

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ

**РАЗРАБОТКА МОДУЛЯ УПРАВЛЕНИЯ ОТХОДАМИ ПРОИЗВОДСТВА
В СТРОИТЕЛЬСТВЕ НА БАЗЕ 1С: УПРАВЛЕНИЕ НЕБОЛЬШОЙ
ФИРМОЙ**

Выпускная квалификационная работа
обучающегося по направлению подготовки 09.03.03
«Прикладная информатика»
очной формы обучения, группы 07001304
Мандрыкиной Натальи Евгеньевны

Научный руководитель:
к.т.н. профессор Ломакин В.В.

БЕЛГОРОД 2017

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 Аналитическая часть.....	7
1.1 Техничко-экономическая характеристика предметной области	7
1.1.1 Характеристика предприятия.....	7
1.1.2 Краткая характеристика подразделения или видов его деятельности ..	8
1.2 Экономическая сущность задачи.....	12
1.3. Обоснование необходимости и цели использования вычислительной техники для решения задачи.....	13
1.4. Постановка задачи	16
1.4.1 Цель и назначение автоматизированного варианта решения задачи ..	16
1.4.2 Общая характеристика организации решения задачи на ЭВМ.....	17
1.5. Анализ существующих разработок и обоснование выбора технологии проектирования.....	19
2 Проектная часть.....	23
2.1 Обоснование проектных решений.....	23
2.1.1 Обоснование проектных решений по техническому обеспечению (ТО)	23
2.1.2 Обоснование проектных решений по информационному обеспечению (ИО).....	24
2.1.3 Обоснование проектных решений по программному обеспечению (ПО).....	26
2.1.4 Обоснование проектных решений по технологическому обеспечению	29
2.1.5 Обоснование выбора программных средств	30
2.2 Информационное обеспечение задачи (комплекса задач, АРМ).....	35

2.2.1 Информационная модель и ее описание	35
3 Программная реализация проектных решений	39
3.1 Программное обеспечение комплекса задач	39
3.1.1 Обеспечение хранения условно-постоянной информации	39
3.1.2 Отражение основных операций в информационной системе	45
3.2 Технологическое обеспечение задачи (комплекса задач, АРМ).....	47
3.3 Описание контрольного примера работы объектов конфигурации	48
3.4. Обоснование экономической эффективности	51
3.4.1 Целесообразность разработки с экономической точки зрения	51
3.4.2 Расчет экономической эффективности разработки	52
ЗАКЛЮЧЕНИЕ	59
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	60
ПРИЛОЖЕНИЕ	62

ВВЕДЕНИЕ

В современном мире, наверное, каждое предприятие заинтересовано в эффективном управлении своими ресурсами, как денежными, так и материальными. Поскольку от грамотной организации всех составляющих зависит рентабельность любого предприятия.

В данной выпускной квалификационной работе рассматривается деятельность строительных организаций, которую можно автоматизировать с целью повышения эффективности работы.

Строительные организации это организации, основным видом деятельности которых является строительство новых объектов (зданий, сооружений), а также реконструкция, капитальный ремонт и расширение действующих объектов.

В деятельности любого предприятия всегда есть отходы производства, в том числе и в строительных организациях. Поэтому перед организациями в сфере строительства возникает проблема управления транспортировкой отходов. Отходы это одна из экологических проблем в современном мире, которая несет в себе потенциальную опасность для жизни и здоровья людей, а также для окружающей среды.

Все вышеописанное доказывает, то выбранная для выпускной квалификационной работы тема исследования актуальна.

Объектом исследования выпускной работы является деятельность предприятия в строительстве.

Предмет исследования – транспортировка отходов на полигоны.

Для разработки модуля управления отходами был использован программный продукт «1С: Предприятие 8.3». «1С: Предприятие 8.3» является совокупностью программных решений для автоматизации различных областей в деятельности любого предприятия.

Данная разработка реализована на базе типовой конфигурации «1С: Управление небольшой фирмой 8».

«1С: Управление небольшой фирмой 8» (УНФ) – это готовое решение для автоматизации оперативного управления на предприятиях малого бизнеса. В программе реализовано все самое необходимое для ведения оперативного учета, контроля, анализа и планирования на предприятии.

Целью данной выпускной работы является повышение эффективности деятельности организации за счет снижения расхода времени и денег.

Для достижения поставленной цели необходимо решить следующие задачи:

- изучение и анализ деятельности организации;
- выявление недостатков в деятельности организации;
- проектирование структурно-функциональных диаграмм;
- разработка решения по устранению недостатков в работе организации;
- тестирование и последующее внедрение разработанного решения.

В соответствии с поставленной целью и задачами выпускной работы были определены следующие методы исследования:

- теоретический анализ нормативно-правовой документации по предметной области;
- изучение различных систем позволяющих управлять отходами в строительстве, их сравнение.

Структура выпускной квалификационной работы обусловлена ее предметом, целью и задачами. Работа состоит из введения, трех глав, заключения и приложений.

Введение раскрывает актуальность, определяет объект, предмет, цели, задачи и методы исследования.

В первой главе дается подробное представление аналитической части, целью которой является рассмотрение существующего состояния организации работы в сфере строительства и постановка задачи организации процессов

управления отходами.

Вторая глава включает обоснование проектных решений по техническому, информационному, программному и технологическому обеспечению организации процессов управления отходами.

Третья глава посвящена описанию проектной части выпускной работы. В данном разделе представлена информация обо всех принятых решениях во время выполнения организации процессов управления отходами. Проектная часть является решением проблем, изложенных в аналитической части. А так же представлена организационно-экономическая составляющая выпускной работы.

В заключении определяется, какие задачи были решены, отражается практическая значимость полученных результатов.

1 Аналитическая часть

1.1 Технико-экономическая характеристика предметной области

1.1.1 Характеристика предприятия

Компания «Строй Дизайн» успешно функционирует на рынке строительных услуг с 2006 года. Основным видом деятельности является «Строительство жилых и нежилых зданий».

Деятельность компании заключается в качественном строительстве зданий, ремонте квартир, возведении и улучшении жилищных условий жителей города Белгорода и Белгородской области. На предприятии трудятся высококвалифицированные мастера со специализированным образованием, постоянные профессиональные бригады рабочих, что способствует качественному и красивому воплощению желаний и потребностей клиентов.

Компанией «Строй Дизайн» было выполнено множество строительных объектов, ремонтных и отделочных работ качество и надежность которых позволяют сказать о том, что фирма уже прочно утвердилась в строительной сфере.

Демократичные цены и гибкая ценовая политика отличают ООО «Строй Дизайн» от других строительных компаний Белгорода и вызывают уважение со стороны клиентов.

1.1.2 Краткая характеристика подразделения или видов его деятельности

Штат сотрудников ООО «Строй Дизайн» составляет высококвалифицированные специалисты. Организационная структура включает в себя следующие подразделения:

- руководство;
- бухгалтерия;
- отдел продаж;
- производственный отдел;
- отдел снабжения;
- отдел информационных технологий.

Организационная структура, представлена на рисунке 1.1:

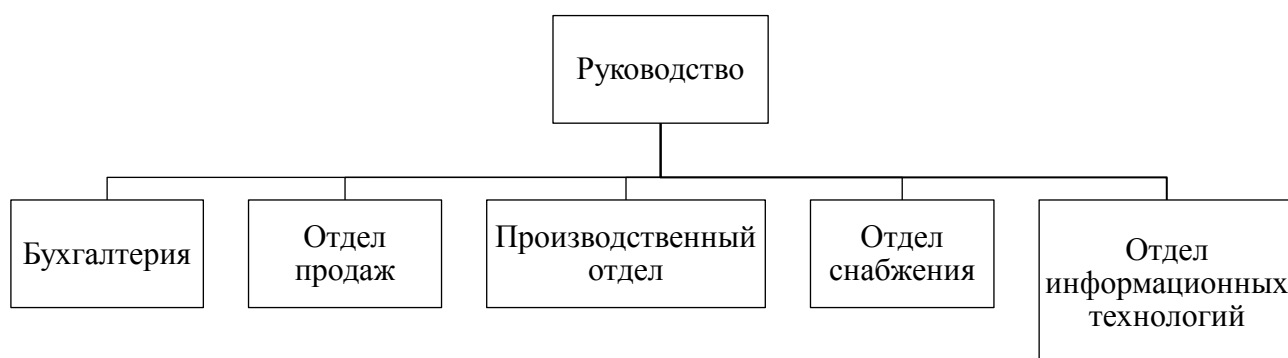


Рисунок 1.1 – Организационная структура ООО «Строй Дизайн»

Руководство принимает решения, осуществляет координацию действий и осуществляет контроль над всей деятельностью организации. Утверждает техническое задание, сроки, а так же бюджет проекта. Это самый главный человек в каждой компании, ответственный за работу каждого сотрудника.

На руководство возложены следующие функции:

- осуществление руководства финансово-хозяйственной деятельности организации;

- обеспечение деятельности организации в рамках законодательства РФ;
- стратегическое планирование всей деятельности организации;
- организация эффективного взаимодействия всех его структурных подразделений.

Бухгалтерия является источником полной, документально обоснованной и структурированной экономической информации, необходимой для принятия управленческих решений в целях обеспечения эффективного хозяйствования.

Основными функциями отдела являются:

- осуществление полного контроля над оформлением документов в срок и согласно законодательству РФ;
- контроль за правильным и экономным расходованием выделенных денежных средств;
- начисление и выплата заработной платы сотрудникам организации;
- участие в проведении инвентаризации денежных средств, расчетов и материальных ценностей, своевременное и правильное определение результатов инвентаризации и отражение их в учете;
- составление и представление в контролирующие органы соответствующей бухгалтерской и статистической отчетности в установленные сроки;
- хранение бухгалтерских документов, смет расходов, других документов, а также сдача их в архив;
- оформление приказов, связанных с деятельностью учреждения.

Отдел продаж – это отдел, главной целью которого является увеличение количества продаж товаров или услуг.

Среди множества задач у отдела продаж есть 4 основные функции:

- поиск и привлечение потенциальных клиентов. Потенциальными клиентами можно считать только тех, кто так или иначе выразил интерес к деятельности организации;

- «первая продажа» клиенту. Её суть заключается, что потенциальных клиентов нужно подвести к заключению сделки. В результате функции могут быть деньги на счете или в кассе от клиента. Так же это может быть или заключенный договор, или часть предоплаты;

- сопровождение клиентов. Сущность функции состоит в регулярном взаимодействии с уже имеющимися клиентами;

- работа с VIP-клиентами. Данная функция касается той части клиентов, с которыми ведется работа на особых условиях. Такими клиентами в основном становятся клиенты с продолжительной истории выгодного сотрудничества. Так же к VIP-клиентам относят тех, кто достигает определенного уровня закупок.

Производственный отдел осуществляет в организации производственное планирование и оперативное управление производством.

Отдел выполняет множество значимых для работы организации функций:

- подготовка производства. (Получение рабочей документации и проверка на соответствие предусмотренных проектом физических объемов ЭМР и спецификации материалов, комплектность пакета документов. Получение разрешений на производство работ и составление его графиков и т. д.);

- своевременная выдача на строительные участки рабочей документации, проектов производства работ, журналов производства работ и всей необходимой документации в соответствии с нормативными требованиями и требованиями заказчиков;

- осуществление технического надзора за строительством;

- сдача объектов в эксплуатацию. (Подготовка документации по завершённым объектам для приемной комиссии);

- работа с субподрядными организациями. (Проведение тендеров, рекомендации по выбору субподрядных организаций, контроль объемов и качества выполненных работ, а так же их графика.);

- прием материальных отчетов, осуществление контрольных обмеров;
- выполнение проектов электроустановок зданий и сооружений при напряжении до 10 кв., на основании договоров с заказчиками;
- рассмотрение, изучение и осуществление электротехнических расчетов, проведение необходимого согласования по выполнению проектного задания согласно информации, полученной от заказчика на проектирование;
- рассчитывает стоимость проектных работ и подготовка документации на оплату. Оформление и передача заказчику готовой проектной документации;
- сохранение контрольных экземпляров выполненных проектов.

Отдел снабжения – это подразделение, деятельность которого направлена на обеспечение необходимыми ресурсами производства. К функциям отдела снабжения можно отнести:

- обеспечение необходимыми материалами на разных этапах строительства;
- хранение и доставка материалов согласно потребностям организации;
- обеспечение строительства всем необходимым оборудованием и инструментами для работы.

Отдел информационных технологий – это подразделение, деятельность которого состоит в создании и сопровождении ИТ–инфраструктуры.

К целям отдела информационных технологий относятся:

- поддержание сети организации в работоспособном состоянии;
- обеспечение бесперебойной работы всего оборудования, к которому относятся компьютеры, принтеры, сканеры, факсы, МФУ, телефоны и многое другое;
- программирование и развитие функциональности программного обеспечения, используемого в организации.

Отдел информационных технологий обязан выполнять следующие

задачи:

- разработка и внедрение проектов совершенствования технологичного управления деятельностью организации;
- минимизация расходов организации за счет внедрения информационных систем;
- выявление и оперативное устранение перебоев в работе оборудования;
- разработка и внедрение программного обеспечения и информационных систем.

1.2 Экономическая сущность задачи

В выпускной квалификационной работе рассматривается транспортировка отходов строительства. Экономической сущностью задачи автоматизации транспортировки отходов является повышение результативности деятельности организации посредством автоматизации деятельности управления транспортировкой отходов[6].

Поскольку процессом автоматизации транспортировки отходов будет заниматься отдел информационных технологий, то следует определить основные функции данного отдела необходимые для автоматизации транспортировки отходов. К ним относятся:

- выбор программного средства для разработки;
- доработка программного продукта под указанные требования для автоматизации деятельности организации;
- установка и настройка программного продукта в строительной организации;
- поддержка правильной работы всей системы сотрудниками отдела информационных технологий.

Путем использования данных функций планируется оптимизировать транспортировку отходов, уменьшая длину пути, и тем самым снизить расходы денег и времени на транспортировку отходов.

1.3. Обоснование необходимости и цели использования вычислительной техники для решения задачи

Ведение учета любой деятельности организации вручную по сравнению с тем, когда используется вычислительная техника, имеет массу недостатков. Одним из недостатков можно назвать обработку больших объемов данных, что является достаточно трудоемким процессом. Над имеющейся информацией проводится множество манипуляций, которые выполняя вручную можно допустить ошибки. С помощью вычислительной техники можно хранить большое количество информации, которой можно легко и быстро управлять. Поэтому использование вычислительной техники пойдет на пользу организации для ведения учета.

Перед тем как перейти к разработке требуется рассмотреть деятельность организации и выявить недостатки в работе.

Описание учета деятельности организации ООО «Строй Дизайн» составлялось при помощи AllFusion Process Modeler 7. В данном программном средстве была создана структурно-функциональная диаграмма («КАК ЕСТЬ») в методологии IDEF0.

AllFusion Process Modeler 7 (ранее BPwin) - мощный инструмент моделирования, используемый при необходимости провести анализ и реструктуризацию происходящих в организации процессов и движения документации. Благодаря детализации процессов, действий, а также необходимых для предприятия ресурсов, можно проанализировать деятельность организации и оценить правильность организации различных

рабочих моментов и движений документации, как по всей организации, так и по конкретным отделам. Разработке программного средства должно предшествовать тщательное изучение протекающих бизнес-процессов.

Деятельность организации можно описать при помощи диаграмм в нотации IDEF0

Методология IDEF0 специализируется на создании функциональной модели, описывающей устройство и опции системы, используемые информационные потоки, а также связанные с этими функциями объекты материальных ценностей. Данная нотация применима для большого круга как автоматизированных, так и не имеющих автоматизации систем.

На рисунке 1.2 изображена контекстная диаграмма деятельности ООО «Строй Дизайн»:

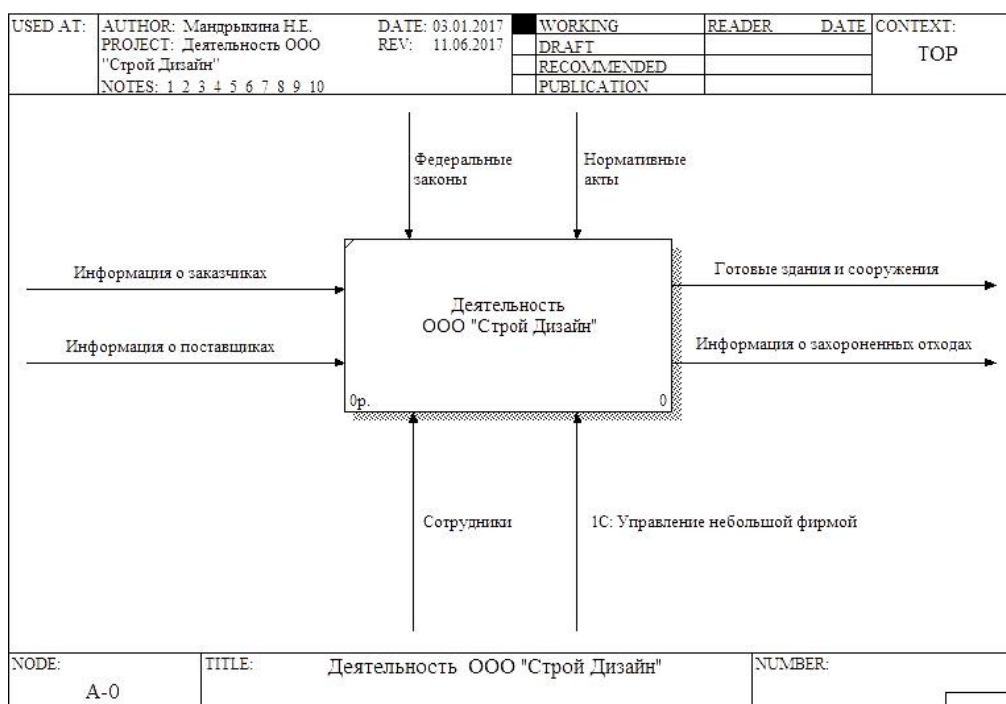


Рисунок 1.2 – Контекстная диаграмма «Деятельность ООО «Строй Дизайн»»

На рисунке 1.3 представлена декомпозиция контекстной диаграммы, где изображены четыре блока: «Оформление заказов клиентов», «Оформление заказчиков у поставщиков», «Строительство зданий и сооружений», «Управление отходами предприятия»:

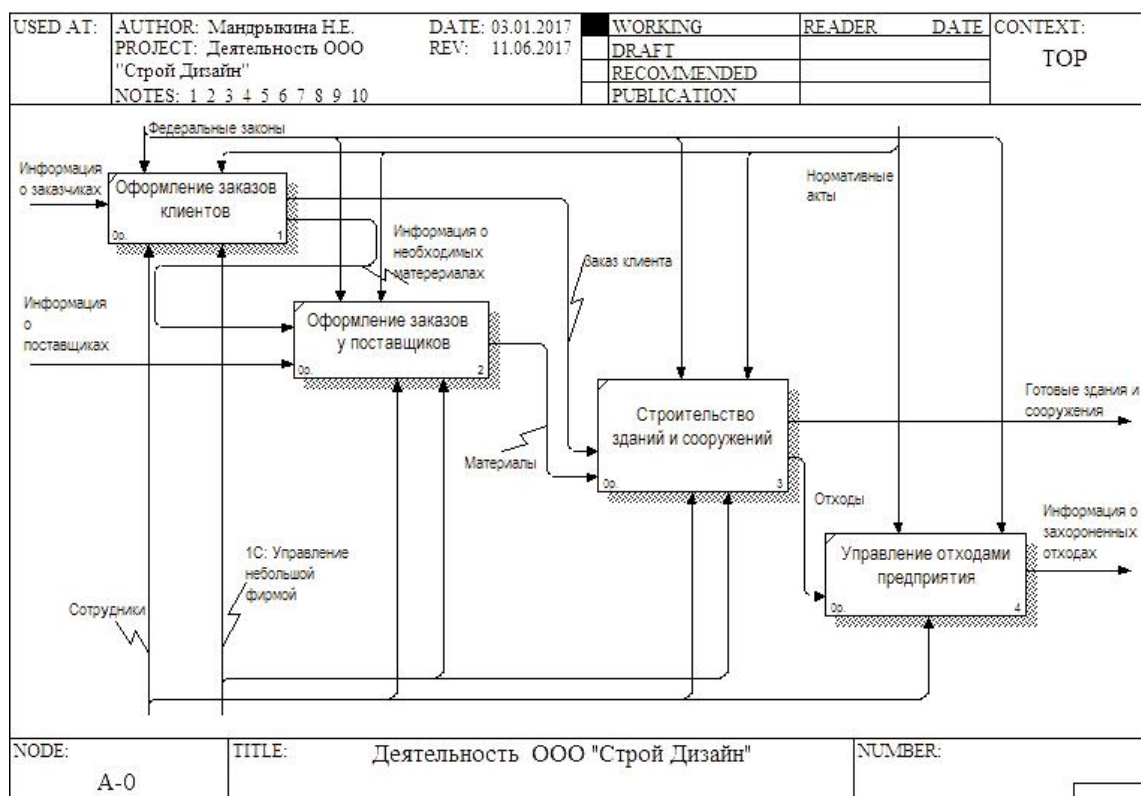


Рисунок 1.3 – Декомпозиция контекстной диаграммы

На структурно-функциональных диаграммах можно заметить, что компьютерная техника при учете деятельности организации используется, но не охватывает все аспекты деятельности. Поэтому необходимость в автоматизации процесса «Управление отходами предприятия» очевидна. Данная разработка должна уменьшению затрат времени и денег. Не стоит забывать, что используя вычислительную технику значительно легче вести учет.

1.4. Постановка задачи

1.4.1 Цель и назначение автоматизированного варианта решения задачи

Целью данной выпускной работы является повышение эффективности деятельности организации за счет снижения расхода времени и денег.

Анализ деятельности организации выявил, что управление отходами ведется только сотрудниками без использования информационных технологий. Для автоматизации данного процесса, были выделены ряд требований и задач:

- ведение справочников виды отходов, маршрут, полигоны, статус строительства, строительные объекты, транспорт, позволяющих хранить, редактировать и использовать информацию;
- ведение учета движения отходов;
- создание и хранения необходимых документов, например, таких как маршрутный лист;
- формирование различной отчетности;
- обработка карт, с целью построения оптимального маршрута.

В итоге реализации требований и задач будет достигнуто:

- увеличение эффективности использования трудовых ресурсов;
- повышение скорости обработки данных;
- увеличение степени защиты информации;
- уменьшение времени транспортировки отходов на полигоны;
- ускорение процесса получения необходимой отчетности.

Можно подвести итог, разработка позволит эффективнее управлять транспортировкой отходов, снизив расход денег и времени.

1.4.2 Общая характеристика организации решения задачи на ЭВМ

Одной из задач организации является транспортировка отходов производства согласно законодательству РФ, причем, необходимо соблюдать правила безопасности.

Как вариант решения поставленной задачи, можно использовать платформу 1С Предприятие 8.3 и прикладное решение 1С: Управление небольшой фирмой.

Модуль предназначен для решения следующих задач:

- регистрация и обработка информации об отходах;
- ведение соответствующей документации;
- учет процессов связанных с транспортировкой отходов;
- составление отчетности.

Требования к функциональным характеристикам

Система должна обеспечивать следующие функции:

1) Ввод, вывод, редактирование, хранение информации иерархического справочника «Видах отходов»:

- код;
- наименование.

2) Ввод, вывод, редактирование, хранение информации справочника «Маршрут»:

- код;
- наименование;
- маршрутный лист.

3) Ввод, вывод, редактирование, хранение информации справочника «Полигоны»:

- код;
- наименование;
- адрес;

- площадь.

4) Ввод, вывод, редактирование, хранение информации справочника

«Статус строительства»:

- код;
- наименование.

5) Ввод, вывод, редактирование, хранение информации справочника

«Строительные объекты»:

- код;
- наименование;
- адрес;
- контрагент;
- статус.

6) Ввод, вывод, редактирование, хранение информации справочника

«Транспорт»:

- код;
- наименование;
- вид транспорта;
- вмещаемый объем.

7) Ввод, вывод, редактирование, хранение и печать документа

«Маршрутный лист»:

- номер;
- дата;
- контрольные точки;
- сроки.

1.5. Анализ существующих разработок и обоснование выбора технологии проектирования

Для анализа была существующих разработок составлена таблица 1.1.

Таблица 1.1 – Информационные системы

№	Существующее программное обеспечение	Описание программного обеспечения
1	2	3
1	1С:Управление строительной организацией	Программный продукт для комплексной автоматизации строительства. Продукт предназначен для полноценной автоматизации различных участников строительного процесса по всем циклам управления: финансами, производством и ресурсами, персоналом и др.
2	1С:Предприятие 8. Бухгалтерия строительной организации	Решение, предназначенное для автоматизации бухгалтерского и налогового учета в малых и средних организациях, осуществляющих любые виды строительной деятельности, а также оптовую и розничную торговлю, оказание услуг, производство и т.д.
3	1С:Предприятие 8. TMS Логистика. Управление перевозками	Решение предназначено для автоматизированного управления бизнес-процессами отдела транспортной логистики предприятия. Система предоставляет возможности управления процессом перевозки товарно-материальных ценностей (ТМЦ) по цепи «поставщик — склад — клиент». Отличительной чертой программы является легкость и простота адаптации к условиям работы практически любого предприятия, специфике его технологических и организационных требований.
4	KiberLog	Облачный сервис для транспортных компаний и логистической деятельности. Онлайн-система управления бизнесом в сфере грузоперевозок. Цель системы - обеспечить своевременный обмен информацией, SaaS документооборот между заказчиками, грузоперевозчиками и транспортно-экспедиционными компаниями в едином информационном пространстве.

Продолжение таблицы 1.1

1	2	3
5	БИТ: Автотранспорт	Система на базе 1С:Предприятие 8 комплексно автоматизирует все бизнес-процессы управления автотранспортом на предприятиях любого масштаба и отрасли.
6	1С:Управление автотранспортом	Система автоматизации управленческого и оперативного учета в автотранспортных предприятиях, а также в транспортных подразделениях торговых, производственных и других компаний с различной отраслевой спецификой.
7	Галактика Управление капитальным строительством	Решение комплексного управления проектами в сфере строительства, мониторинга хода выполнения работ, предназначенное для строительных дирекций крупных корпораций и холдингов
8	Галактика Управление логистикой	Предназначен для эффективного управления материальными и связанными с ними информационными и финансовыми потоками предприятия и позволяет организовать совместную деятельность подразделений предприятия для эффективного продвижения продукции по цепи «закупка сырья – производство продукции – сбыт».
9	Корс Автопредприятие	Профессиональная программа учета автотранспорта, путевых листов, расхода топлива и смазочных материалов, составления маршрутов движения транспортных средств на автотранспортных предприятиях. Типовые формы путевых листов для различных видов автотранспорта.
10	Галактика Управление строительством	Решение предназначено для комплексного управления компанией строительной сферы и строительными проектами и полностью учитывает специфику бизнес-процессов строительной деятельности.

Проанализировав данные системы можно выделить основные положительные и отрицательные характеристики системы (плюсы и минусы использования той или иной системы).

- 1) 1С:Управление строительной организацией.

Положительная сторона: автоматизирует практически все аспекты деятельности строительной организации.

Отрицательная сторона: не рассмотрен аспект «управления отходами», в том числе не реализована транспортная логистика по перевозке отходов в место захоронения с разных объектов(строительства) организации.

2) 1С:Предприятие 8. Бухгалтерия строительной организации.

Положительная сторона: хорошо автоматизирована бухгалтерия, ориентированная под данную сферу.

Отрицательная сторона: не рассмотрены многие аспекты, в том числе аспект, который планируется реализовать в ВКР.

3) 1С:Предприятие 8. TMS Логистика. Управление перевозками.

Положительная сторона: хорошо реализован аспект, который планируется реализовать в ВКР.

Отрицательная сторона: не учитывает все особенности строительных организаций и их остальные аспекты работы.

4) KiberLog.

Положительная сторона: хорошо реализован аспект, который планируется реализовать в ВКР.

Отрицательная сторона: не учитывает все особенности строительных организаций и их остальные аспекты работы.

5) БИТ:Автотранспорт.

Положительная сторона: хорошо реализован аспект, который планируется реализовать в ВКР.

Отрицательная сторона: не учитывает все особенности строительных организаций и их остальные аспекты работы.

6) 1С:Управление автотранспортом.

Положительная сторона: хорошо реализован аспект, который планируется реализовать в ВКР.

Отрицательная сторона: не учитывает все особенности строительных организаций и их остальные аспекты работы.

7) Галактика Управление капитальным строительством.

Положительная сторона: автоматизирует многие аспекты деятельности строительной организации.

Отрицательная сторона: не рассмотрен аспект «управления отходами», в том числе не реализована транспортная логистика по перевозке отходов в место захоронения с разных объектов(строительства) организации.

8) Галактика Управление логистикой.

Положительная сторона: хорошо реализован аспект, который планируется реализовать в ВКР.

Отрицательная сторона: не учитывает все особенности строительных организаций и их остальные аспекты работы.

9) Корс Автопредприятие.

Положительная сторона: хорошо реализован аспект, который планируется реализовать в ВКР.

Отрицательная сторона: не учитывает все особенности строительных организаций и их остальные аспекты работы.

10) Галактика Управление строительством .

Положительная сторона: автоматизирует многие аспекты деятельности строительной организации.

Отрицательная сторона: не рассмотрен аспект «управления отходами», в том числе не реализована транспортная логистика по перевозке отходов в место захоронения с разных объектов(строительства) организации.

Рассмотрев имеющиеся на рынке системы видно, что представленные системы или не учитывают основные аспекты строительных организаций или же отсутствует транспортная логистика для оптимального управления отходами производства. Разработка должна совместить основные преимущества.

2 Проектная часть

2.1 Обоснование проектных решений

2.1.1 Обоснование проектных решений по техническому обеспечению (ТО)

Техническое обеспечение это целый комплекс взаимосвязанных технических средств, предназначенная для функционирования информационной системы, а также сопутствующая документация по этим средствам и технологические процессы[8].

Комплекс технических средств включает:

- электронно-вычислительные машины любых моделей;
- устройства сбора, накопления, обработки, передачи и вывода информации;
- устройства передачи данных и линий связи;
- оргтехника и устройства автоматического съема информации;
- эксплуатационные материалы.

При выборе типа компьютера имеют значение различные факторы. Можно представить такие факторы как объем памяти, надежность, производительность, так же стоимость и другие.

В настоящее время компьютеры можно разделить на супер, большие, мини и микро электронно-вычислительные машины (ЭВМ). СуперЭВМ и большие ЭВМ имеют достаточно высокое быстродействие. Такие типы ЭВМ предназначены для решения сложных и объемных задач.

Мини-ЭВМ являются промежуточным звеном между микро- и большими ЭВМ. Эти компьютеры могут использовать, например, большие организации, научные учреждения. Представители класса микро-ЭВМ - это персональные компьютеры, которые предназначены для личного

пользования. Такие компьютеры предоставляют возможность повысить производительность интеллектуального труда для различного профиля специалистов. Персональные компьютеры предназначены для автономной работы с пользователем при использовании диалогового режима.

Для корректной работы модуля и самой информационной системы достаточно имеющегося в организации персонального компьютера со следующими характеристиками:

- процессор AMD Athlon 64 FX-51
- материнская плата ASUS P4C800 (i875P);
- видеокарта NVIDIA GeForce FX 5900 Ultra;
- сетевая карта ACorp L-1000S;
- оперативная память Kingston KVR13N9S6/2 DDR3;
- жесткий диск Western Digital WD400JB;
- ЖК-монитор Acer eb222Qb
- мышь проводная Defender Orion 400;
- клавиатура Sven Standard 309M.

В данном случае нет никакой необходимости закупать новое оборудование, поэтому и затраты на внедрение разрабатываемого модуля существенно снизятся.

2.1.2 Обоснование проектных решений по информационному обеспечению (ИО)

Информационное обеспечение описывает общность системы кодирования и классификации различной информации, документации, которые могут быть также представлены схемами потоков информации, обращающихся и работающих в организации.

Для решения поставленных задач модуля управления транспортировкой отходов выбор пал на платформу 1С: Предприятие 8.3.

В данной программной среде можно реализовать множество различных операций и функций необходимых для данной организации. С ее помощью можно обработать и упорядочить большое количество информации, в том числе и об отходах организации.

Система позволяет хранить, перемещать и обрабатывать данные, облегчающие учетную деятельность организации.

Обоснование проектных решений по информационному обеспечению определяется путем внутримашинного (входные, выходные и промежуточные данные) и немашинного (документы и справочники) обеспечения.

Подобным образом были разработаны объекты модуля:

- 1) Справочники:
 - а) Виды отходов;
 - б) Полигоны;
 - в) Строительные объекты;
 - г) Транспорт;
 - д) Маршрут.
- 2) Документы:
 - а) Маршрутный лист;
 - б) Заявка на транспортировку отходов;
 - в) Движение отходов.
- 3) Отчеты:
 - а) Отчет о движении отходов;
 - б) Отчет по заявкам на транспортировку отходов.

Для хранения таблиц и индексов базы данных система 1С:Предприятие использует файлы в формате ELF и CD, которые находятся в каталоге информационной базы системы.

Любой документ в информационной системе можно вывести на печать в виде отчета на экран, а также через принтер.

2.1.3 Обоснование проектных решений по программному обеспечению (ПО)

Программное обеспечение это совокупность программ и документации по их эксплуатации. К программному обеспечению так можно отнести область деятельности по проектированию и разработке ПО, которая включает в себя:

- анализ качества работы программ;
- технологию проектирования программ;
- методы тестирования и отладки программ;
- методы доказательства правильности программ;
- документирование программ;
- разработка и использование программных средств, для автоматизации процесса проектирования программного обеспечения, и многое другое.

Программное обеспечение представляет собой часть компьютерной системы в целом, являющееся логическим продолжением технических средств. Область применения конкретного компьютера, определяется используемым в нем программным обеспечением. Компьютер по сути это набор технических устройств в одном корпусе. Все нужные знания, для той или иной области собраны в выполняемых на компьютерах программах.

В настоящее время программное обеспечение для персональных компьютеров включает миллионы программ. Для реализации поставленных перед выпускной квалификационной работой задач, можно использовать любую современную СУБД. Для удобства взаимодействия пользователя с системой необходимо спроектировать и разработать интерфейс системы[7].

Чтобы база данных функционировала эффективно необходимо правильно разработать ее структуру, правильно определить логические взаимосвязи между ее сущностями. Это приводит к выводу, что для разработки

базы данных и программного продукта, где она используется, затрачивалось много времени.

Для решения проблемы управления данными, нужно точно определить, все данные необходимые для подсистемы, образ их хранения в памяти компьютера и способ организации системы доступа. Помимо этого требуется выбрать способы решения предстоящих типовых задач с помощью возможностей компьютера.

Подводя итог всему рассмотренному ранее, можно сделать вывод, что для оперативного решения поставленных задач необходимы мощная реляционная СУБД и система разработки приложений. Такой системой и является 1С:Предприятие 8.3. Система 1С:Предприятие 8.3 включает в себя две составляющие:

- технологическую платформу;
- прикладные решения автоматизации различных типов деятельности, которые создаются с помощью технологической платформы.

Такая организация позволяет с легкостью адаптировать прикладные решения под конкретную сферу деятельности. Это позволяет разработчику, который даже если он и не участвовал в создании прикладного решения, использовать все требуемые ресурсы для внесения изменений в конфигурацию. Открытость архитектуры системы, показывает, что 1С:Предприятие очень удобной для программистов любого уровня.

Даже если в организации всего один программист, это не мешает, если необходимо настроить большинство стандартных прикладных решений под свою деятельность. Прикладное решение может модифицироваться без использования каких-либо других программных продуктов - все ресурсы для разработки уже входят в состав технологической платформы.

В технологической платформе выделяются две составляющие:

- Среда исполнения;
- Среда разработки.

Среда исполнения принято называть режимом работы 1С:Предприятия, а среду разработки режимом конфигурирования.

В среде разработки используется так называемая технология метаданных. Метаданные это совокупность различных объектов, которые описывают конкретное прикладное решение. Среда исполнения графически представляет метаданные.

При работе с метаданными, разработчику доступно для использования огромное количество визуальных средств разработки. В результате чего программисту нет необходимости писать программный код для добавления нового объекта метаданных. Основной процесс программирования, состоит в щелчках мыши и настройке необходимых свойств и параметров объектов. В результате выполнения всех этих действий, система автоматически генерирует программный код.

Так же используется встроенный язык программирования 1С:Предприятие, а также язык запросов используются для описания специфических алгоритмов прикладного решения. Язык программирования 1С:Предприятие имеет много общих черт с другими языками программирования, такими как Visual Basic, JavaScript, C\C++ и другие. Для построения запросов в систему 1С:Предприятие включен - язык запросов. Язык запросов в системе 1С:Предприятие отличается от остальных невозможностью изменения данных в запросе, поскольку доступ к данным возможен только для чтения. Чтобы записать информацию в базу используются конструкции встроенного языка 1С:Предприятие.

Моделирование бизнес процессов ООО «Строй Дизайн» выполнялось с помощью CASE средства AllFusion Process Modeler 7 (ранее известное как BPwin). Данное программной средство предназначено для моделирования, анализа, документирования и последующей оптимизации бизнес-процессов. Программа позволяет наглядно пронаблюдать все особенности в работе организации.

AllFusion Process Modeler 7 позволяет максимально точно отразить бизнес-процессы в результате чего повышается эффективность ИТ-решений. Используя данное средство можно получить целостную картину деятельности предприятия.

2.1.4 Обоснование проектных решений по технологическому обеспечению

Повсеместное использование вычислительной техники во всех отраслях человеческой деятельности остро ставит вопрос о технологическом обеспечении информационных систем и технологий.

Технологическое обеспечение реализует информационные процессы в информационных системах управления деятельностью организации с помощью ЭВМ и других технических средств.

Разработка технологического обеспечения требует учета особенностей структуры системы организации. Прежде всего - это сложность организационного взаимодействия, которое вызывает необходимость создания многоуровневых иерархических систем со сложными информационными связями.

В состав технологического обеспечения входят:

- техническое обеспечение;
- информационное обеспечение;
- математическое обеспечение;
- программное обеспечение;
- лингвистическое обеспечение;
- организационное обеспечение;
- методическое обеспечение;
- эргономическое обеспечение;

– правовое обеспечение.

Эффективность проектирования технологического процесса представляют гарантию снижения трудовых, стоимостных затрат в будущем.

Технологический процесс состоит из нескольких этапов. Первым этапом является регистрация, сбор, передача данных для дальнейшей обработки. Результатом при этом является ввод документа. Целью второго этапа является первоначальное формирование информационной базы и перенос данных на машинные носители. Третий этап состоит из операций сортировки, обработки, накопления и корректировки данных.

Вариант технологического процесса выбирается, учитывая нижеперечисленные требования:

- 1) возможность решения задачи с использованием различных режимов;
- 2) решение задач в определенные сроки;
- 3) минимальные стоимостные и трудовые затраты на обработку информации;
- 4) обеспечение достоверности обрабатываемых данных;
- 5) наличие возможности обработки данных на ЭВМ;
- 6) Соблюдая выше описанные требования, необходимо спроектировать модуль управления, позволяющего вести деятельность организации более эффективно.

2.1.5 Обоснование выбора программных средств

Программным средством называют компьютерные программы, процедуры и, возможно, соответствующая документация и данные, относящиеся к функционированию компьютерной системы.

Во время разработки модуля управления транспортировкой отходами в ООО «Строй Дизайн» были использованы следующие программные средства:

а) технологическая платформа 1С:Предприятие 8.3 и конфигурация Управление небольшой фирмой;

б) CASE средство AllFusion Process Modeler 7

Платформа 1С:Предприятие 8.3 разработана для автоматизации работы на предприятии. Благодаря большому количеству преимуществ этот комплекс стал чрезвычайно востребованным для множества фирм, работающих в различных сегментах[5].

Основные особенности платформы 1С:Предприятие 8.3:

- автоматизация работы предприятия любого размера. Данная среда разработки дает возможность оптимизировать фирмы с любым количеством рабочих мест.

- платформа имеет возможность функционировать в операционных системах Windows, Linux, FreeBSD.

- возможность обновления через Интернет. Платформа 1С:Предприятие все время совершенствуется, с момента выпуска самой первой версии программы.

- 1С:Предприятие 8.3 дает клиенту наилучшие решения для автоматизации бизнеса. Данная среда практически не имеет конкурентов, поскольку разрабатываемые приложения способны удовлетворить все требования и особенности работы клиентов.

Преимущества платформы 1С:Предприятие 8.3. состоят в следующем:

1) Эргономика интерфейса.

В 1С:Предприятии 8.3 реализован современный дизайн интерфейса и повышена комфортность работы пользователей при работе с системой в течение длительного времени. Интерфейс системы спроектирован с учетом необходимости массового ввода информации, а также с учетом работы менее опытных пользователей.

2) Масштабируемость.

Масштабируемостью называют способность системы адаптироваться к расширению предъявляемых требований и возрастанию объемов решаемых задач. Система 1С:Предприятие 8,3 имеет хорошие возможности масштабирования. Она позволяет работать как в файловом варианте, так и с использованием технологии "клиент-сервер".

3) Подключение через Интернет.

Подключение через Интернет позволяет обеспечить удаленную on-line работу пользователей с информационными базами. Это возможно благодаря использованию тонкого клиента и веб-клиента. Они подключаются к специальному образом настроенному веб-серверу, который осуществляет их взаимодействие с кластером или с файловой информационной базой.

4) Отказоустойчивость.

Отказоустойчивость системы обеспечивается при работе в клиент-серверном варианте с использованием кластера серверов. Система обеспечивает бесперебойную работу пользователей при программных и аппаратных сбоях в кластере серверов.

5) Облачные технологии.

Облачные технологии 1С:Предприятия обеспечивают повсеместную и удобную работу с прикладными решениями на различных клиентских устройствах с различными операционными системами.

При этом клиенты могут использовать устройства с малой вычислительной мощностью, мобильные устройства. Клиенты могут вообще не устанавливать на свои устройства программное обеспечение 1С:Предприятия, или могут установить легкое клиентское приложение, не требующее большого количества ресурсов[4].

Все основные вычисления, прикладная логика реализуется в кластере серверов 1С:Предприятия, который обеспечивает масштабируемость, отказоустойчивость, динамическое перераспределение нагрузки и взаимодействие с СУБД, хранящими данные прикладных решений. При

необходимости кластер серверов может быть усилен инфраструктурой сервиса, которая позволяет предоставлять клиентам услуги пользования программным обеспечением как сервисом, вести учет потребления этих услуг, осуществлять общее администрирование сервиса и др.

Термин «облако» используется в обозначении этих технологий как метафора. Она основана на том, что интернет изображается на компьютерных диаграммах в виде облака, за которым от клиента скрывается сложная инфраструктура и все технические детали.

б) Многоплатформенность.

Многоплатформенность - это способность системы работать под управлением различных операционных систем. Основные компоненты системы могут работать как под управлением операционной системы Windows, так и под управлением операционной системы Linux. Кроме этого клиентская часть 1С:Предприятия может быть запущена и на компьютерах с операционными системами OS X и Apple iOS.

7) Экономическая и аналитическая отчетность, механизмы

Механизмы, предназначенные для формирования экономической и аналитической отчетности, представляют собой комплекс средств, позволяющих формировать не просто печатные формы, а интерактивные документы, тесно интегрированные в прикладное решение. Пользователь может не только распечатать отчет, но и работать с ним практически так же, как с любой экранной формой - изменять параметры отчета, перестраивать его, использовать "расшифровки" - возможность формирования дополнительных отчетов на основе отдельных элементов уже сформированного отчета и т.д.

AllFusion Process Modeler представляет собой мощный инструмент моделирования, который используется для анализа, документирования и реорганизации сложных бизнес-процессов. На данный момент времени Process Modeler соответствует требованиям бизнеса, так как позволяет четко документировать различные аспекты деятельности - действия, которые необходимо предпринять, способы их осуществления, требующиеся для этого

ресурсы. Process Modeler дает возможность сформировать целостную картину деятельности предприятия - от моделей организации работы в маленьких отделах до сложных иерархических структу.

Модели, спроектированные в Process Modeler дают основу для осмысления бизнес-процессов и оценки влияния тех или иных событий, а также дает описание взаимодействия процессов и потоков информации в организации. Неэффективная, высоко затратная или избыточная деятельность может проявиться и, следовательно, имеется возможность усовершенствования, изменения или устранения в соответствии с основными целями организации.

AllFusion Process Modeler позволяет адаптироваться к постоянно меняющимся реалиям современного рынка. Для выживания организации на рынке необходимо иметь мгновенную реакцию на новые возможности, угрозы и потребности покупателей. В настоящее время постоянные изменения никого не удивят. Поскольку бизнес-процессы становятся все более сложными, требуются решения, представляющие интегрированный взгляд на функционирование компании.

AllFusion Process Modeler позволяет управлять сложными бизнес-процессами. Модели бизнес-процессов в масштабах всего предприятия могут оказаться достаточно сложными. AllFusion Process Modeler предоставляет возможности, призванные облегчить инкрементальную разработку моделей и разграничение процессов. Иногда возникает необходимость более детально изучить определенную часть общей модели. Process Modeler позволяет разбить модель на фрагменты, поработать с ними, а затем вновь объединить их в одно целое.

Анализ бизнеса с различных сторон: поддержка нотаций IDEF0, IDEF3 и DFD. AllFusion Process Modeler совмещает в одном инструменте средства моделирования функций (IDEF0), потоков данных (DFD) и потоков работ (IDEF3), координируя эти три основных аспекта бизнеса для соответствия потребностям бизнес-аналитиков и системных аналитиков. Имеется

возможность повторно использовать основополагающую информацию моделирования.

Функциональное моделирование (нотация IDEF0), дает возможность провести систематический анализ бизнеса, сосредоточившись на регулярно решаемых задачах (функциях), свидетельствующих об их правильном выполнении показателях, необходимых для этого ресурсах, результатах и исходных материалах (сырье).

Моделирование потоков данных (DFD), частенько используется при разработке программного обеспечения, сосредоточено вокруг потоков данных, передающихся между различными операциями, включая их хранение, для достижения максимальной доступности и минимального времени ответа. Такое моделирование позволяет рассмотреть конкретный процесс, проанализировать операции, из которых он состоит, а также точки принятия решений, влияющих на его ход.

Моделирование потоков работ (нотация IDEF3) позволяет рассмотреть конкретный процесс, проанализировать операции, из которых он состоит, а также точки принятия решений, влияющих на его ход.

2.2 Информационное обеспечение задачи (комплекса задач, АРМ)

2.2.1 Информационная модель и ее описание

В данной выпускной квалификационной работе было необходимо разработать модуль управления отходами. Разрабатываемый модуль должен повысить эффективность деятельности организации за счет решения проблемы с транспортировкой отходов[10].

Исходя из вышеперечисленных требований, которые были описаны ранее, была построена структурно-функциональная модель (рисунок 2.1).

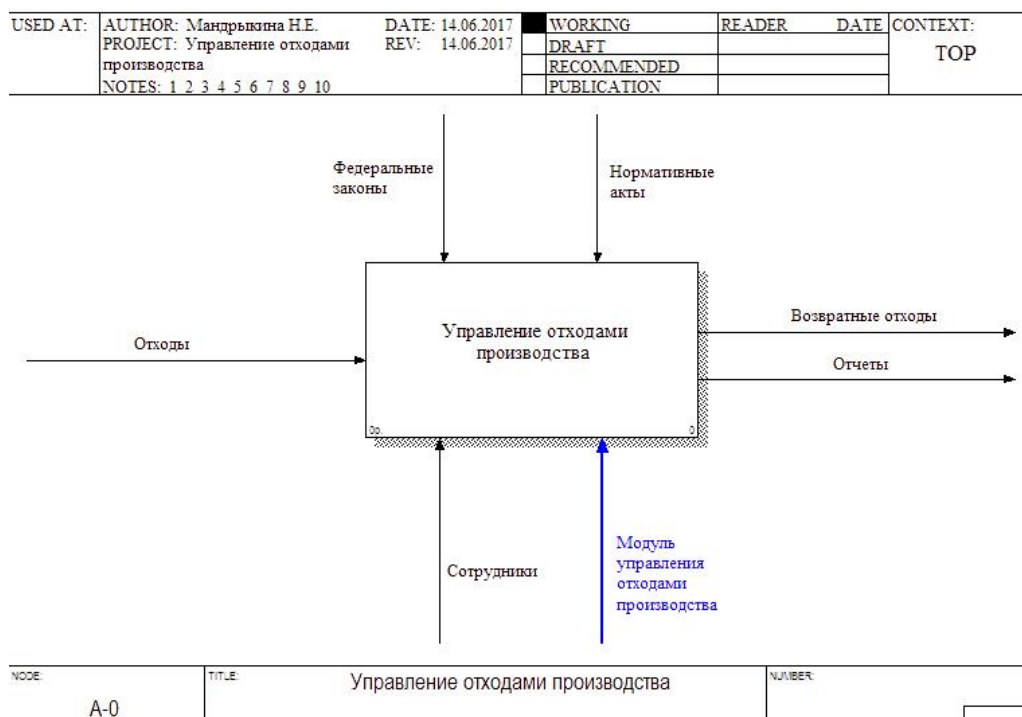


Рисунок 2.1 - Контекстная модель «Управление отходами производства»

Входной информацией будут являться отходы.

С помощью среды разработки 1С:Предприятие 8.3 будет осуществляться разработка модуля управления отходами производства в строительстве на базе 1С: Управление небольшой фирмой. Вся деятельность организации выполняется согласно нормативным актам и федеральным законам. Выходной информацией же являются возвратные отходы и отчеты[15].

В свою очередь процесс управления отходами можно разделить на 3 раздела:

- Сортировка отходов;
- Захоронение отходов;
- Составление отчетности.

Описанные выше разделы и движения потоков между ними представлены на диаграмме декомпозиции (рисунок 2.2).

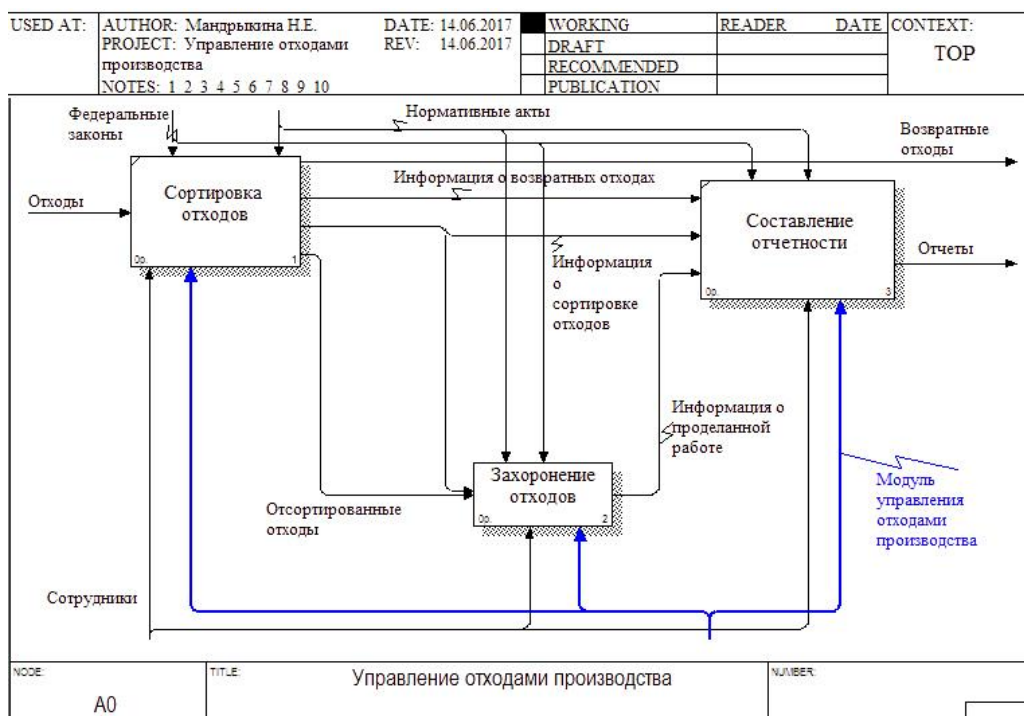


Рисунок 2.2 – Диаграмма декомпозиции «Управление отходами производства»

Стоит понимать, что делает каждый блок.

Сортировка отходов – отвечает за корректное разделение отходов, дабы при транспортировке можно было учитывать, какие условия транспортировки требуется соблюдать.

Составление отчетности – отвечает за составляемую отчетную информацию, которая должна четко и достоверно отразить реальную ситуацию[9].

Захоронение отходов – отвечает за транспортировку отходов. Эта часть работы описывает более эффективный способ транспортировки отходов и их захоронения (рисунок 2.3).

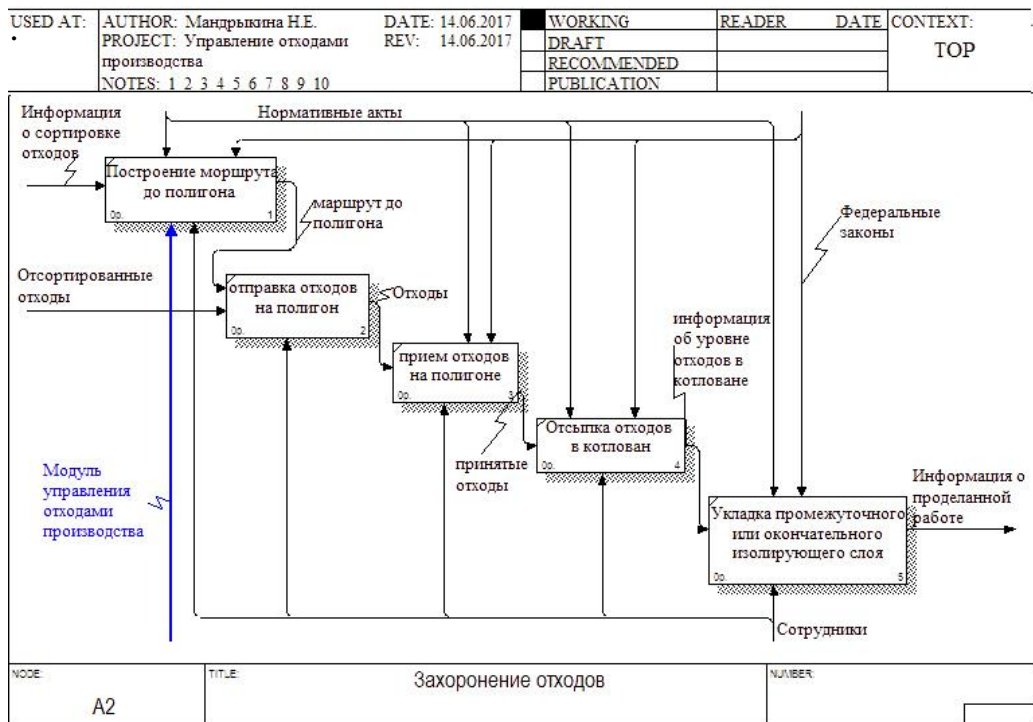


Рисунок 2.3 – Диаграмма декомпозиции «Захоронение отходов»

3 Программная реализация проектных решений

3.1 Программное обеспечение комплекса задач

3.1.1 Обеспечение хранения условно-постоянной информации

Одно из основных требований, поставленных перед модулем, это то, что вся информация, касающаяся отходов, хранилась в базе данных как можно более в полном и развернутом виде. Для этого в систему были добавлен ряд справочников, для хранения условно-постоянной информации.[11] Перечень разработанных справочников конфигурации показаны в таблице 3.1.

Таблица 3.1 – Справочники модуля управления отходами производства

Наименование справочника	Длина и тип кода	Иерархия	Назначение	Примеры
1	2	3	4	5
Виды Отходов	Строка, 25 знаков	Да	Хранение условно-постоянной информации о видах отходов	Наименование: Битое стекло
Статус Строительства	Строка, 25 знаков	Нет	Хранение условно-постоянной информации о статусе строительства	Наименование: Завершен
Маршрут	Строка, 50 знаков	Нет	Хранение условно-постоянной информации о маршруте	Наименование: Маршрут Юго-западного района

Продолжение таблицы 3.1

1	2	3	4	5
Полигоны	Строка, 50 знаков	Нет	Хранение условно-постоянной информации о полигоне	Наименование: Полигон №1, адрес: г. Белгород, пр-т. Богдана Хмельницкого 123, площадь: 100
Строительные объекты	Строка, 50 знаков	Нет	Хранение условно-постоянной информации о строительном объекте	Наименование: Объект №1 Школа № 17, адрес: г. Белгород, пр-т. Богдана Хмельницкого 103, контрагент: МБОУ СОШ №17, статус: Завершен
Транспорт	Строка, 50 знаков	Нет	Хранение условно-постоянной информации о транспортных средствах	Наименование: КАМАЗ, вид транспорта: грузовик, вместаемый объем: 15
Маршрут	Строка, 50 знаков	Нет	Хранение условно-постоянной информации о строительном объекте	Дата: 17.05.2017, контрольные точки: Объект №1, дата начала: 1 7.05.2017, дата окончания: 19.05.17

В таблице 3.1 приведен полный список справочников модуля. Создание справочника имеет свою специфику. Как именно создаются справочники можно на нескольких примерах[2].

Создание справочника «Маршрут» как и любого другого справочника, происходит с помощью встроенный мастер создания, который представлен на рисунке 3.1

Рисунок 3.1 – Мастер создания объекта конфигурации «Справочник»

Справочник «Маршрут» предназначен для хранения, редактирования и использования информации о маршрутах. Данные любого справочника можно разделить на реквизиты и табличные части. Структура данных справочника «Маршрут» представлена следующим образом:

- 1) Реквизиты:
 - а) номер;
 - б) дата начала;
 - в) дата окончания.
- 2) Табличная часть:
 - а) контрольные точки.

В системе 1С: Предприятие 8.3 имеется возможность использовать несколько форм объектов конфигурации. При помощи встроенного мастера создания форм, для справочника были разработаны формы элемента и списка. Форма списка отображает все записи справочника, а форма элемента

используется для ввода информации о маршрутном листе и ее редактировании.

Формы справочника «Маршрут» представлены на рисунках 3.2 и 3.3

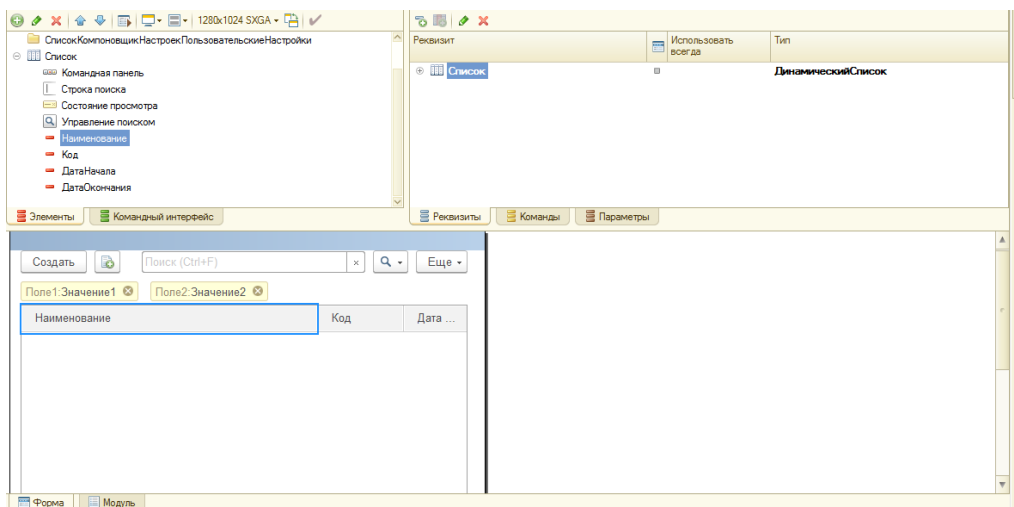


Рисунок 3.2 – Форма элемента справочника «Маршрут»

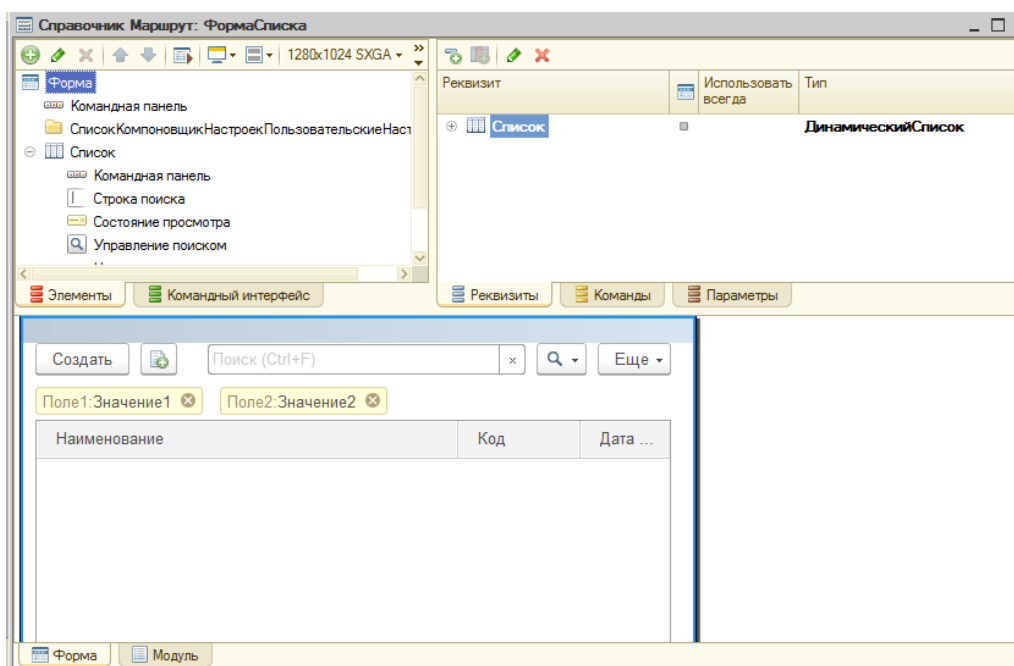


Рисунок 3.3 – Форма списка справочника «Маршрут»

Справочник «Виды отходов» предназначен для хранения, редактирования и использования информации о видах отходов. Справочник является иерархическим и имеет только стандартные реквизиты. Для данного справочника так же были разработаны форма элемента, списка и выбора, рисунки 3.4 и 3.5

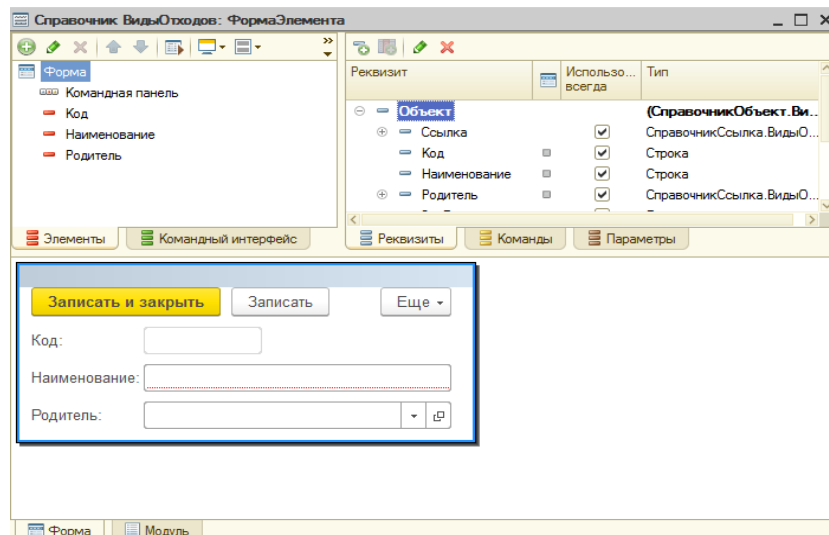


Рисунок 3.4 – Форма элемента справочника «Виды Отходов»

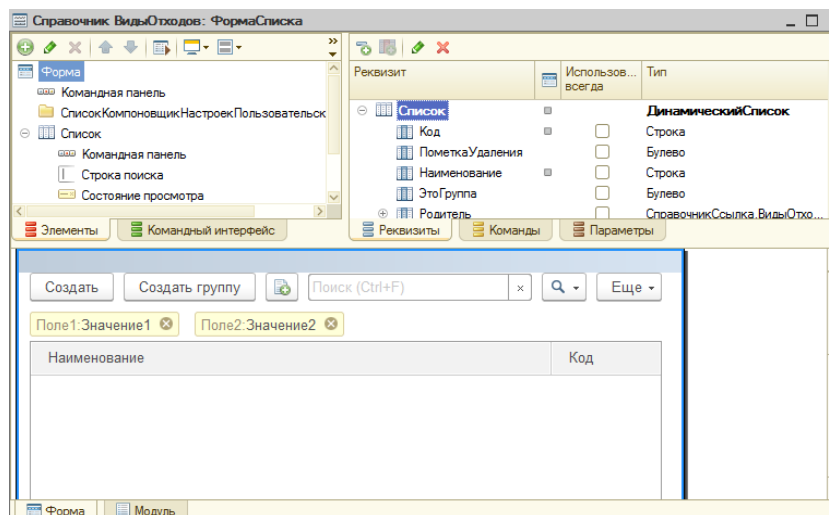


Рисунок 3.5 – Форма списка справочника «Виды Отходов»

По аналогии были созданы и остальные справочники модуля. Для разработчика меняется набор реквизитов, а набор форм не изменен в разрезе данной разработки. Поэтому рассматривать создание остальных справочников не имеет смысла.

Помимо этих двух справочников в модуле созданы и другие.

Справочник «Полигоны» предназначен для хранения, редактирования и использования информации о полигонах. Полигоном называется специально оборудованная площадь для сбора и хранения отходов.

Структура данных справочника «Полигоны» представлена следующим образом:

- 1) Реквизиты:
 - а) адрес;
 - б) общая площадь;
 - в) свободная площадь.

Справочник «Состояние заказов на транспортировку» предназначен для хранения, редактирования и использования информации о состоянии заказов на транспортировку. Структура данных справочника «Состояние заказов на транспортировку» состоит только из стандартных реквизитов: код и наименование.

Справочник «Статус строительства» предназначен для хранения, редактирования и использования информации о статусе строительства. Структура справочника так же состоит только из стандартных отводов.

Справочник «Строительные объекты» предназначен для хранения, редактирования и использования информации о строительных объектах, как текущих, так и завершенных.

Структура данных справочника «Строительные объекты» представлена следующим образом:

- 1) Реквизиты:
 - а) Адрес;
 - б) Контрагент;
 - в) Статус.

Справочник «Транспорт» предназначен для хранения, редактирования и использования информации о транспортных средствах.

Структура данных справочника «Транспорт» представлена следующим образом:

- 1) Реквизиты:
 - а) Тип транспорта;
 - б) Вместаемый объем.

3.1.2 Отражение основных операций в информационной системе

Учет операций в программе реализуется с помощью объекта «Документ». Для корректной работы программы требуется соблюдать последовательность создания объектов конфигурации типа документ и регистрации их в системе[2].

Для достижения поставленных перед модулем задач создавался документ «Заказ на транспортировку», он служит для регистрации заказа на транспортировку. Структура данного документа представлена с помощью следующих реквизитов:

- автор;
- тип транспортировки;
- документ основание;
- заметки;
- комментарий;
- организация;
- ответственный;
- состояние заказа;
- статус;
- старт;
- финиш.
- вид рекламы;
- контрагент;
- дополнительно;
- статус заказа;
- груз (табличная часть).

Для наиболее удобного введения, хранения и использования создаваемых и регистрируемых документов разработаны форма документа и форма списка (рисунки 3.6, 3.7).

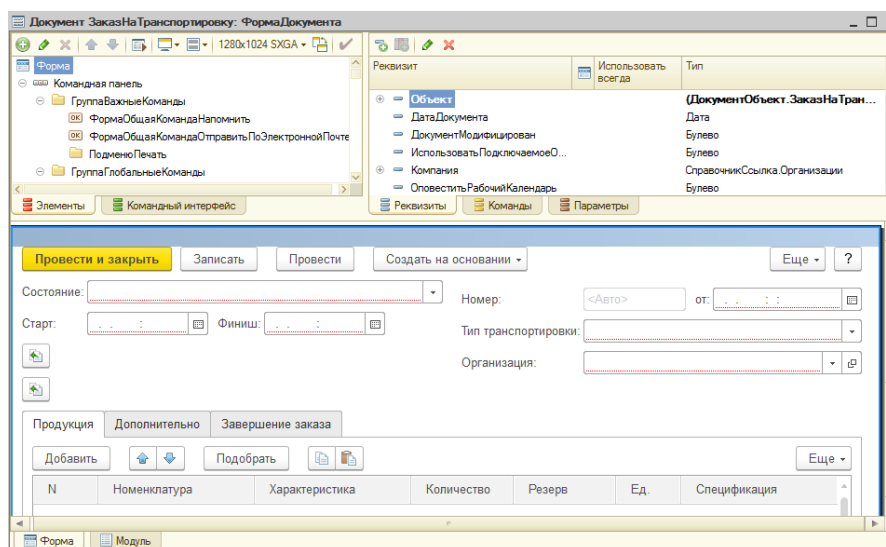


Рисунок 3.6 – Форма документа «Заказ на транспортировку»

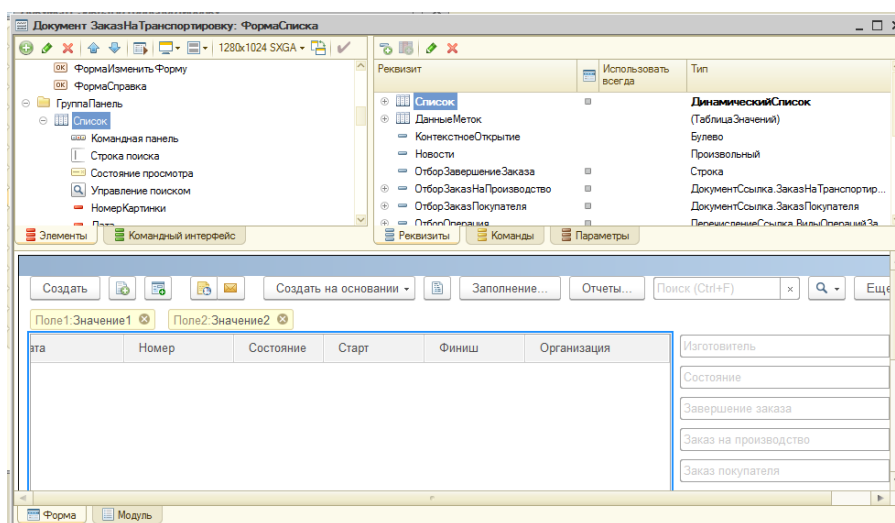


Рисунок 3.7 – Форма списка документа «Заказ на транспортировку»

После заполнения всех реквизитов документа его необходимо распечатать и передать на бумажном носителе[6]. Для этого, была создана печатная форма документа: На рисунке 3.8 представлен макет печати документа «Заказ на транспортировку».

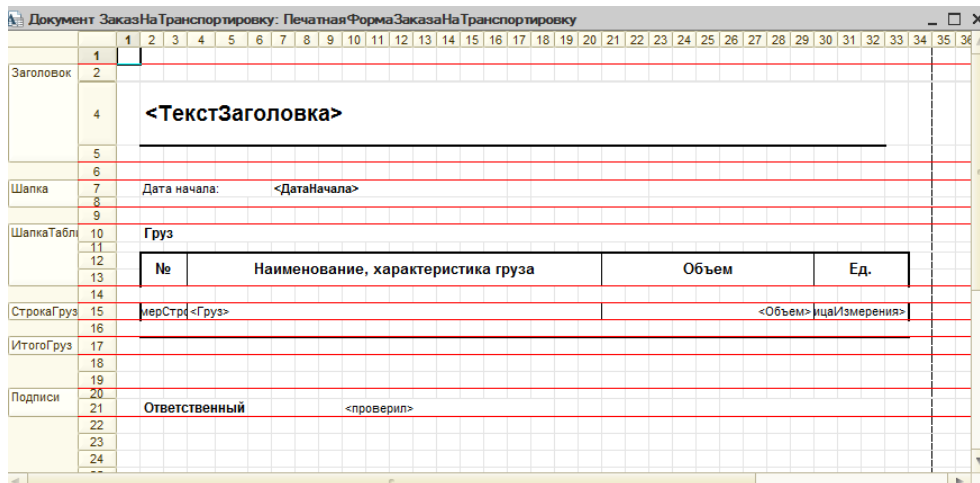


Рисунок 3.8 - Макет печати документа «Заказ на транспортировку»

В модуле имеется так же документ «Маршрутный лист», он предназначен для хранения и использования информации о маршруте транспортных средств с целью перевозки отходов на полигоны. В данном документе представлена информация в стандартизированной форме и используется для предоставления необходимым сотрудникам.

3.2 Технологическое обеспечение задачи (комплекса задач, АРМ)

В начале работы в системе 1С: Управление небольшой фирмой, в которой встроен модуль управления отходами, требуется запустить программу с помощью мышки, дважды щелкнув по ярлыку. Далее следует выбрать необходимого пользователя (себя) из раскрывающегося списка с помощью мышки. Затем требуется ввести пароль и нажать кнопку «Ок» (рисунок 3.9).

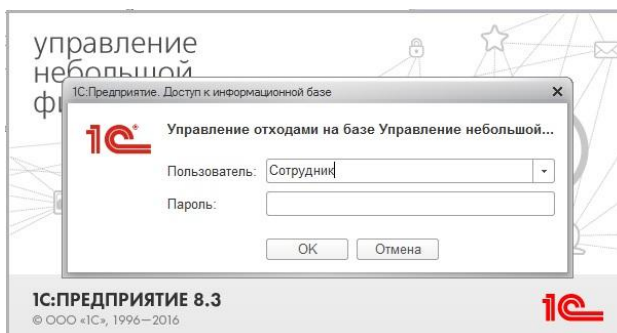


Рисунок 3.9 – Вход в систему

Запустив систему, сразу открывается начальная страница, на левой части которой есть меню (рисунок 3.10).

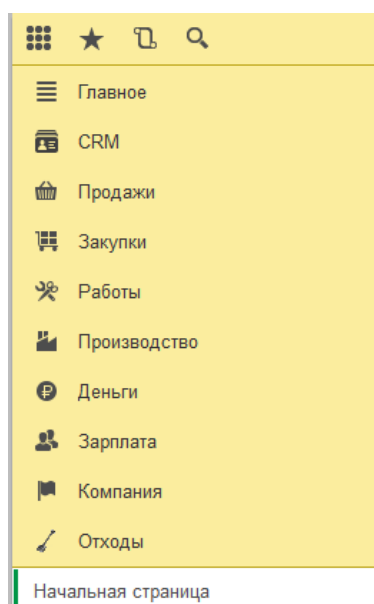


Рисунок 3.10 – Меню программы

Модуль управления отходами производства в строительстве реализован в среде разработки технологической платформы 1С: Предприятие 8.3. в виде подсистемы «Отходы». На рисунке 3.10 видно, что данная подсистема графически отразилась в меню программы. То есть другими словами в среде разработки была создан объект «Подсистемы», а в среде исполнения платформа отобразила данный объект в виде новой строки в меню, что наглядно показывает основной принцип работы выбранного технологического обеспечения[12].

3.3 Описание контрольного примера работы объектов конфигурации

Работа программы при запуске описана в предыдущем разделе. Здесь же интересует уже сам разработанный модуль управления отходами производства в строительстве. Для удобства пользователя реализовался он в виде отдельной

подсистемы «Отходы». Данное решение дает возможность, перейти к объектам данной предметной области щелкнув мышью по строке в меню.

В подсистему «Отходы» вынесены наиболее важные объекты (рисунок 3.11).

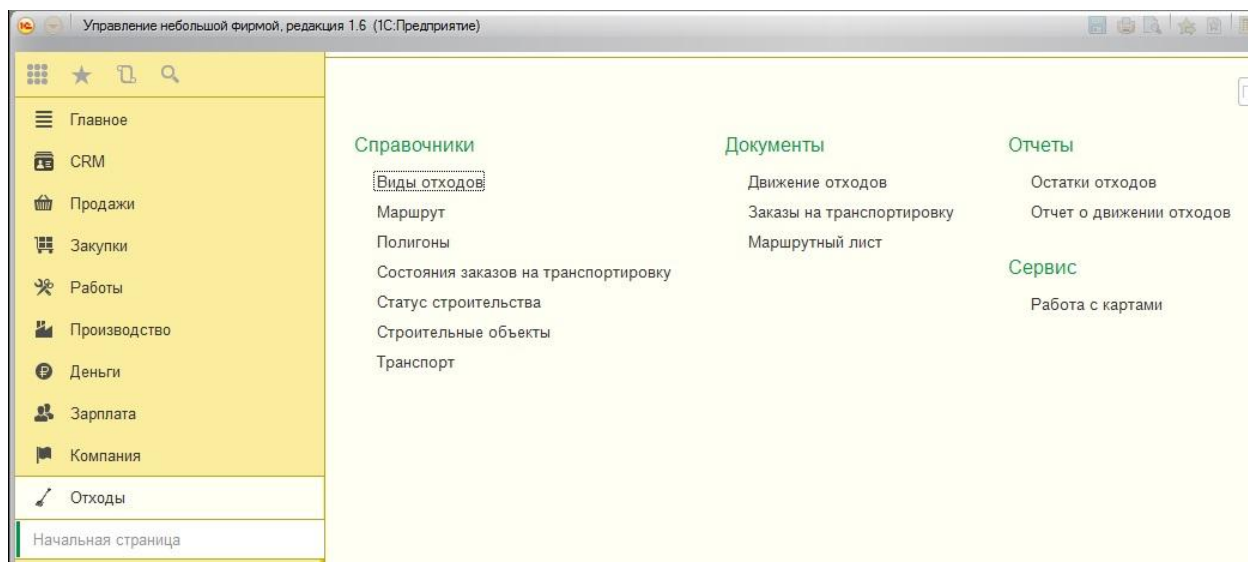


Рисунок 3.11 – Подсистема «Отходы»

Как видно на рисунке модуль состоит из следующих объектов:

1) Справочники:

- а) Виды отходов;
- б) Маршрут;
- в) Полигоны;
- г) Состояние заказов на транспортировку;
- д) Статус строительства;
- е) Строительные объекты;
- ж) Транспорт.

2) Документы:

- а) Движение отходов;
- б) Заказ на транспортировку;
- в) Маршрутный лист.

3) Отчеты:

- а) Остатки отходов;

б) Отчет о движении отходов.

4) Обработки:

а) Работа с картами.

Помимо этого используются и вспомогательные объекты, например, перечисления.

Работа в программах 1С, как правило, начинается с заполнения справочника всей известной информацией. Это говорит о том, что справочники заполняются в первую очередь, поскольку информация из них используется при создании других объектов, например, документов[1].

В качестве примера выбраны справочник «Виды отходов» и «Полигоны» (Рисунок 3.12-3.13).

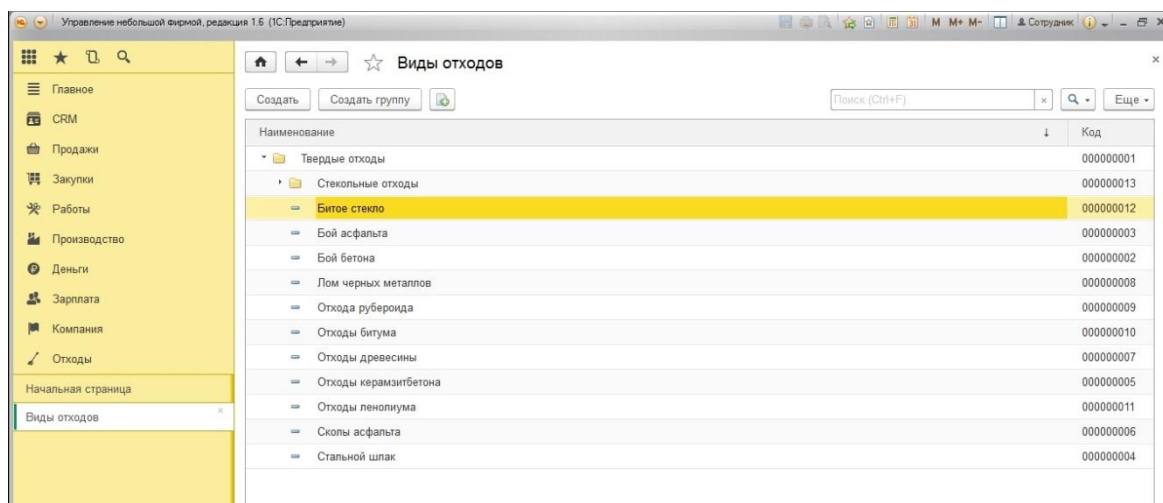


Рисунок 3.12 – Справочник «Виды отходов»

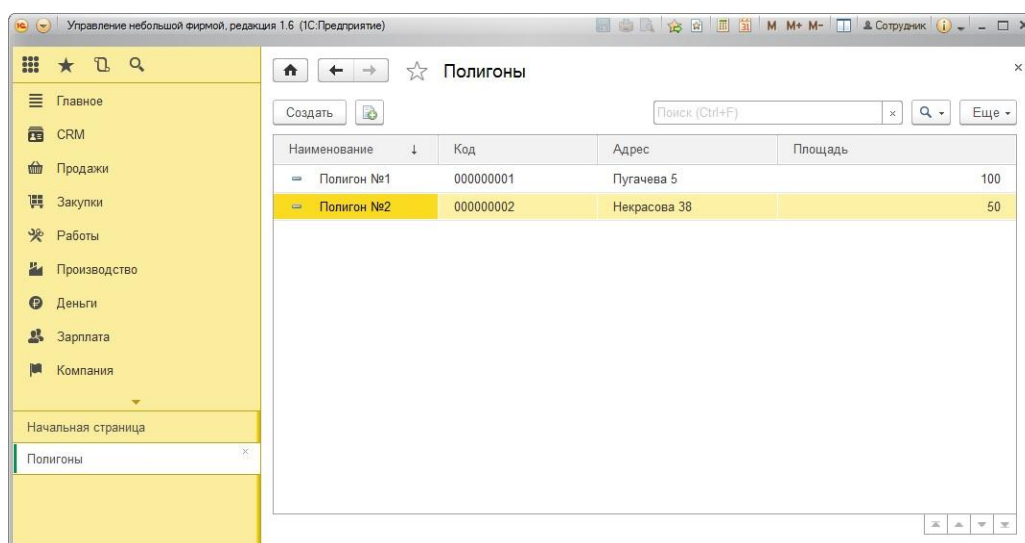


Рисунок 3.13 – Справочник «Полигоны»

Заполнение объектов конфигурации практически ничем не отличается друг от друга. Заполнение, как справочников, так и документов представляет собой внесение информации в соответствующие поля. Однако в документах еще реализована возможность их печати. На основе проведенных документов формируется информация о движении. Примеры заполнения созданных объектов приведены в приложении.

3.4. Обоснование экономической эффективности

3.4.1 Целесообразность разработки с экономической точки зрения

Оценка экономической эффективности информационной системы представляет собой процесс, который состоит из определения, понимания и измерения того, насколько выгодным экономически является внедрение информационной системы для конкретной организации. Так же можно добавить что экономическая эффективность зачастую определяется как денежный эквивалент того, насколько произошло колебание доходов или расходов организации в результате вложения в ИС[14].

При оценке экономической эффективности информационной системы можно выделить следующие этапы:

- расчёт общей стоимости владения и использования ИС;
- определение экономической эффективности проектов осуществляется на основе единых принципов;
- определение экономической эффективности, используя традиционные способы, т.е. эффективность, определяется как соотношение результатов и затрат;
- разработка сбалансированной системы показателей для определения экономического эффекта;

- оценка внедрения информационной системы как инвестиционного проекта.

Финансовые результаты внедрения информационной системы отражаются в виде показателей коммерческой эффективности проекта в целом. Основными показателями для расчёта коммерческой эффективности проекта являются следующие:

- чистый дисконтированный доход;
- внутренняя норма доходности;
- чистый доход;
- срок окупаемости;
- индексы доходности затрат и инвестиций.

В свете всего выше сказанного, можно сделать вывод, что процесс оценки экономической эффективности информационных систем не так прост, как кажется с первого взгляда. В каждом отдельном случае следует учитывать особенности анализируемой системы и организации, для которой она разрабатывалась. Используя предложенные методики и схемы, можно снизить погрешности и исключить «человеческий фактор».

3.4.2 Расчет экономической эффективности разработки

В данном разделе проводится расчет экономических показателей и анализ эффективности разработки модуля управления отходами производства в ООО «Строй Дизайн».

Используя разрабатываемый модуль в ходе выполнения выпускной квалификационной работы, предполагается получение эффекта в сфере управления транспортировкой отходов за счет, повышения эффективности планирования, контроля работы организации; а также в информационной сфере

за счет снижения трудоемкости и сокращения времени на обработку информации.

Реализация разработки в результате должна обеспечить следующие цели:

1) автоматизация работы менеджеров отдела продаж по ведению учета транспортировок отходов на 10% и снижение трудоемкости анализа деятельности на 10% за счет сокращения затрат времени на составление отчетности – т.е. планируется в общей сложности снизить трудоемкость работы на 20%.

В результате предполагается достигнуть следующего эффекта:

Число менеджеров в отделе – 3 человека.

Средняя заработная плата – 8 000 руб.

В среднем на работу по учету и анализу перевозок менеджер тратит 30% рабочего времени. Таким образом, экономия от сокращения трудоемкости составит $0,2 * 3 * 8000 \text{ руб.} * 12 \text{ мес.} * 0,3 = 17280 \text{ руб.}$

Для оценки экономического эффекта от проекта по внедрению рассматриваемой системы были произведены расчеты затрат организации на реализацию разработки. Далее поэтапно приводится расчет приведенных затрат на разработку. Все расчеты сведены в таблицы.

1) Определим количество объектных точек:

Предполагается создание и настройка следующих форм:

- Справочник «Виды отходов»;
- Справочник «Маршрут»;
- Справочник «Полигоны»;
- Справочник «Состояние заказов на транспортировку»;
- Справочник «Статус строительства»;
- Справочник «Строительные объекты»;
- Справочник «Транспорт»;
- Документ «Движение отходов»;
- Документ «Заказ на транспортировку»;

– Документ «Маршрутный лист».

Таким образом, требуется настройка 10 форм. Разработку таких форм можно считать умеренным уровнем сложности.

Также предполагается разработка как минимум 2 отчетов умеренной сложности:

- 2 отчета по работе с отходами производства.

Расчет количества объектных точек представлен в таблице 3.1.

Таблица 3.1 – Расчет количества объектных точек

Элемент проектирования	Количество	Уровень сложности
Экранные формы	10	3
Отчеты	2	4
Количество объектных точек	$10*3+2*4$	38

2) Оценка затрат проводится по методу КСМ2 на уровне предварительного прототипирования (так как проектирование системы проводится по методу типового объектного проектирования, существует прототип системы, требующий конфигурирования). Процент повторного использования оценивается в 15%. Расчет затрат времени на разработку системы представлен в таблице 3.2.

Таблица 3.2– Расчет времени разработки системы

1	Обозначение	Формула	Значение
	2	3	4
Количество объектных точек	КОТ		38
% повторного использования	%		15%
КОТ предварительного прототипирования	КОТ _{пп}	$КОТ*(1-\%)$	32,3
Среднее число строк на точку	s		20
Размер, строк	R	$КОТ_{пп}*s$	646
Число разработчиков, чел.	Ч		1
Средняя производительность одного разработчика, строк/мес.	PR		100
Время разработки, мес.	T	$R/(PR*Ч)$	6,46

Таким образом, для разработки проекта потребуется около 6,46 месяцев.

В таблице 3.3 представлен расчет затрат на разработку системы. При расчете предполагается что разработчики тратят на работу над рассматриваемым проектом 50% времени (коэффициент отчислений на зарплату – 0,5).

Таблица 3.3 – Расчет затрат на проектирование

	Обозначение	Значение
Число сотрудников, чел.	R_i	1
Заработная плата, руб./мес.	Z_i	10 000
Время разработки, мес.	T_i	6,46
Коэффициент отчислений	"	0,5
Количество категорий разработчиков.	N	1
Прочие расходы, руб.	$D_{пр}=0,6 \cdot K_{проект}$	48 450
Затраты на проектирование, руб.	$K_{проект} = \sum_{i=1}^N R_i \cdot Z_i \cdot T_i \cdot \eta + D_{пр}$	80 750

Расчет приведенных затрат на реализацию проекта сведен в таблицу 3.5. На этапе предпроектного обследования были рассмотрены и проанализированы уже использующиеся в организации информационные системы. Был сделан вывод, что приобретать новый программный продукт не имеет смысла, гораздо более целесообразно добавить модуль управления отходами.

Таблица 3.5 – Расчет приведенных затрат

№	Наименование	Расчет	Значение
1	2	3	4
1	Капитальные затраты	$1=2+5+11$	81897,91
2	стоимость разработки, руб.	$2=3+4$	80 750

Продолжение таблицы 3.5

1	2	3	4
3	стоимость ПП, руб.		–
4	стоимость проектирования, руб.		80 750
5	восстановление основного оборудования (ПК), руб.	$5=6/7*8*9*10$	1076
6	стоимость ПК, руб.		20 000
7	срок полезного использования, лет		5
8	кол-во ПК, шт.		1
9	% времени работы SE, %		50%
10	время разработки, лет	6,46мес/12мес	0,538
11	электроэнергия, руб.	$11=12*13*14*15*16$	71,91
12	время разработки, час	$6,46мес*8раб.час.*22дня$	1136,96
13	средние затраты электроэнергии на работу компьютера в течении часа, кВт		0,05
14	цена, руб./кВт час		2,53
15	кол-во ПК, шт.		1
16	% времени работы SE, %		50%
17	Эксплуатационные затраты	$17=18+19+20+21+27$	97 702
18	з/плата персоналу, руб./год	$5000руб.*12мес.$	60 000
19	вспомогательные материалы	$500руб.*12мес.$	6 000
20	плата за сопровождение, руб./год		30 000
21	электроэнергия, руб./год	$21=22*23*24*25*26$	102

Продолжение таблицы 3.5

1	2	3	4
22	время работы, час/год	8час/день 252 раб.дня	2 016
23	средние затраты электроэнергии на работу компьютера в течении часа, кВт		0,05
24	цена, руб./кВт час		2,53
25	кол-во ПК, шт.		4
26	% времени работы SE, %		10%
27	восстановления основного оборудования, руб./год	$27=28/29*30*31$	1600
28	стоимость ПК, руб.		20 000
29	срок полезного использования, лет		5
30	кол-во ПК		4
31	% времени работы SE, %		10%
32	Нормативный коэффициент приведения		25%
33	Приведенные затраты	$31=1+17*30$	106 323,41

К эксплуатационным затратам в случае разрабатываемого модуля управления отходами относятся следующие:

а) затраты на заработную плату персонала (сюда относят премии и пр. надбавки за корректную и четкую работу с системой). Выделяется по 5000руб. в месяц.

б) затраты на вспомогательные материалы, связанные с функционированием системы (бумага, картриджи и пр.)

$$P_{\text{вспом.год}} = 500\text{руб.} * 12\text{мес.} = 6\ 000\ \text{руб.}$$

в) абонентская плата сопровождения за год: 30 000 руб.

- г) затраты на электроэнергию
- д) затраты на восстановление оборудования

Как видно из расчетной таблицы приведенные затраты на реализацию разработки составляют 106 323,41 рублей.

Для вычисления срока окупаемости использовались следующие расчеты:

Эксплуатационные расходы на выполнение задачи составят:

$$P = P_{осн} + P_{ауп} + P_{общ}$$

$$P_{осн} = (10\,000 * 12 * 1,385) * 2 = 332,4 \text{ тыс. руб}$$

$$P_{ауп} = 0,25 P_{осн} = 83,1 \text{ тыс. руб}$$

$$P_{общ} = 0,2 (P_{осн} + P_{общ}) = 83,1 \text{ тыс. руб}$$

Итого эксплуатационные расходы на задачу до внедрения системы составляют 498,6 тыс. руб.

Эксплуатационные расходы на выполнение задачи составят:

$$P = P_{осн} + P_{ауп} + P_{общ}$$

$$P_{осн} = (12\,000 * 12 * 1,385) * 1 = 199,44 \text{ тыс. руб}$$

$$P_{ауп} = 0,25 P_{осн} = 49,86 \text{ тыс. руб}$$

$$P_{общ} = 0,2 (P_{осн} + P_{общ}) = 49,86 \text{ тыс. руб}$$

Итого эксплуатационные расходы на задачу после внедрения системы составляют 299,160 тыс. руб.

Условный экономический эффект:

$$\Delta Э = P_0 - P_1 = 498,6 - 299,160 = 199,44 \text{ тыс. руб}$$

Срок окупаемости разработанной системы составит:

$$T_p = K / \Delta Э = 80,75 / 199,44 = 0,4$$

Срок окупаемости системы без учета высвобожденного денежного потока составляет 5 месяцев.

ЗАКЛЮЧЕНИЕ

В деятельности любого производства (предприятия) всегда имеются отходы, поэтому перед любой организацией возникает проблема с транспортировкой отходов. Ведь отходы требуется убрать так, чтобы они не могли навредить ни людям, ни окружающей среде. Все это доказывает актуальность темы работы.

Целью данной выпускной работы являлось повышение эффективности деятельности организации за счет снижения расхода времени и денег.

Для достижения поставленной цели были решены следующие задачи:

- изучена и проанализирована деятельность организации;
- выявлены недостатки в деятельности организации;
- спроектированы структурно-функциональные диаграммы;
- разработаны решения по устранению недостатков в работе организации;
- протестировано разработанное решение.

Разработанный модуль управления отходами производства несет ряд функциональных возможностей:

- хранение, редактирование и использование информации об отходах, и их транспортировке;
- создание и регистрация соответствующей документации;
- обработка карт с целью построения маршрутов;
- формирование различной отчетности.

Разработанный модуль позволит существенно повысить эффективность работы организации, снизив ее расходы. Экономический анализ, показал, что система эффективна с экономической точки зрения и после ввода его в эксплуатацию компании обеспечит экономию денежных средств. В будущем предполагается дальнейшее развитие и совершенствование разработанного модуля и системы в целом.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Алексеев, А.А. Руководство пользователя 1С:Предприятие 8.3 [Текст] / А.А. Алексеев, А.И. Безбородов. - Москва: 1С-Паблишинг, 2015.- 276 с.
2. Алексеев, А.А. Руководство разработчика 1С:Предприятие 8.3 [Текст] / А.А. Алексеев, А.И. Безбородов. - Москва: 1С-Паблишинг, 2015.- 844 с.
3. Балдин, К.В.: Информационные системы в экономике [Текст] / К.В. Балдин. - Москва: ИНФРА-М, 2012 г. –182 с.
4. Вендров, А.М. Проектирование программного обеспечения экономических информационных систем: Учебник [Текст]/ А.М. Вендров. - Москва: Финансы и статистика, 2013 – 522с.
5. Голицына, О.Л. Программное обеспечение [Текст]/ О. Л. Голицына, И. И. Попов, Т. Л. Партыка. – Москва: Форум, 2013. – 448 с.
6. Избачков, Ю.С. Информационные системы [Текст]/ Ю.С. Избачков. - СПб.: Питер, 2012 г. – 764 с.
7. Каймин, В.А. Бизнес- процессы: Учебник [Текст] / В.А. Каймин - М.: ИНФРА-М, 2013 – 179 с.
8. Маклаков, С.В. Создание информационных систем с AllFusion Modeling Suite [Текст]/ С.В. Маклаков. – Москва: Диалог-МИФИ, 2012. - 432 с.
9. Назарова, С.В. Компьютерные технологии обработки информации [Текст] / С.В. Назарова - Москва: Финансы и статистика, 2013. – 248 с.
10. Петров, В.Н. Информационные системы [Текст]/ В.Н. Петров. – Санкт-Петербург: Питер, 2012. – 688 с.
11. Радченко, М.Г. 1С:Предприятие 8.3. Коротко о главном. Новые возможности версии 8.3 [Текст]/ М.Г. Радченко. – Москва: ООО 1С-Паблишинг, 2015. - 534 с.

12. Радченко, М.Г. «1С:Предприятие 8.3» Практическое пособие разработчика. Примеры и типовые приемы [Текст]/ М.Г. Радченко. – Москва: 1С-Паблишинг, 2015 – 764с.

13. Репин, В.В, Елиферов В.Г. Процессный подход к управлению. Моделирование бизнес-процессов: [Текст]/ В.В Репин, В.Г. Елиферов. – Москва: Манн, Иванов и Фербер, 2013 – 524 с.

14. Старик Д.Е. Расчеты экономической эффективности инвестиций [Текст]/ Д.Е. Старик. – Москва: МГТУ им. Н.Э. Баумана, 2012. – 334 с.

15. Федорова, Е.Н. Теоретические основы программирования: учебное пособие [Текст]/ Е. Н. Федорова.- Москва: МГИУ, 2012.-214 с.

16. Хрусталева, Е. Ю. Язык запросов «1С:Предприятия 8» [Текст]/ Е.Ю. Хрусталева. – Москва: ООО 1С-Паблишинг, 2013. - 369 с.

ПРИЛОЖЕНИЕ

Программный код объекта «Работа с картами»

Перем Поставщик;
Перем ИсточникСобытийКарты;
Перем МассивВременныхФайлов;

```
////////////////////////////////////  
////////////////////////////////ПРОЦЕДУРЫ ФОРМЫ////////////////////////////////  
////////////////////////////////////
```

Процедура ПриОткрытии()
 Заголовок = "Работа с картами. Поставщик - " + Поставщик;
 ИнициализироватьКарту();
КонецПроцедуры

Процедура ПриЗакрытии()
 //удаление файла карты
 Файл = Новый Файл(КаталогВременныхФайлов() + "Карта.html");
 Если Файл.Существует() Тогда
 УдалитьФайлы(Файл.ПолноеИмя);
 КонецЕсли;

 //удаление временных файлов
 Для Каждого ТекЭлемент Из МассивВременныхФайлов Цикл
 УдалитьФайлы(ТекЭлемент);
 КонецЦикла;
КонецПроцедуры

Процедура ИнициализироватьКарту()
 ПутьКФайлу = КаталогВременныхФайлов() + "Карта.html";
 Ф = новый Файл(ПутьКФайлу);
 Если Ф.Существует() Тогда
 УдалитьФайлы(ПутьКФайлу);
 КонецЕсли;

 Т = новый ТекстовыйДокумент;
 ТД = ЭтотОбъект.ПолучитьМакет("Макет" + Поставщик);
 Т.УстановитьТекст(ТД.ПолучитьТекст());
 Т.Записать(ПутьКФайлу);

 ЭлементыФормы.Эксплорер.Перейти(ПутьКФайлу);
КонецПроцедуры

Процедура ОчиститьКарту()
 ВыполнитьСкрипт("Reset()");
КонецПроцедуры

Процедура ВыполнитьСкрипт(Команда) Экспорт

```
ЭлементыФормы.Эксплорер.document.getElementById("WebClientOperation").value =  
Команда;  
ЭлементыФормы.Эксплорер.document.getElementById("WebClient").click();  
КонецПроцедуры
```

```
////////////////////////////////////  
////////////////////////////////////ГЕОКОДИРОВАНИЕ И ПОИСК АДРЕСА////////////////////////////////////  
////////////////////////////////////
```

```
Процедура АдресПриИзменении(Элемент)  
НайтиАдресНаКарте(ТекАдрес);  
КонецПроцедуры
```

```
Процедура ПоискАдресаНажатие(Элемент)  
НайтиАдресНаКарте(ТекАдрес);  
КонецПроцедуры
```

```
Процедура НайтиАдресНаКарте(Адрес = "")  
//поиск адреса  
Если Адрес = "" Тогда  
Адрес = "Москва";  
КонецЕсли;
```

```
// дальше пробуем с помощью геокодинга вывести данные поиска в таблицу  
ТаблицаАдресов.Очистить();
```

```
ПоискАдреса(Адрес);  
Если Поставщик = "Яндекс" Тогда  
ПроизвестиГеокодинг_Яндекс();  
ИначеЕсли Поставщик = "Гугл" Тогда  
ПроизвестиГеокодинг_Гугл();  
ИначеЕсли Поставщик = "2ГИС" Тогда  
ПроизвестиГеокодинг_2ГИС();  
КонецЕсли;  
КонецПроцедуры
```

```
Процедура ПоискАдреса(Адрес)  
ВыполнитьСкрипт("FindAdres('"' + Адрес + "'");");  
КонецПроцедуры
```

```
//Процедура выводит в таблицу данные геокодинга Яндекса  
//
```

```
//Параметры:
```

```
// ТекАдрес - Строка
```

```
Процедура ПроизвестиГеокодинг_Яндекс()
```

```
Яндекс = Новый HTTPСоединение("geocode-maps.yandex.ru", , , , 20, Новый  
ЗащищенноеСоединениеOpenSSL);
```

```
ВременныйФайл = КаталогВременныхФайлов() + "Yandex_geocode_" +  
СокрЛП(Новый УникальныйИдентификатор);  
Попытка
```

```

        Яндекс.Получить("/1.x/?geocode=" + ТекАдрес + "&results=10",
ВременныйФайл);
        Исключение
        Сообщить("Ошибка при попытке геокодировать по яндексу адрес: " +
ТекАдрес);
        Сообщить(ОписаниеОшибки());
        Возврат;
        КонецПопытки;

        ЧтениеXML = Новый ЧтениеXML;
        ЧтениеXML.ОткрытьФайл(ВременныйФайл);

        ПостроительDOM = Новый ПостроительDOM;
        ДокументДОМ = ПостроительDOM.Прочитать(ЧтениеXML);

        СписокТекст =
ДокументДОМ.ПолучитьЭлементыПоИмени("text");
        СписокPos =
ДокументДОМ.ПолучитьЭлементыПоИмени("pos");

        Если (СписокТекст.Количество() = 0) ИЛИ (СписокPos.Количество() = 0) Тогда
            Возврат;
        КонецЕсли;

        Для ь = 0 по СписокТекст.Количество()-1 Цикл
            Координаты = СписокPos[ь].ТекстовоеСодержимое;
            Разделитель = Найти(Координаты, " ");
            Широта = Число(Сред(Координаты, Разделитель + 1));
            Долгота = Число(Лев(Координаты, Разделитель - 1));

            Если Широта = 0 ИЛИ Долгота = 0 Тогда
                Продолжить;
            КонецЕсли;

            стрАдрес = ТаблицаАдресов.Добавить();
            Если СписокТекст.Количество() > ь Тогда
                стрАдрес.Адрес = СписокТекст[ь].ТекстовоеСодержимое;
                стрАдрес.Широта = Широта;
                стрАдрес.Долгота = Долгота;
            КонецЕсли;
        КонецЦикла;
        МассивВременныхФайлов.Добавить(ВременныйФайл);
    КонецПроцедуры

//Процедура выводит в таблицу данные геокдинга Гугл
//
//Параметры:
// ТекАдрес - Строка
Процедура ПроизвестиГеокдинг_Гугл()
    Гугл = Новый НТТРСоединение("maps.googleapis.com");

```



```

    ВременныйФайл = КаталогВременныхФайлов() + "Google_geocode_" +
    СокрЛП(Новый УникальныйИдентификатор);
    Попытка
        Гугл.Получить("/maps/api/geocode/xml?address=" + ТекАдрес +
"&language=ru&sensor=false", ВременныйФайл);
    Исключение
        Сообщить("Ошибка при попытке геокодировать по Google адрес: " +
ТекАдрес);
        Сообщить(ОписаниеОшибки());
    Возврат;
    КонецПопытки;

    ЧтениеXML = Новый ЧтениеXML;
    ЧтениеXML.ОткрытьФайл(ВременныйФайл);

    ПостроительDOM = Новый ПостроительDOM;
    ДокументДОМ = ПостроительDOM.Прочитать(ЧтениеXML);
    ТаблицаРезультатов = ДокументДОМ.ПолучитьЭлементыПоИмени("result");

    Если ДокументДОМ.ПолучитьЭлементыПоИмени("status")[0].ТекстовоеСодержимое
    < "ОК" ИЛИ ТаблицаРезультатов.Количество() = 0 Тогда
        Возврат;
    КонецЕсли;

    Для ТекРезультат = 0 по ТаблицаРезультатов.Количество() - 1 Цикл

        СписокТекст =
ТаблицаРезультатов[ТекРезультат].ПолучитьЭлементыПоИмени("formatted_address");
        ЭлементыШиротаДолгота =
ТаблицаРезультатов[ТекРезультат].ПолучитьЭлементыПоИмени("location");
        Широта =
ЭлементыШиротаДолгота[0].ПолучитьЭлементыПоИмени("lat")[0].ТекстовоеСодержимое;
        Долгота =
ЭлементыШиротаДолгота[0].ПолучитьЭлементыПоИмени("lng")[0].ТекстовоеСодержимое;
        Если Широта = 0 ИЛИ Долгота = 0 Тогда
            Продолжить;
        КонецЕсли;

        стрАдрес = ТаблицаАдресов.Добавить();
        стрАдрес.Широта = Широта;
        стрАдрес.Долгота = Долгота;
        стрАдрес.Адрес = СписокТекст[0].ТекстовоеСодержимое;
    КонецЦикла;
    МассивВременныхФайлов.Добавить(ВременныйФайл);
КонецПроцедуры

//Процедура выводит в таблицу данные геодинга 2ГИС
//
//Параметры:
// ТекАдрес - Строка
Процедура ПроизвестиГеокодинг_2ГИС()

```

```
//для НТТР геокодинга нужен ключ API. У меня его нет, поэтому я и не писал
процедуры, т.к. нет возможности проверить правильность ее выполнения
стрАдрес = ТаблицаАдресов.Добавить();
стрАдрес.Адрес = "Для НТТР геокодинга нужен ключ API";
КонецПроцедуры
```

```
Процедура ТаблицаАдресовВыбор(Элемент, ВыбраннаяСтрока, Колонка,
СтандартнаяОбработка)
```

```
Если ЗначениеЗаполнено(ВыбраннаяСтрока.Широта) И
ЗначениеЗаполнено(ВыбраннаяСтрока.Долгота) Тогда
    СтандартнаяОбработка = Ложь;
КонецЕсли;
```

```
Широта = Формат(ВыбраннаяСтрока.Широта, "ЧЦ=10; ЧДЦ=7; ЧРД=.; ЧРГ=");
Долгота = Формат(ВыбраннаяСтрока.Долгота, "ЧЦ=10; ЧДЦ=7; ЧРД=.; ЧРГ=");
```

```
ОбратнПоискАдреса(Широта, Долгота, ВыбраннаяСтрока.Адрес);
КонецПроцедуры
```

```
Процедура ОбратнПоискАдреса(Широта, Долгота, Адрес)
```

```
ВыполнитьСкрипт("ReverseSearchAdres(" + Широта + ", " + Долгота + ", "" + Адрес +
""");");
КонецПроцедуры
```

```
////////////////////////////////////
////////////////////////////////////МАРШРУТИЗАЦИЯ, КЛАСТЕРА И ПОЛИГОН////////////////////////////////////
////////////////////////////////////
```

```
Процедура КоманднаяПанель1ПостроитьМаршрут(Кнопка)
```

```
Если ТаблицаТочек.Количество() <= 1 Тогда
    Предупреждение("Недостаточно точек для построение маршрута!");
    Возврат;
КонецЕсли;
```

```
ИсточникСобытийКарты = "МаршрутныйЛист";
```

```
Если Поставщик = "Яндекс" Тогда
```

```
    ПостроитьМаршрут_Яндекс();
```

```
ИначеЕсли Поставщик = "Гугл" Тогда
```

```
    ПостроитьМаршрут_Гугл();
```

```
ИначеЕсли Поставщик = "2ГИС" Тогда
```

```
    ПостроитьМаршрут_2ГИС();
```

```
КонецЕсли;
```

```
КонецПроцедуры
```

```
//Процедура выстраивает маршрут для Яндекса
```

```
//
```

```
//Параметры:
```

```
//
```

```
Процедура ПостроитьМаршрут_Яндекс()
```

```
    ПараметрыМаршрута = ПолучитьПараметрыМаршрутаЯндекс();
```

```
    ОчиститьКарту();
```

```

        ВыполнитьСкрипт("calcRoute(" + ПараметрыМаршрута + ")");
КонецПроцедуры

//Функция получает массив точек, для передачи параметров в Яндекс
//
//Параметры:
//
//Возвращаемое значение:
// Строка
Функция ПолучитьПараметрыМаршрутаЯндекс()
    Результат = "";

    Результат = Результат + "[" + СтрЗаменить(Строка(ТаблицаТочек[0].Широта), ",", ".")
+ ", " + СтрЗаменить(Строка(ТаблицаТочек[0].Долгота), ",", ".") + "],";

    Для Ин = 1 По ТаблицаТочек.Количество() - 2 Цикл
        Результат = Результат + "[" + СтрЗаменить(Строка(ТаблицаТочек[ин].Широта),
",", ".") + ", " + СтрЗаменить(Строка(ТаблицаТочек[ин].Долгота), ",", ".") + "],";
    КонецЦикла;

    Результат = Результат + "[" +
СтрЗаменить(Строка(ТаблицаТочек[ТаблицаТочек.Количество() - 1].Широта), ",", ".") + ", " +
СтрЗаменить(Строка(ТаблицаТочек[ТаблицаТочек.Количество() - 1].Долгота), ",", ".") + "]]";

    Возврат Результат;
КонецФункции

//Процедура выстраивает маршрут для Гугл
//
//Параметры:
//
Процедура ПостроитьМаршрут_Гугл()
    ПараметрыМаршрута = ПолучитьПараметрыМаршрутаГугл();

    ОчиститьКарту();
    ВыполнитьСкрипт("calcRoute(" + ПараметрыМаршрута + ")");
КонецПроцедуры

//Функция получает массив точек, для передачи параметров в Гугл
//
//Параметры:
//
//Возвращаемое значение:
// Строка
Функция ПолучитьПараметрыМаршрутаГугл()
    Результат = "";
    ВнутрМассив = "";

    Результат = Результат + "[" + СтрЗаменить(Строка(ТаблицаТочек[0].Широта), ",", ".")
+ ", " + СтрЗаменить(Строка(ТаблицаТочек[0].Долгота), ",", ".") + "],";

    Если ТаблицаТочек.Количество() = 2 Тогда

```

```

        Результат = Результат + "[],";
    Иначе
        Для Ин = 1 По ТаблицаТочек.Количество() - 2 Цикл
            ВнутрМассив = ВнутрМассив + "[" +
СтрЗаменить(Строка(ТаблицаТочек[ин].Широта), ",", ".") + "," +
СтрЗаменить(Строка(ТаблицаТочек[ин].Долгота), ",", ".") + "],";
            КонечЦикла;
        Результат = Результат + "[" + Сред(ВнутрМассив, 1, СтрДлина(ВнутрМассив) -
1) + "],";
    КонечЕсли;

    Результат = Результат + "[" +
СтрЗаменить(Строка(ТаблицаТочек[ТаблицаТочек.Количество() - 1].Широта), ",", ".") + "," +
СтрЗаменить(Строка(ТаблицаТочек[ТаблицаТочек.Количество() - 1].Долгота), ",", ".") + "]]";

    Возврат Результат;
КонечФункции

//Процедура выстраивает маршрут для 2ГИС
//
//Параметры:
//
Процедура ПостроитьМаршрут_2ГИС()
    Предупреждение("Для карт 2ГИС пока не доступна маршрутизация!");
КонечПроцедуры

//обработка щелчка, для считывания координат меток
//
Процедура ЭксплорерonClick(Элемент, pEvtObj)
    ПолучитьКоординаты();
КонечПроцедуры

//Процедура получает координаты для Яндексa
Процедура ПолучитьКоординаты()
    //Если Поставщик = "2ГИС" Тогда
    //    Если ЭлементыФормы.Эксплорер.document.parentWindow.click Тогда
    //        Попытка
    //            Элемент = ЭлементыФормы.Эксплорер;
    //            ЧислоТип = Новый ОписаниеТипов("Число", Новый
КвалификаторыЧисла(15, 12));
    //
    //            КоординатаХ =
ЭлементыФормы.Эксплорер.document.getElementById("CoordX").value;
    //            КоординатаХ = ЧислоТип.ПривестиЗначение(КоординатаХ);
    //
    //            КоординатаУ =
ЭлементыФормы.Эксплорер.document.getElementById("CoordY").value;
    //            КоординатаУ = ЧислоТип.ПривестиЗначение(КоординатаУ);
    //
    //            Если КоординатаХ > 0 И КоординатаУ > 0 Тогда
    //                НоваяСтрока = ТаблицаТочек.Добавить();

```

```

//          НоваяСтрока.Точка = "Точка" +
ТаблицаТочек.Количество();
//          НоваяСтрока.Широта = КоординатаХ;
//          НоваяСтрока.Долгота = КоординатаУ;
//          КонецЕсли;
//          Исключение
//          КонецПопытки;
//          КонецЕсли;
//Иначе
    Попытка
        Элемент = ЭлементыФормы.Эксплорер;
        ЧислоТип = Новый ОписаниеТипов("Число", Новый
КвалификаторыЧисла(15, 12));

        КоординатаХ =
ЭлементыФормы.Эксплорер.document.getElementById("CoordX").value;
        КоординатаХ = ЧислоТип.ПривестиЗначение(КоординатаХ);

        КоординатаУ =
ЭлементыФормы.Эксплорер.document.getElementById("CoordY").value;
        КоординатаУ = ЧислоТип.ПривестиЗначение(КоординатаУ);

        Если КоординатаХ > 0 И КоординатаУ > 0 Тогда
            НоваяСтрока = ТаблицаТочек.Добавить();
            НоваяСтрока.Точка = "Точка" + ТаблицаТочек.Количество();
            НоваяСтрока.Широта = КоординатаХ;
            НоваяСтрока.Долгота = КоординатаУ;

ЭлементыФормы.Эксплорер.document.getElementById("CoordX").value = "0";

ЭлементыФормы.Эксплорер.document.getElementById("CoordY").value = "0";
        КонецЕсли;
    Исключение
        КонецПопытки;
//КонецЕсли;

КонецПроцедуры

Процедура КоманднаяПанель1ОчиститьВсе(Кнопка)
    ТаблицаТочек.Очистить();
    ОчиститьКарту();
КонецПроцедуры

Процедура ОсновныеДействияПерегрузитьКарту(Кнопка)
    ИнициализироватьКарту();
КонецПроцедуры

Процедура ОсновныеДействияСправка(Кнопка)
    ФормаСправки = ЭтотОбъект.ПолучитьФорму("СправочнаяИнформация",
ЭтаФорма);
    ФормаСправки.Поставщик = Поставщик;

```

```

        ФормаСправки.Открыть();
    КонецПроцедуры

Процедура ОсновныеДействияРазработчик(Кнопка)
    ЗапуститьПриложение("http://smaylukk.com.ua/?lang=Ru");
КонецПроцедуры

Процедура Эксплорерondatasetcomplete(Элемент, pEvtObj)
    Если ИсточникСобытийКарты = "МаршрутныйЛист" Тогда
        ИнформацияОМаршруте =
ЭлементыФормы.Эксплорер.document.getElementById("RouteInfo").value;
        //еще не конец - надо теперь обработать строку - удалить символы HTML
        Пока Истина Цикл //удаление символов вида &#160;
            ПозицияНач = Найти(ИнформацияОМаршруте, "&");
            Если ПозицияНач = 0 Тогда
                Прервать;
            КонецЕсли;

            ПозицияКон = Найти(ИнформацияОМаршруте, ";");
            Если ПозицияКон = 0 Тогда
                Прервать;
            КонецЕсли;
            ИнформацияОМаршруте = Сред(ИнформацияОМаршруте, 1,
ПозицияНач - 1) + Сред(ИнформацияОМаршруте, ПозицияКон + 1);
        КонецЦикла;

        ДополнительнаяИнформация = ИнформацияОМаршруте;
    КонецЕсли;
КонецПроцедуры

Процедура КоманднаяПанель2СоздатьКластера(Кнопка)
    Если ТаблицаТочек.Количество() <= 1 Тогда
        Предупреждение("Недостаточно точек для построение кластера!");
        Возврат;
    КонецЕсли;

    Если Поставщик = "2ГИС" Тогда
        Предупреждение("Для " + Поставщик + " не работает");
    Иначе
        ОчиститьКарту();
        ПостроитьКластера();
    КонецЕсли;
КонецПроцедуры

Процедура ПостроитьКластера()
    Кол = ТаблицаТочек.Количество();
    Индекс = 1;
    Для Каждого ТекСтрока Из ТаблицаТочек Цикл
        Широта = формат(ТекСтрока.Широта, "ЧРД=");
        Долгота = формат(ТекСтрока.Долгота, "ЧРД=");
        СодержимоеТочки = "Содержимое точки"; //опять же можно вставить свое
название

```

```

        ВыполнитьСкрипт("addToPointArray(" + Широта + "," + Долгота + "," +
ТекСтрока.Точка + ",", "" + СодержимоеТочки + "");");

        Состояние("Обработан " + Индекс + " из " + кол);
        Индекс = Индекс + 1;
    КонечЦикла;

    ВыполнитьСкрипт("drawCluster()");
КонечПроцедуры

Процедура КоманднаяПанель2ПостроитьПолигон(Кнопка)
    Если ТаблицаТочек.Количество() <= 1 Тогда
        Предупреждение("Недостаточно точек для построение полигона!");
        Возврат;
    КонечЕсли;

    ОчиститьКарту();

    //Если Поставщик = "2ГИС" Тогда
    //    Сообщить("Для " + Поставщик + " не работает",
СтатусСообщения.Информация);
    //Иначе
        ПостроитьПолигон();
    //КонечЕсли;
КонечПроцедуры

//Процедура выстраивает маршрут для Яндексa
//
//Параметры:
//
Процедура ПостроитьПолигон()
    МассивТочек = "[";
    Для Каждого ТекТочка Из ТаблицаТочек Цикл
        Широта = формат(ТекТочка.Широта, "ЧРД=");
        Долгота = формат(ТекТочка.Долгота, "ЧРД=");
        МассивТочек = МассивТочек + "[" + Широта + "," + Долгота + "],";
    КонечЦикла;
    МассивТочек = Сред(МассивТочек, 1, СтрДлина(МассивТочек) - 1) + "]";

    //генерация цвета фона полигона случайным образом
    Строка16 = "0123456789ABCDEF";
    ГСЧ = Новый ГенераторСлучайныхЧисел;
    Цвет16 = "#";
    Для н = 1 По 6 Цикл
        м = ГСЧ.СлучайноеЧисло(1, 16);
        Цвет16 = Цвет16 + Сред(Строка16, м, 1);
    КонечЦикла;

    Название = "Полигон";//вставить свое :)
    ВыполнитьСкрипт("createPolygon(" + МассивТочек + "," + Название + ",", "" + Цвет16
+ "");");
КонечПроцедуры

```

```
Процедура КоманднаяПанель2ОчиститьКарту(Кнопка)
    ОчиститьКарту();
КонецПроцедуры
```

```
Процедура КоманднаяПанель2ОчиститьТаблицу(Кнопка)
    ТаблицаТочек.Очистить();
КонецПроцедуры
```

```
////////////////////////////////////
```

```
Поставщик = СтруктураПоставщиковКарт.Получить(ЭтотОбъект.ТипКарты);
ИсточникСобытийКарты = "";
МассивВременныхФайлов = Новый Массив;
```

```
Процедура ЯндексНажатие(Элемент)
    ЗапуститьПриложение("http://legal.yandex.ru/maps_api/");
КонецПроцедуры
```

```
Процедура ГуглНажатие(Элемент)
    ЗапуститьПриложение("https://developers.google.com/maps/terms?hl=ru-RU");
КонецПроцедуры
```

```
Процедура ДубльГИСНажатие(Элемент)
    ЗапуститьПриложение("http://help.2gis.ru/licensing-agreement/");
КонецПроцедуры
```

```
Процедура КартинкаЯндексНажатие(Элемент)
    ТипКарты = 0;
    ЭтотОбъект.ПолучитьФорму("Форма", ЭтаФорма, "Яндекс").Открыть();
КонецПроцедуры
```

```
Процедура КартинкаГуглНажатие(Элемент)
    ТипКарты = 1;
    ЭтотОбъект.ПолучитьФорму("Форма", ЭтаФорма, "Гугл").Открыть();
КонецПроцедуры
```

```
Процедура Картинка2ГИСНажатие(Элемент)
    ТипКарты = 2;
    ЭтотОбъект.ПолучитьФорму("Форма", ЭтаФорма, "2ГИС").Открыть();
КонецПроцедуры
```

```
Процедура ПриОткрытии()
    ЭлементыФормы.Справка.УстановитьТекст(ЭтотОбъект.ПолучитьМакет("Справка").
ПолучитьОбласть("Справка" + Поставщик).Область().Текст);
КонецПроцедуры
```

```
&НаКлиенте
```


Процедура ЛицензионноеСоглашениеЯндексНажатие(Элемент)
ЗапуститьПриложение("http://legal.yandex.ru/maps_api/");
КонецПроцедуры

&НаКлиенте
Процедура ЛицензионноеСоглашениеГуглНажатие(Элемент)
ЗапуститьПриложение("https://developers.google.com/maps/terms?hl=ru-RU");
КонецПроцедуры

&НаКлиенте
Процедура ЯндексНажатие(Элемент)
Объект.ТипКарты = 0;

ПараметрыФормы = Новый Структура;
ПараметрыФормы.Вставить("ТипКарты", Объект.ТипКарты);
ПараметрыФормы.Вставить("ЗакрыватьПриЗакрытииВладельца", Истина);
ОткрытьФорму("Обработка.РаботаСКартами.Форма.ФормаУпр", ПараметрыФормы,
ЭтаФорма, "Яндекс");
КонецПроцедуры

&НаКлиенте
Процедура ГуглНажатие(Элемент)
Объект.ТипКарты = 1;

ПараметрыФормы = Новый Структура;
ПараметрыФормы.Вставить("ТипКарты", Объект.ТипКарты);
ПараметрыФормы.Вставить("ЗакрыватьПриЗакрытииВладельца", Истина);
ОткрытьФорму("Обработка.РаботаСКартами.Форма.ФормаУпр", ПараметрыФормы,
ЭтаФорма, "Гугл");
КонецПроцедуры

&НаКлиенте
Процедура ЛицензионноеСоглашение2ГИСНажатие(Элемент)
ЗапуститьПриложение("http://help.2gis.ru/licensing-agreement/");
КонецПроцедуры

&НаКлиенте
Процедура ДубльГИСНажатие(Элемент)
Объект.ТипКарты = 2;

ПараметрыФормы = Новый Структура;
ПараметрыФормы.Вставить("ТипКарты", Объект.ТипКарты);
ПараметрыФормы.Вставить("ЗакрыватьПриЗакрытииВладельца", Истина);
ОткрытьФорму("Обработка.РаботаСКартами.Форма.ФормаУпр", ПараметрыФормы,
ЭтаФорма, "ДубльГис");
КонецПроцедуры

////////////////////////////////////
////////////////////////////////ПРОЦЕДУРЫ ФОРМЫ////////////////////////////////
////////////////////////////////////

```

&НаКлиенте
Процедура НайтиАдрес(Команда)
    НайтиАдресНаКарте(ТекАдрес);
КонецПроцедуры

&НаКлиенте
Процедура ИнициализацияКарты(Команда)
    ИнициализироватьКарту();
КонецПроцедуры

&НаКлиенте
Процедура СправочнаяИнформация(Команда)
    Элементы.ГруппаСправка.Видимость = Не Элементы.ГруппаСправка.Видимость;
КонецПроцедуры

&НаКлиенте
Процедура Разработчик(Команда)
    ЗапуститьПриложение("http://smaylukk.com.ua/?lang=Ru");
КонецПроцедуры

&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
    Объект.ТипКарты = Параметры.ТипКарты;
    ТекОбъект = РеквизитФормыВЗначение("Объект");
    Поставщик = ТекОбъект.СтруктураПоставщиковКарт.Получить(Объект.ТипКарты);

    Макет = ТекОбъект.ПолучитьМакет("Справка");
    ТекстСправки = Макет.ПолучитьОбласть("Справка" + Поставщик).Область().Текст;
    Справка = ТекстСправки;
КонецПроцедуры

&НаКлиенте
Процедура ПриОткрытии(Отказ)
    Элементы.ГруппаСправка.Видимость = Ложь;
    Заголовок = "Работа с картами. Поставщик - " + Поставщик;
    ИнициализироватьКарту();
КонецПроцедуры

&НаКлиенте
Процедура ПриЗакрытии()
    //удаление временных файлов
    Для Каждого ТекЭлемент Из МассивВременныхФайлов Цикл
        УдалитьФайлы(ТекЭлемент.Значение);
    КонецЦикла;
КонецПроцедуры

&НаКлиенте
//Процедура составляет имя процедуры поставщика
// и запускает ее на исполнение
//Параметры:
// НачалоИмени - Строка
Процедура ВыполнитьПроцедуруПоставщика(ИмяПроцедуры)

```

```
        Выполнить ИмяПроцедуры;  
КонецПроцедуры
```

```
&НаКлиенте
```

```
//процедура инициализирует карту поставщика из макета  
Процедура ИнициализироватьКарту()  
    Текст = ПолучитьТекстМакета("Макет" + Поставщик);
```

```
        Эксплорер = Текст;  
КонецПроцедуры
```

```
&НаСервере
```

```
Функция ПолучитьТекстМакета(ИмяМакета)  
    Макет = РеквизитФормыВЗначение("Объект").ПолучитьМакет(ИмяМакета);  
    Результат = Макет.ПолучитьТекст();  
    Возврат Результат;  
КонецФункции
```

```
&НаКлиенте
```

```
Процедура ОчисткаКарты()  
    Элементы.Эксплорер.document.getElementById("WebClientOperation").value =  
"Reset()";  
    Элементы.Эксплорер.document.getElementById("WebClient").click();  
КонецПроцедуры
```

```
////////////////////////////////////  
////////////////////////////////////ГЕОКОДИРОВАНИЕ И ПОИСК АДРЕСА////////////////////////////////////  
////////////////////////////////////
```

```
&НаКлиенте
```

```
Процедура НайтиАдресНаКарте(Адрес = "")  
    //поиск адреса  
    Если Адрес = "" Тогда  
        Адрес = "Москва";  
    КонецЕсли;  
  
    // дальше пробуем с помощью геокодинга вывести данные поиска в таблицу  
    ТаблицаАдресов.Очистить();
```

```
    ПоискАдреса(Адрес);  
    Если Поставщик = "Яндекс" Тогда  
        ПроизвестиГеокодинг_Яндекс();  
    ИначеЕсли Поставщик = "Гугл" Тогда  
        ПроизвестиГеокодинг_Гугл();  
    ИначеЕсли Поставщик = "2ГИС" Тогда  
        ПроизвестиГеокодинг_2ГИС();  
    КонецЕсли;  
КонецПроцедуры
```

```
&НаКлиенте
```

```
Процедура ПоискАдреса(Адрес)
```

```

        Элементы.Эксплоер.document.getElementById("WebClientOperation").value =
"FindAdres(" + Адрес + "");";
        Элементы.Эксплоер.document.getElementById("WebClient").click();
КонецПроцедуры

&НаСервере
//Процедура выводит в таблицу данные геокодинга Яндекса
//
//Параметры:
// ТекАдрес - Строка
Процедура ПроизвестиГеокодинг_Яндекс()
        Яндекс = Новый HTTPСоединение("geocode-maps.yandex.ru", , , , 20, Новый
ЗащищенноеСоединениеOpenSSL);

        ВременныйФайл = КаталогВременныхФайлов() + "Yandex_geocode_" +
СокрЛП(Новый УникальныйИдентификатор);
        Попытка
                Яндекс.Получить("/1.x/?geocode=" + ТекАдрес + "&results=10",
ВременныйФайл);
        Исключение
                Сообщить("Ошибка при попытке геокодировать по яндексу адрес: " +
ТекАдрес);
                Сообщить(ОписаниеОшибки());
        Возврат;
КонецПопытки;

        ЧтениеXML = Новый ЧтениеXML;
        ЧтениеXML.ОткрытьФайл(ВременныйФайл);

        ПостроительDOM                = Новый ПостроительDOM;
        ДокументДОМ                    = ПостроительDOM.Прочитать(ЧтениеXML);

        СписокТекст                    =
ДокументДОМ.ПолучитьЭлементыПоИмени("text");
        СписокPos                      =
ДокументДОМ.ПолучитьЭлементыПоИмени("pos");

        Если (СписокТекст.Количество() = 0) ИЛИ (СписокPos.Количество() = 0) Тогда
                Возврат;
        КонецЕсли;

        Для Ь = 0 по СписокТекст.Количество()-1 Цикл
                Координаты = СписокPos[Ь].ТекстовоеСодержимое;
                Разделитель = Найти(Координаты, " ");
                Широта = Число(Сред(Координаты, Разделитель + 1));
                Долгота = Число(Лев(Координаты, Разделитель - 1));

                Если Широта = 0 ИЛИ Долгота = 0 Тогда
                        Продолжить;
                КонецЕсли;

                стрАдрес = ТаблицаАдресов.Добавить();

```

```

        Если СписокТекст.Количество() > ь Тогда
            стрАдрес.Адрес = СписокТекст[Ъ].ТекстовоеСодержимое;
            стрАдрес.Широта = Широта;
            стрАдрес.Долгота = Долгота;
        КонецЕсли;
    КонецЦикла;
    МассивВременныхФайлов.Добавить(ВременныйФайл);
КонецПроцедуры

&НаСервере
//Процедура выводит в таблицу данные геокдинга Гугл
//
//Параметры:
// ТекАдрес - Строка
Процедура ПроизвестиГеокдинг_Гугл()
    Гугл = Новый НТТРСоединение("maps.googleapis.com");

    ВременныйФайл = КаталогВременныхФайлов() + "Google_geocode_" +
СокрЛП(Новый УникальныйИдентификатор);
    Попытка
        Гугл.Получить("/maps/api/geocode/xml?address=" + ТекАдрес +
"&language=ru&sensor=false", ВременныйФайл);
    Исключение
        Сообщить("Ошибка при попытке геокодировать по Google адрес: " +
ТекАдрес);
        Сообщить(ОписаниеОшибки());
    Возврат;
КонецПопытки;

    ЧтениеXML = Новый ЧтениеXML;
    ЧтениеXML.ОткрытьФайл(ВременныйФайл);

    ПостроительDOM = Новый ПостроительDOM;
    ДокументДОМ = ПостроительDOM.Прочитать(ЧтениеXML);
    ТаблицаРезультатов = ДокументДОМ.ПолучитьЭлементыПоИмени("result");

    Если ДокументДОМ.ПолучитьЭлементыПоИмени("status")[0].ТекстовоеСодержимое
<> "ОК" ИЛИ ТаблицаРезультатов.Количество() = 0 Тогда
        Возврат;
    КонецЕсли;

    Для ТекРезультат = 0 по ТаблицаРезультатов.Количество() -1 Цикл

        СписокТекст =
ТаблицаРезультатов[ТекРезультат].ПолучитьЭлементыПоИмени("formatted_address");
        ЭлементыШиротаДолгота =
ТаблицаРезультатов[ТекРезультат].ПолучитьЭлементыПоИмени("location");
        Широта =
ЭлементыШиротаДолгота[0].ПолучитьЭлементыПоИмени("lat")[0].ТекстовоеСодержимое;
        Долгота =
ЭлементыШиротаДолгота[0].ПолучитьЭлементыПоИмени("lng")[0].ТекстовоеСодержимое;
        Если Широта = 0 ИЛИ Долгота = 0 Тогда

```

```

        Продолжить;
    КонецЕсли;

    стрАдрес = ТаблицаАдресов.Добавить();
    стрАдрес.Широта = Широта;
    стрАдрес.Долгота = Долгота;
    стрАдрес.Адрес = СписокТекст[0].ТекстовоеСодержимое;
    КонецЦикла;
    МассивВременныхФайлов.Добавить(ВременныйФайл);
КонецПроцедуры

&НаСервере
//Процедура выводит в таблицу данные геокодинга 2ГИС
//
//Параметры:
// ТекАдрес - Строка
Процедура ПроизвестиГеокодинг_2ГИС()
    //для HTTP геокодинга нужен ключ API. У меня его нет, поэтому я и не писал
    процедуры, т.к. нет возможности проверить правильность ее выполнения
    стрАдрес = ТаблицаАдресов.Добавить();
    стрАдрес.Адрес = "Для HTTP геокодинга нужен ключ API";
КонецПроцедуры

&НаКлиенте
Процедура ОбратнПоискАдреса(Широта, Долгота, Адрес)
    Элементы.Эксплорер.document.getElementById("WebClientOperation").value =
"ReverseSearchAdres(" + Широта + "," + Долгота + "," + Адрес + "));";
    Элементы.Эксплорер.document.getElementById("WebClient").click();
КонецПроцедуры

&НаКлиенте
Процедура ТаблицаАдресовВыбор(Элемент, ВыбраннаяСтрока, Поле,
СтандартнаяОбработка)
    Если ЗначениеЗаполнено(ТаблицаАдресов[ВыбраннаяСтрока].Широта) И
ЗначениеЗаполнено(ТаблицаАдресов[ВыбраннаяСтрока].Долгота) Тогда
        СтандартнаяОбработка = Ложь;
    КонецЕсли;

    Широта = Формат(ТаблицаАдресов[ВыбраннаяСтрока].Широта, "ЧЦ=10; ЧДЦ=7;
ЧРД=.; ЧРГ=");
    Долгота = Формат(ТаблицаАдресов[ВыбраннаяСтрока].Долгота, "ЧЦ=10; ЧДЦ=7;
ЧРД=.; ЧРГ=");

    ОбратнПоискАдреса(Широта, Долгота, ТаблицаАдресов[ВыбраннаяСтрока].Адрес);
КонецПроцедуры

&НаКлиенте
Процедура ТекАдресПриИзменении(Элемент)
    НайтиАдресНаКарте(ТекАдрес);
КонецПроцедуры

```

```
////////////////////////////////////  
////////////////////////////////////МАРШРУТИЗАЦИЯ, КЛАСТЕРА И ПОЛИГОН////////////////////////////////////  
////////////////////////////////////
```

```
&НаКлиенте  
Процедура ЭксплорерПриНажатии(Элемент, ДанныеСобытия, СтандартнаяОбработка)  
    ПолучитьКоординаты();  
КонецПроцедуры
```

```
&НаКлиенте  
Процедура ПостроитьМаршрут(Команда)  
    Если ТаблицаТочек.Количество() <= 1 Тогда  
        Предупреждение("Недостаточно точек для построение маршрута!");  
        Возврат;  
    КонецЕсли;  
  
    Если Поставщик = "Яндекс" Тогда  
        ПостроитьМаршрут_Яндекс();  
    ИначеЕсли Поставщик = "Гугл" Тогда  
        ПостроитьМаршрут_Гугл();  
    ИначеЕсли Поставщик = "2ГИС" Тогда  
        ПостроитьМаршрут_2ГИС();  
    КонецЕсли;  
КонецПроцедуры
```

```
&НаКлиенте  
//Процедура выстраивает маршрут для Яндкса  
//  
//Параметры:  
//  
Процедура ПостроитьМаршрут_Яндекс()  
    ПараметрыМаршрута = ПолучитьПараметрыМаршрутаЯндкса();  
  
    ОчисткаКарты();  
  
    Элементы.Эксплорер.document.getElementById("WebClientOperation").value =  
"calcRoute(" + ПараметрыМаршрута + ")";  
    Элементы.Эксплорер.document.getElementById("WebClient").click();  
КонецПроцедуры
```

```
&НаКлиенте  
//Функция получает массив точек, для передачи параметров в Яндекс  
//  
//Параметры:  
//  
//Возвращаемое значение:  
// Строка  
Функция ПолучитьПараметрыМаршрутаЯндкса()  
    Результат = "";  
  
    Результат = Результат + "[" + СтрЗаменить(Строка(ТаблицаТочек[0].Широта), ",", ".")  
+ "," + СтрЗаменить(Строка(ТаблицаТочек[0].Долгота), ",", ".") + "],";
```

```

    Для Ин = 1 По ТаблицаТочек.Количество() - 2 Цикл
        Результат = Результат + "[" + СтрЗаменить(Строка(ТаблицаТочек[ин].Широта),
        ",", ".") + "," + СтрЗаменить(Строка(ТаблицаТочек[ин].Долгота), ",", ".") + "],";
    КонечЦикла;

```

```

    Результат = Результат + "[" +
    СтрЗаменить(Строка(ТаблицаТочек[ТаблицаТочек.Количество() - 1].Широта), ",", ".") + "," +
    СтрЗаменить(Строка(ТаблицаТочек[ТаблицаТочек.Количество() - 1].Долгота), ",", ".") + "]]";

```

```

    Возврат Результат;
КонечФункции

```

&НаКлиенте

```
//Процедура выстраивает маршрут для Гугл
```

```
//
```

```
//Параметры:
```

```
//
```

```
Процедура ПостроитьМаршрут_Гугл()
```

```
    ПараметрыМаршрута = ПолучитьПараметрыМаршрутаГугл();
```

```
    ОчисткаКарты();
```

```
    Элементы.Эксплорер.document.getElementById("WebClientOperation").value =
"calcRoute(" + ПараметрыМаршрута + ")";
```

```
    Элементы.Эксплорер.document.getElementById("WebClient").click();
```

```
КонечПроцедуры
```

&НаКлиенте

```
//Функция получает массив точек, для передачи параметров в Гугл
```

```
//
```

```
//Параметры:
```

```
//
```

```
//Возвращаемое значение:
```

```
// Строка
```

```
Функция ПолучитьПараметрыМаршрутаГугл()
```

```
    Результат = "";
```

```
    ВнутрМассив = "";
```

```
    Результат = Результат + "[" + СтрЗаменить(Строка(ТаблицаТочек[0].Широта), ",", ".")
+ "," + СтрЗаменить(Строка(ТаблицаТочек[0].Долгота), ",", ".") + "],";
```

```
    Если ТаблицаТочек.Количество() = 2 Тогда
```

```
        Результат = Результат + "[],";
```

```
    Иначе
```

```
        Для Ин = 1 По ТаблицаТочек.Количество() - 2 Цикл
```

```
            ВнутрМассив = ВнутрМассив + "[" +
```

```
            СтрЗаменить(Строка(ТаблицаТочек[ин].Широта), ",", ".") + "," +
```

```
            СтрЗаменить(Строка(ТаблицаТочек[ин].Долгота), ",", ".") + "],";
```

```
        КонечЦикла;
```

```
        Результат = Результат + "[" + Сред(ВнутрМассив, 1, СтрДлина(ВнутрМассив) -
1) + "],";
```



```

КонецЕсли;

Результат = Результат + "[" +
СтрЗаменить(Строка(ТаблицаТочек[ТаблицаТочек.Количество() - 1].Широта), ",", ".") + "," +
СтрЗаменить(Строка(ТаблицаТочек[ТаблицаТочек.Количество() - 1].Долгота), ",", ".") + "]]";

Возврат Результат;
КонецФункции

&НаКлиенте
//Процедура выстраивает маршрут для 2ГИС
//
//Параметры:
//
Процедура ПостроитьМаршрут_2ГИС()
ПоказатьПредупреждение( "Для карт 2ГИС пока не доступна маршрутизация!");
КонецПроцедуры

&НаКлиенте
Процедура СоздатьКластер(Команда)
Если ТаблицаТочек.Количество() <= 1 Тогда
ПоказатьПредупреждение("Недостаточно точек для построение кластера!");
Возврат;
КонецЕсли;

Если Поставщик = "2ГИС" Тогда
ПоказатьПредупреждение( "Для " + Поставщик + " не работает");
Иначе
ОчисткаКарты();
ПостроитьКластера();
КонецЕсли;
КонецПроцедуры

&НаКлиенте
Процедура ПостроитьКластера()
Кол = ТаблицаТочек.Количество();
Индекс = 1;
Для Каждого ТекСтрока Из ТаблицаТочек Цикл
Широта = формат(ТекСтрока.Широта, "ЧРД=");
Долгота = формат(ТекСтрока.Долгота, "ЧРД=");
СодержимоеТочки = "Содержимое точки"; //опять же можно вставить свое
название
Элементы.Эксплорер.document.getElementById("WebClientOperation").value =
"addToPointArray(" + Широта + "," + Долгота + "," + ТекСтрока.Точка + "," + "" +
СодержимоеТочки + """);";
Элементы.Эксплорер.document.getElementById("WebClient").click();

Состояние("Обработан " + Индекс + " из " + кол);
Индекс = Индекс + 1;
КонецЦикла;

```

```
Элементы.Эксплорер.document.getElementById("WebClientOperation").value =
"drawCluster()";
Элементы.Эксплорер.document.getElementById("WebClient").click();
КонецПроцедуры
```

```
&НаКлиенте
Процедура ОчиститьВсе(Команда)
    ОчисткаКарты();
    ТаблицаТочек.Очистить();
КонецПроцедуры
```

```
&НаКлиенте
Процедура ОчиститьКарту(Команда)
    ОчисткаКарты();
КонецПроцедуры
```

```
&НаКлиенте
Процедура ОчиститьТаблицу(Команда)
    ТаблицаТочек.Очистить();
КонецПроцедуры
```

```
&НаКлиенте
//Процедура получает координаты установленной точки
Процедура ПолучитьКоординаты()
    Если Поставщик = "2ГИС" Тогда
        Если Элементы.Эксплорер.document.parentWindow.click Тогда
            Попытка
                ЧислоТип = Новый ОписаниеТипов("Число", Новый
КвалификаторыЧисла(15, 12));
```

```
                КоординатаХ =
Элементы.Эксплорер.document.getElementById("CoordX").value;
                КоординатаХ = ЧислоТип.ПривестиЗначение(КоординатаХ);
```

```
                КоординатаУ =
Элементы.Эксплорер.document.getElementById("CoordY").value;
                КоординатаУ = ЧислоТип.ПривестиЗначение(КоординатаУ);
```

```
                Кол = ТаблицаТочек.Количество();
                Если КоординатаХ > 0 И КоординатаУ > 0 И (Кол = 0 Или
КоординатаХ <> ТаблицаТочек[Кол - 1].Широта И КоординатаУ <> ТаблицаТочек[Кол -
1].Долгота) Тогда
```

```
                    НоваяСтрока = ТаблицаТочек.Добавить();
                    НоваяСтрока.Точка = "Точка" +
ТаблицаТочек.Количество();
```

```
                    НоваяСтрока.Широта = КоординатаХ;
                    НоваяСтрока.Долгота = КоординатаУ;
```

```
                    КонецЕсли;
```

```
                Исключение
                    КонецПопытки;
```

```
                КонецЕсли;
```

```
            Иначе
```

```

        Попытка
            ЧислоТип = Новый ОписаниеТипов("Число", Новый
КвалификаторыЧисла(15, 12));

            КоординатаХ =
Элементы.Эксплорер.document.getElementById("CoordX").value;
            КоординатаХ = ЧислоТип.ПривестиЗначение(КоординатаХ);

            КоординатаУ =
Элементы.Эксплорер.document.getElementById("CoordY").value;
            КоординатаУ = ЧислоТип.ПривестиЗначение(КоординатаУ);

            Кол = ТаблицаТочек.Количество();
            Если КоординатаХ > 0 И КоординатаУ > 0 И (Кол = 0 Или КоординатаХ
<> ТаблицаТочек[Кол - 1].Широта И КоординатаУ <> ТаблицаТочек[Кол - 1].Долгота) Тогда
                НоваяСтрока = ТаблицаТочек.Добавить();
                НоваяСтрока.Точка = "Точка" + ТаблицаТочек.Количество();
                НоваяСтрока.Широта = КоординатаХ;
                НоваяСтрока.Долгота = КоординатаУ;
            КонецЕсли;
        Исключение
            КонецПопытки;
        КонецЕсли;
КонецПроцедуры

&НаКлиенте
Процедура ПострениеПолигона(Команда)
    Если ТаблицаТочек.Количество() <= 1 Тогда
        Предупреждение("Недостаточно точек для построение полигона!");
        Возврат;
    КонецЕсли;

    ОчисткаКарты();

    ПостроитьПолигон();
КонецПроцедуры

&НаКлиенте
//Процедура выстраивает маршрут для
//
//Параметры:
//
Процедура ПостроитьПолигон()
    МассивТочек = "[";
    Для Каждого ТекТочка Из ТаблицаТочек Цикл
        Широта = формат(ТекТочка.Широта, "ЧРД=");
        Долгота = формат(ТекТочка.Долгота, "ЧРД=");
        МассивТочек = МассивТочек + "[" + Широта + "," + Долгота + "],";
    КонецЦикла;
    МассивТочек = Сред(МассивТочек, 1, СтрДлина(МассивТочек) - 1) + "];";

    Цвет16 = Получить16Цвет();

```

```
    Название = "Полигон";//вставить свое :)
    Элементы.Эксплорер.document.getElementById("WebClientOperation").value =
"createPolygon(" + МассивТочек + ", "" + Название + "", "" + Цвет16 + "");";
    Элементы.Эксплорер.document.getElementById("WebClient").click();
КонецПроцедуры
```

&НаСервере

```
//Функция возвращает значение случайного цвета в 16-ричном формате
```

```
//
```

```
//Параметры:
```

```
// нет
```

```
//Возвращаемое значение:
```

```
// Строка
```

```
Функция Получить16Цвет()
```

```
    Результат = "";
```

```
    Строка16 = "0123456789ABCDEF";
```

```
    ГСЧ = Новый ГенераторСлучайныхЧисел;
```

```
    Результат = "#";
```

```
    Для н = 1 По 6 Цикл
```

```
        м = ГСЧ.СлучайноеЧисло(1, 16);
```

```
        Результат = Результат + Сред(Строка16, м, 1);
```

```
    КонецЦикла;
```

```
    Возврат Результат;
```

```
КонецФункции
```

Выпускная квалификационная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

« ___ » _____ Г.

(подпись)

(Ф.И.О.)