatabaseHandler.

Одним из основных классов является класс Task, который обладает методом solution(), выполняющим основную задачу программы: нахождение весовых коэффициентов методом ранга.

Последним в рассмотрении – класс Note, который имеет определенные сходства с ранее описанными классами, т.к. имеет свои getters и setters, а также активно взаимодействует с базой данных «course».

Его основным методом является addNote(), который после решения задачи заносит подсчитанные данные в таблицу, которую в дальнейшем сможет просмотреть администратор

**8 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

Для того, чтобы начать работу с приложением, требуется зайти на главную страницу сайта (рисунок 8.1), на которой представлена возможность регистрации в приложении, входа зарегистрированному пользователю и, конечно же, решении задачи курсового проекта – подбора транспорта для перевозок грузов.

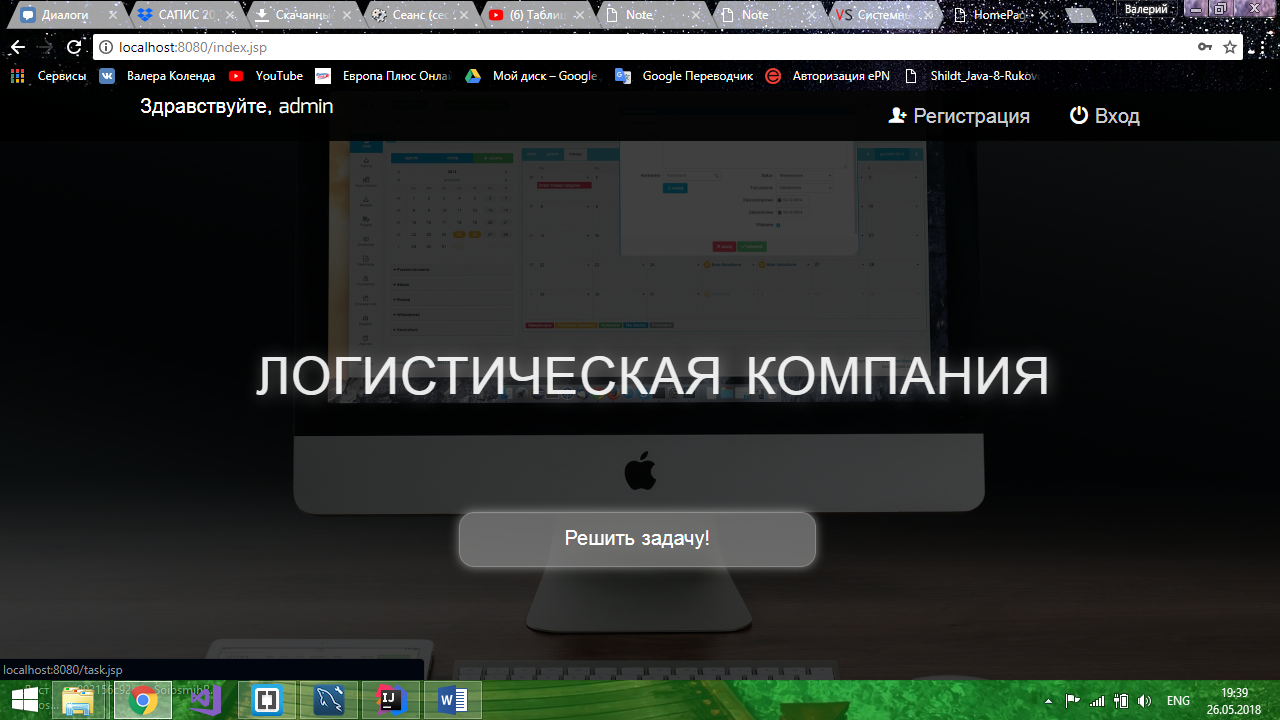


Рисунок 8.1 – Главная страница сайта

Далее, для начала работы с приложением, нужно пройти регистрацию. Регистрационная форма для новых пользователей представлена на рисунке 8.2. Пользователь, который не зарегистрировался в системе, не имеет доступа к функциональным возможностям приложения.

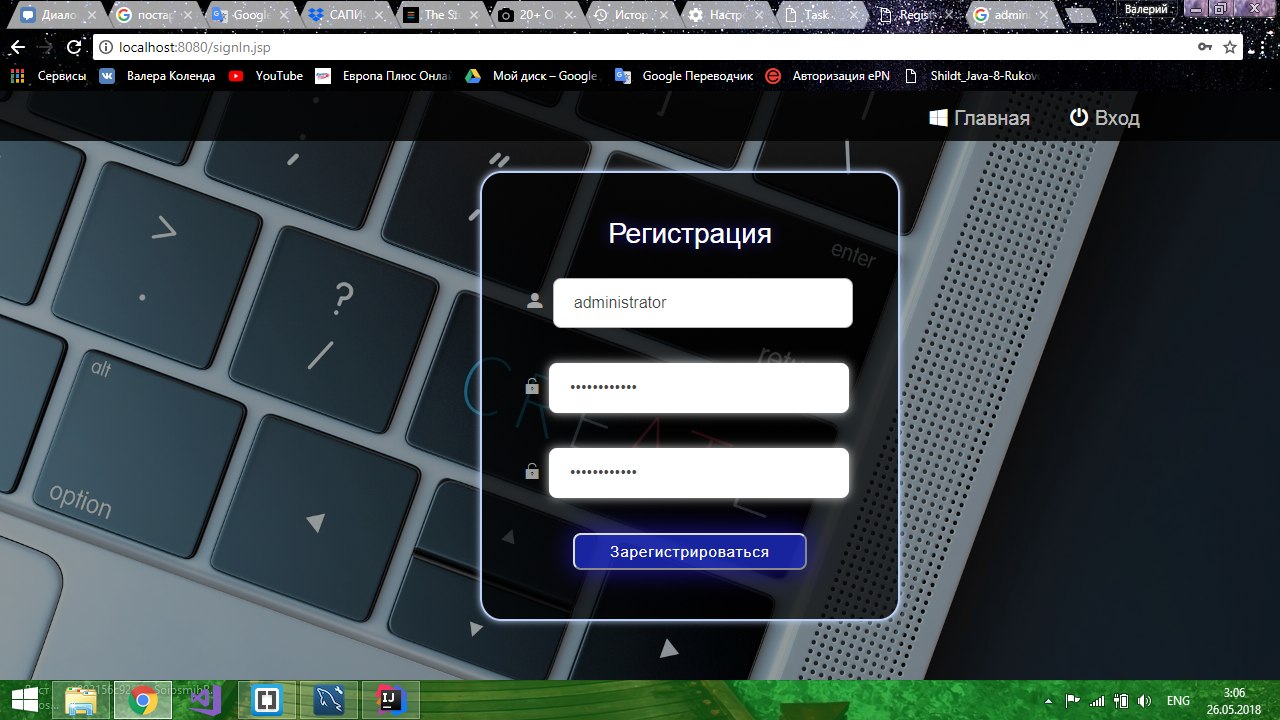


Рисунок 8.2 – Форма регистрации нового пользователя

Если пользователь ранее проходил регистрацию в приложении, то после входа в него ему требуется нажать на кнопку «вход» на главной странице сайта (рисунок 8.1) и ввести свои данные (логин и пароль) в открывшуюся форму входа (рисунок 8.3). Если же пользователь впервые пользуется приложением, то в форму входа ему требуется войти только после успешно пройденной регистрации.

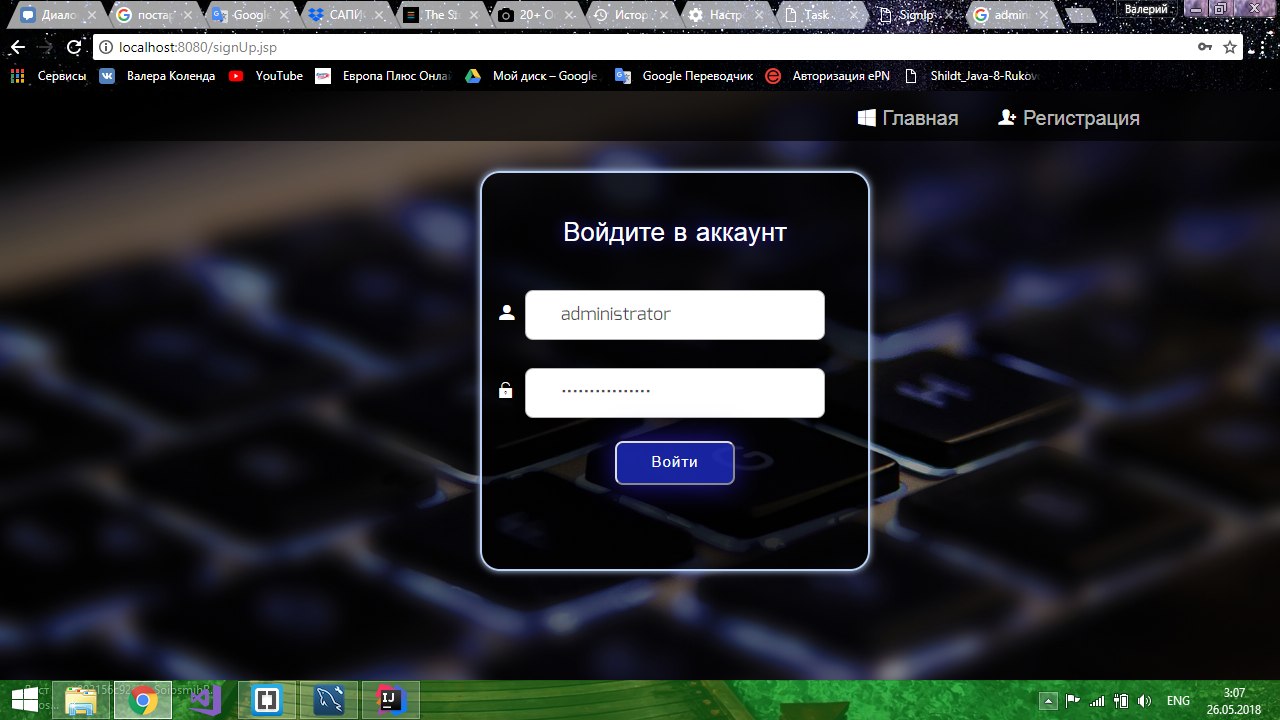


Рисунок 8.3 – Форма входа в аккаунт

После входа в аккаунт пользователям становится доступна форма для решения задачи о выборе средства транспортировки грузов (рисунок 8.4). В этой форме пользователь должен ввести оценки экспертов относительно транспортируемого груза и нажать кнопку «Обработать значения» для получения результата. В случае ошибочного ввода значений можно сделать сброс, нажав на кнопку «Сброс значений».

В случае успешной регистрации и выполненного входа пользователю становится доступна самая главная функция данного приложения – подбор транспорта для перевозки груза. В открывшуюся форму для ввода значений (рисунок 8.4) требуется ввести оценки экспертов и, если данные были введены без ошибок, нажать кнопку «Обработать значения», если же ошибки были допущены можно нажать на кнопку «Сброс значений» и повторить ввод данных, после чего, соответственно, нажать кнопку «Обработать значения».

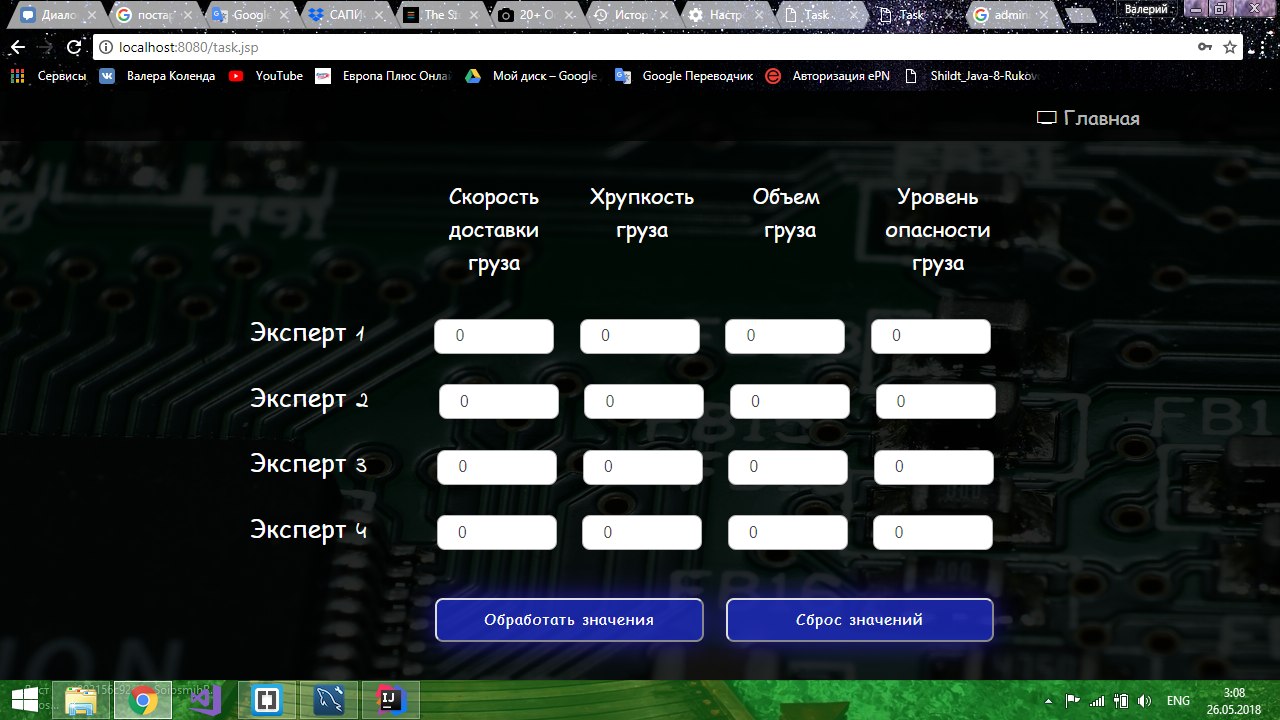


Рисунок 8.4 – Матрица ввода экспертных оценок

После обработки введённых значений приложение показывает результат своей работы: весовые коэффициенты всех видов транспорта и наилучший вариант, выбранный из них (рисунок 8.5). Если пользователю требуется более подробная информация, то он может получить ее, нажав кнопку «справка», находящуюся под весовыми коэффициентами.



Рисунок 8.5 – Результат задачи

Если же у пользователя имеются права администратора, то у него есть возможность просматривать результаты предыдущих решенных программой задач. Это реализуется нажатием на кнопку «блокнот». В итоге на экран будет выведена сводная таблица с весовыми коэффициентами (рисунок 8.6).



Рисунок 8.6 – Таблица коэффициентов ранее решенных задач

**9 КОНТРОЛЬНЫЙ ПРИМЕР**

Согласно задуманной концепции, в данном программном приложении группа независимых экспертов оценивают подготавливаемый к транспортировке груз в соответствие со следующими критериями: скорость доставки груза, хрупкость груза, объём груза, уровень опасности груза. В зависимости от этих характеристик эксперты принимают решение об оценке того или иного параметра груза, а система, свою очередь, подбирает наиболее подходящий вариант транспортировки.

Оценки экспертов предложенных вариантов приведены в матрице:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Эj/Zi** | **Z1** | **Z2** | **Z3** | **Z4** |
| **Э1** | 8 | 3 | 2 | 5 |
| **Э2** | 3 | 1 | 7 | 4 |
| **Э3** | 2 | 4 | 3 | 8 |
| **Э4** | 1 | 9 | 4 | 2 |

В данном примере первый эксперт считает, что наибольшую важность при транспортировке именно этого груза имеет скорость доставки груза; менее важный фактор — опасность груза, еще немного менее важный — хрупкость груза; значительно менее важный фактор — объём груза.

Находим суммарные оценки альтернатив всеми экспертами:

C1 = 8+3+2+5 = 18;

C2 = 3+1+7+4 = 15;

C3 = 2+4+3+8 = 17;

C4 = 1+9+4+2 = 16.

Составляем матрицу нормированных оценок:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Эj/Zi** | **Z1** | **Z2** | **Z3** | **Z4** |
| **Э1** | 8/18 | 3/18 | 2/18 | 5/18 |
| **Э2** | 3/15 | 1/15 | 7/15 | 4/15 |
| **Э3** | 2/17 | 4/17 | 3/17 | 8/17 |
| **Э4** | 1/16 | 9/16 | 4/16 | 2/16 |

Искомые веса целей:

ω1 = 0,206; ω2 = 0,257; ω3 = 0,251; ω4 = 0,285.

Наиболее предпочтительной, по мнению экспертов, является альтернатива, имеющая максимальный вес, а именно вторая. Следовательно, наиболее предпочтительным видом транспорта в данном случае является железнодорожный.

Попробуем запустить программу и проверить ее работу на таких же данных. Результат работы программы (рисунок 9.1 и рисунок 9.2):

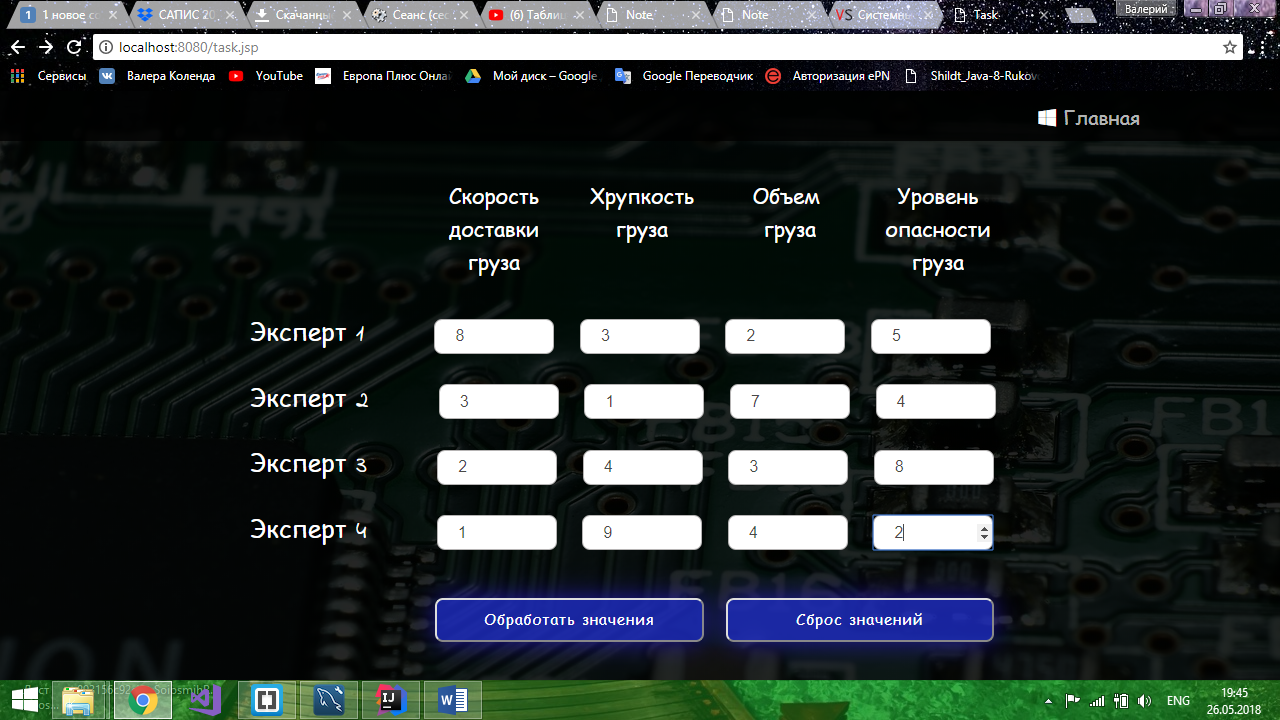


Рисунок 9.1 – Ввод данных

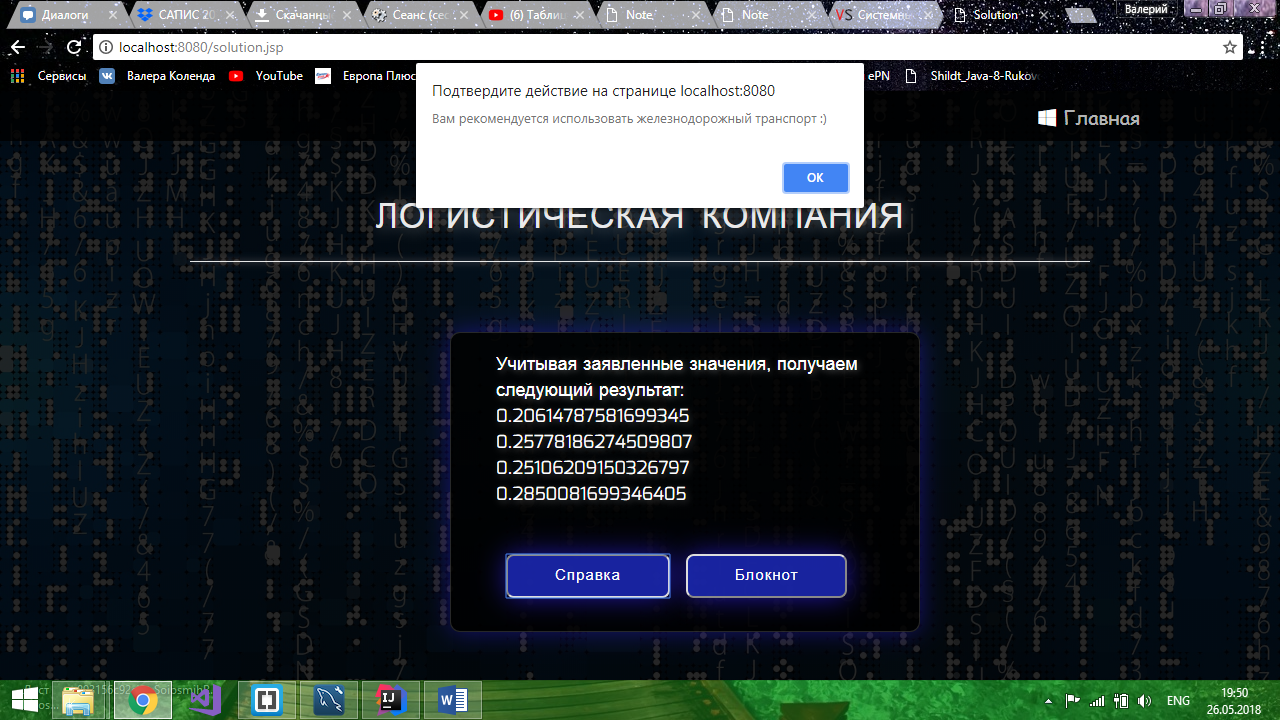


Рисунок 9.2 – Результат решения задачи

# 

# **ЗАКЛЮЧЕНИЕ**

В данном курсовом проекте было разработано программное приложение по выбору вида транспорта для транспортировки грузов. Итогом данного курсового проекта является функционирующее программное приложение, пригодное для использования сотрудниками логистической компании при работе с разного рода перевозками.

Данное программное приложение можно применять для любой другой предметной области при расчете оптимального решения, так как принцип расчета функции и переменных останется прежним.

В проекте предусмотрено решение задачи, а также реализован режим администрирования. Предусмотрена сохранность данных, то есть обычный пользователь не имеет доступа к данным об заказах других пользователей. Кроме этого, в программе предусмотрены исключительные ситуации, которые могут возникнуть в процессе использования данного программного продукта. Всё это повышает продуктивность и надежность программы, а также обеспечивает высокую производительность.

В результате разработанная программа оснащена простым и удобным для пользователей интерфейсом, несложной системой навигации и необходимыми функциями для основной задачи курсового проекта. Данное приложение позволит принимать более взвешенные и продуманные решения для решения вопроса о виде транспорта, используемого при транспортировке грузов. Сотрудники смогут оптимизировать свою работу и выполнять её более качественно и быстро.

Подводя итог всему вышесказанному, можно утверждать, что разработка таких программ позволит оптимизировать и улучшить работу многих логистических компаний, а также предоставит возможный вариант увеличения прибыли при минимальных затратах. Данное приложение можно усовершенствовать путём доработки деталей интерфейса или добавления новых функций. Иными словами, приложение имеет возможность изменения и расширения своей функциональности, а также возможные перспективы использования и развития. Следовательно, программа может стать еще более компактной, оптимизированной и многофункциональной.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Эккель Б. Философия Java. Библиотека программиста. – СПб.: Питер, 2001. – 880 с.: ил.
2. Дейт, К. Введение в системы баз данных/ К. Дейт: пер. с англ. – 6—е изд. — СПб.: Издательский дом «Вильямс», 2000.— 848 с.
3. Code Conventions[Электронный ресурс]. – Электронные данные. Режим доступа: http://www.oracle.com/technetwork/java/codeconvtoc-136057.html
4. MySQL Workbench [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.mysql.com/products/workbench/
5. Кулябов Д.С., Королькова А.В. Введение в формальные методы описания бизнес-процессов. Учебное пособие. - М.: РУДН, 2008. - 173 с.

ПРИЛОЖЕНИЕ **А**

**Модели системы**

title

Рисунок А.1 - Организационная структура предприятия

Продолжение приложения А

title

Рисунок А.2 – Стратегическая карта компании

Продолжение приложения А

IDEF0

title

Рисунок А.3 – Декомпозиция блока «Подписание договора»

title

Рисунок А.4 – Декомпозиция блока «Запись хар-к материалов в форму»

Продолжение приложения А

title

Рисунок А.5 – Декомпозиция блока «Просмотр и анализ критериев»

title

Рисунок А.6 – Декомпозиция блока «Транспортировка материалов»

Продолжение приложения А

title

Рисунок А.7 – Декомпозиция блока «Оплата услуг компании»

ПРИЛОЖЕНИЕ **Б**

**Схема работы программы**

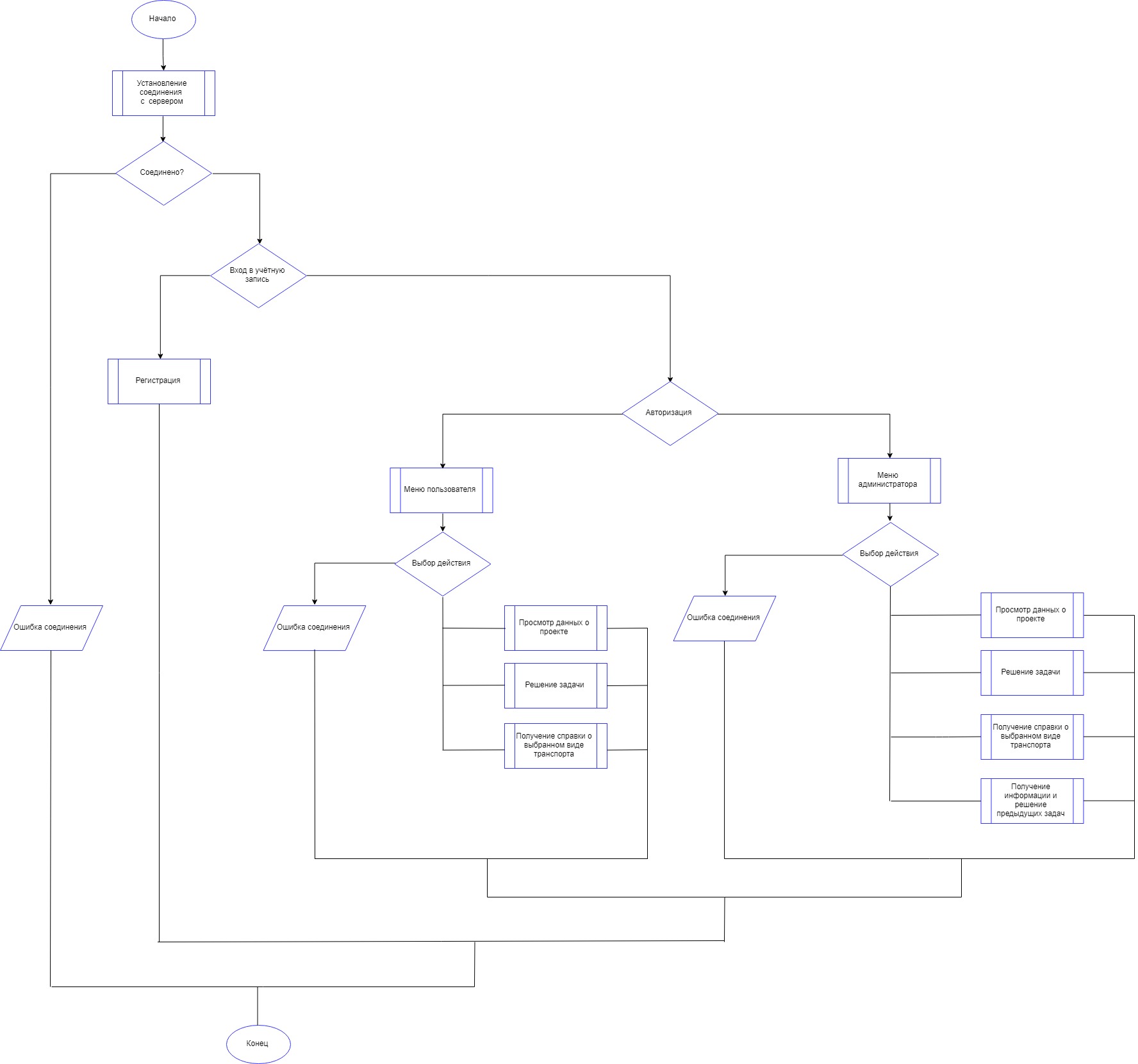
****

Рисунок Б.1 – Схема программы

ПРИЛОЖЕНИЕ **В**

**Листинг кода программы**

Class DatabaseHandler

package jspClasses;

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

public class DatabaseHandler {

private static DatabaseHandler handler;

private Statement st;

private DatabaseHandler() {

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/course?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC", "root", "1324");

this.st = con.createStatement();

}catch(Exception e){

System.out.println(e);

}

}

public static DatabaseHandler getInstance() {

if (DatabaseHandler.handler == null) {

DatabaseHandler.handler = new DatabaseHandler();

}

return DatabaseHandler.handler;

}

public boolean signUp(String userName, String pass, String passConfirm) throws SQLException {

if ( userName != null && pass != null && pass.equals(passConfirm) ) {

UserInfo user = new UserInfo();

Продолжение приложения В

boolean bool = user.entering(userName, pass);

if (bool) {

st.executeUpdate("insert into users( login, password)values('" + userName + "' , '" + pass + "')");

return true;

}

else

return false;

}

else

return false;

}

public List<Note> getNotes() throws SQLException {

ResultSet rs = st.executeQuery("SELECT \* FROM weight");

List<Note> notesList = new ArrayList<Note>();

while (rs.next()) {

Note record = new Note(

rs.getInt("idweight"),

rs.getDouble("first"),

rs.getDouble("second"),

rs.getDouble("third"),

rs.getDouble("fourth")

);

notesList.add(record);

}

return notesList;

}

}

Class UserInfo

package jspClasses;

import java.sql.\*;

public class UserInfo {

private int id;

private String login;

Продолжение приложения В

private String password;

private Statement st;

public UserInfo() {

try {

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/course?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC", "root", "1324");

this.st = con.createStatement();

} catch (Exception e) {

System.out.println(e);

}

}

public void setId(int id) {

this.id = id;

}

public void setLogin(String login) {

this.login = login;

}

public void setPassword(String password) {

this.password = password;

}

public int getId() {

return id;

}

public String getLogin() {

return login;

}

public String getPassword() {

return password;

}

public boolean compare(String login, String password) throws SQLException {

if ( login != null && password != null) {

Продолжение приложения В

ResultSet rs = st.executeQuery("SELECT \* FROM users");

while (rs.next()) {

UserInfo man = new UserInfo();

man.setId(rs.getInt("idusers"));

man.setLogin(rs.getString("login"));

man.setPassword(rs.getString("password"));

if (man.login.equals(login)) {

if (man.password.equals(password))

return true;

}

}

}

return false;

}

public boolean entering(String login, String password) throws SQLException {

ResultSet rs = st.executeQuery("SELECT \* FROM users");

while (rs.next()) {

UserInfo man = new UserInfo();

man.setId(rs.getInt("idusers"));

man.setLogin(rs.getString("login"));

man.setPassword(rs.getString("password"));

if (man.login.equals(login))

return false;

}

return true;

}

}

Class Task

package jspClasses;

import java.sql.\*;

public class Task {

Продолжение приложения В

private Statement st;

public Task() {

try {

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/course?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC", "root", "1324");

this.st = con.createStatement();

} catch (Exception e) {

System.out.println(e);

}

}

public double [] solution(double [][] array){

double [][] modifyMatrix = new double[4][4];

final int N = 4;

double [] divider = new double[4];

double [] weight = new double[4];

for(int i = 0; i < N; i++){

for(int j = 0; j < N; j++){

divider[i] += array[i][j];

}

}

for(int i = 0; i < N; i++){

for(int j = 0; j < N; j++){

modifyMatrix [i][j] =(array[i][j] / divider[i]);

}

}

for(int i = 0; i < N; i++){

for(int j = 0; j < N; j++) {

weight[i] += modifyMatrix[j][i] / N;

}

}

return weight;

}

}

Продолжение приложения В

Class Note

package jspClasses;

import java.sql.\*;

public class Note {

double firstWeight;

double secondWeight;

double thirdWeight;

double fourthWeight;

private Statement st;

int id;

public void setId(int id) {

this.id = id;

}

public void setFirstWeight(double firstWeight) {

this.firstWeight = firstWeight;

}

public void setSecondWeight(double secondWeight) {

this.secondWeight = secondWeight;

}

public void setThirdWeight(double thirdWeight) {

this.thirdWeight = thirdWeight;

}

public void setFourthWeight(double fourthWeight) {

this.fourthWeight = fourthWeight;

}

public int getId() {

return id;

}

public double getFirstWeight() {

return firstWeight;

}

public double getSecondWeight() {

Продолжение приложения В

return secondWeight;

}

public double getThirdWeight() {

return thirdWeight;

}

public double getFourthWeight() {

return fourthWeight;

}

public Note() {

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/course?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC", "root", "1324");

this.st = con.createStatement();

}catch(Exception e){

System.out.println(e);

}

}

public Note(int id, double firstWeight, double secondWeight, double thirdWeight, double fourthWeight) {

this.id = id;

this.firstWeight = firstWeight;

this.secondWeight = secondWeight;

this.thirdWeight = thirdWeight;

this.fourthWeight = fourthWeight;

}

public void addNote(double first, double second, double third, double fourth) throws SQLException {

st.executeUpdate("insert into weight( first,second,third,fourth)values('" + first + "' , '" + second + "','" + third + "' , '" + fourth + "')");

}

}