

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
(наименование института полностью)

Кафедра «Прикладная математика и информатика»  
(наименование)

09.03.03 Прикладная информатика  
(код и наименование направления подготовки)

Бизнес-информатика  
(направленность (профиль))

## ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка информационной системы управления договорами туристического предприятия (на примере ООО «Туризм Европа»)»

Студент

В.В. Марковский

(И.О. Фамилия)

(личная подпись)

Научный  
руководитель

Н.Н. Казаченок

(ученая степень, звание, И.О. Фамилия)



Тольятти 2020



**Росдистант**  
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

## Аннотация

В выпускной квалификационной работе была разработана автоматизированная информационная система управления договорами туристического предприятия на примере ООО «Туризм Европа» средствами платформы WPF (Windows Presentation Foundation), являющейся частью .NET Framework, на языке программирования C# с использованием расширяемого языка разметки XAML для описания пользовательского интерфейса.

Объект исследования – деятельность, направленная на автоматизацию процесса управления договорами, формирования статистической и отчетной документации, которую можно вывести на печать.

Предмет исследования – автоматизация процесса управления договорами на составление турпакетов.

Методами исследования являются:

- методология анализа и моделирования бизнес-процессов IDEF0;
- методология объектно-ориентированного анализа и проектирования.

В процессе работы был проведен анализ предметной области и аналогов информационных систем. Проект системы разработан по методологии UML [3]. Разработаны и описаны физические и логические модели базы данных. Реализация программы выполнена в среде разработки Visual Studio на платформе NET Framework на языке программирования C#. Функционирует под управлением операционной системы Windows 7 и выше. База данных реализована за счёт встраиваемой СУБД SQLite, которая хранится на отдельном сервере.

В выпускной квалификационной работе 73 страницы, 44 рисунка, 5 таблиц, 29 литературных источников.

## Оглавление

Введение.....	4
Глава 1 Анализ туристического предприятия.....	7
1.1 Техничко-экономическая характеристика туристического предприятия.....	7
1.1.1 Характеристика предприятия ООО «Туризм Европа» .....	7
1.2 Концептуальное моделирование предметной области .....	9
1.2.1 Разработка и анализ модели бизнес-процесса «КАК ЕСТЬ»....	10
1.2.2 Анализ существующих разработок и обоснование выбора технологии проектирования.....	12
1.2.3 Определение критериев анализа .....	16
1.2.4 Сравнительная характеристика существующих разработок.....	18
1.3 Постановка задачи.....	19
1.4 Разработка модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ» .....	20
Выводы по главе 1 .....	23
Глава 2 Разработка и реализация проектных решений .....	24
2.1 Логическое моделирование деятельности турфирмы.....	24
2.1.1 Логическая модель и ее описание .....	24
2.1.2 Характеристика нормативно-справочной и входной оперативной информации .....	29
2.1.3 Характеристика выходной информации .....	36
2.2 Физическое моделирование АИС управления договорами на ООО «Туризм Европа».....	37
2.2.1 Выбор архитектуры АИС управления договорами на ООО «Туризм Европа».....	37

2.2.2 Функциональная схема проекта .....	39
2.2.3 Структурная схема проекта и описание программных модулей .....	41
2.3 Контрольный пример реализации проекта и его описание .....	43
Выводы по главе 2 .....	53
Глава 3 Оценка и обоснование экономической эффективности проекта .....	54
3.1 Выбор и обоснование методики расчета экономической эффективности .....	54
3.2 Расчет показателей экономической эффективности проекта.....	60
Выводы по главе 3 .....	67
Заключение .....	69
Список используемой литературы и источников .....	71
Приложение А – Печатные формы выходной документации .....	74
Приложение Б – Макеты и программная реализация продукта.....	79
Приложение В – Листинг программного кода модуля главной формы Win_dow.....	79

## Введение

Состояние мировой экономики находится в непосредственной созависимости от экономической деятельности её секторов, в том числе сектора туристического бизнеса. По мере развития стран и повышения качества жизни населения отрасль туризма увеличивает своё влияние на такие макроэкономические показатели как: формирование внутреннего валового продукта страны (ВВП), создание дополнительных внешнеэкономических связей внутри страны и за её пределами, обеспечение занятости населения и размещение дополнительных рабочих мест, предоставляя также спрос на смежные услуги транспорта, связи, строительства и другие.

Таким образом, сектор туризма, как посредник между потребителем-туристом и производителем услуги-принимающей страны, является важным фактором роста мировой экономики и зависит от устойчивого развития технологических решений отрасли.

В свою очередь, выбор функционала и средств разработки технологического решения находится в непосредственной зависимости от масштаба деятельности организации, располагаемых ресурсов и целей будущей разработки. Так, в настоящее время, при множестве программных решений на рынке разработок успех достигается путём сбалансированности между необходимым функционалом будущего технического решения, интуитивно понятным интерфейсом и целесообразностью капитальных вложений на разработку и внедрение.

Объектом исследования является деятельность, направленная на автоматизацию процесса управления договорами, формирования статистической и отчетной документации, которую можно вывести на печать.

Предмет исследования – автоматизация процесса управления договорами на составление турпакетов.

Таким образом, целью выпускной бакалаврской работы является разработка информационной системы управления договорами туристического предприятия.

Для достижения обозначенной цели работы, необходимо выполнение следующих задач:

- проведение анализа предметной области предприятия;
- проведение анализа бизнес-процессов, нуждающихся в оптимизации;
- проведение анализа существующих систем и выявление оснований для данной автоматизации;
- выполнение функционального проектирования информационной системы;
- составление характеристики входных и выходных данных системы;
- выполнение проектирования будущей базы данных системы;
- выполнение реализации спроектированной информационной системы.

Методами исследования являются:

- методологии анализа и моделирования бизнес-процессов IDEF0;
- методология объектно-ориентированного анализа и проектирования.

В выпускной квалификационной работе рассматриваются вопросы анализа, проектирования и разработки автоматизированной информационной системы, упрощающей подбор и оформление индивидуальных туристических туров посредством автоматизированного централизованного сбора данных в базу данных, своевременное формирование, хранение договоров и путевок с их последующим выводом на печать.

В состав выпускной квалификационной работы входит введение, три главы, заключение, список используемой литературы и приложения.

В первой главе описано концептуальное моделирование предметной области.

Во второй – логическое и физическое моделирование предметной области, а также продемонстрированы примеры реализации разрабатываемой системы и их описание.

В третьей главе проведены расчеты показателей экономической эффективности от внедрения и разработки, а также их необходимость и целесообразность для целей организации.

В заключении отражены степень и полнота решения ранее поставленных задач выпускной квалификационной работы.

В приложениях представлены печатные формы отчетной документации, а также программная часть реализации.

# Глава 1 Анализ туристического предприятия

## 1.1 Технико-экономическая характеристика туристического предприятия

### 1.1.1 Характеристика предприятия ООО «Туризм Европа»

Предметной областью бакалаврской работы является деятельность туристического предприятия.

Бакалаврская работа по разработке информационной системы выполняется на основе данных о бизнес-процессах ООО «Туризм Европа». Туристическая деятельность ООО «Туризм Европа» заключается в сопровождении временных выездов граждан по России и странам СНГ для различных целей: отдых, оздоровление, развлечения, изучение других культур, официальные и деловые поездки.

Организационная структура ООО «Туризм Европа» состоит из трёх основных отделов, представленных на рисунке 1.

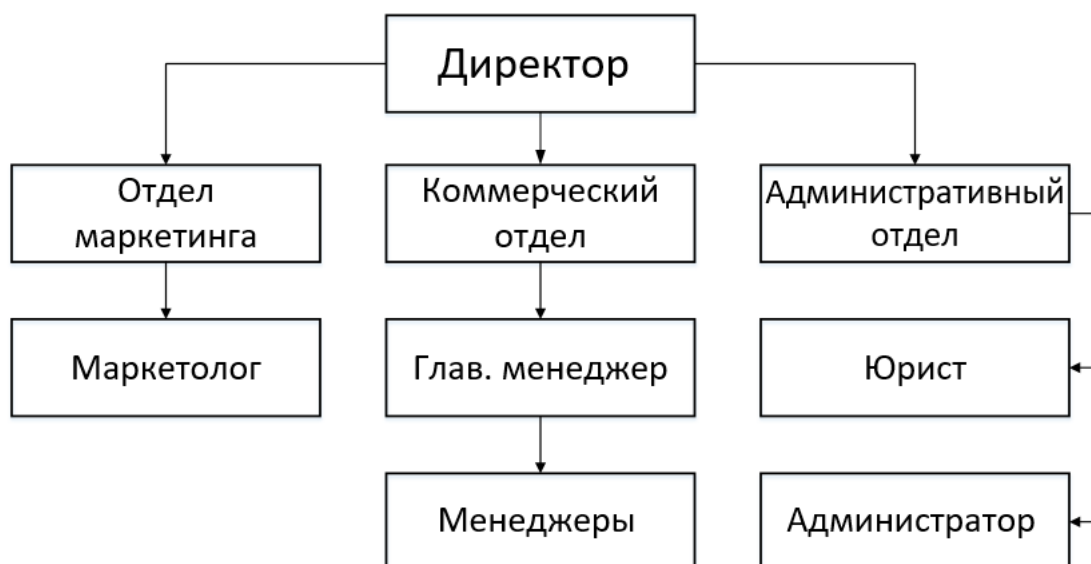


Рисунок 1 – Организационная структура ООО «Туризм Европа»



Организационная структура ООО «Туризм Европа» состоит из:

- отдел маркетинговых исследований отвечает за формирование, реализацию, результативность рекламных кампаний и их последующий анализ;
- коммерческий отдел или отдел по работе с клиентами занимается коммерческим сопровождением предоставляемых услуг;
- административный отдел отвечает за потоковое бесперебойное функционирование текущей деятельности предприятия.

Перейдем к рассмотрению внутренней деятельности отдела по работе с клиентами, как непосредственного объекта исследования на ООО «Туризм-Европа».

Первым этапом выступает взаимодействие отдела по работе с клиентами и потенциального Заказчика.

За сопровождение потенциального клиента до завершающей стадии двухсторонних отношений, заключения договора, отвечает менеджер. Первоначально проводится сбор предпочтений о предстоящем туре у потенциального заказчика, где важной ролью является возможность выбора под индивидуальное пожелание клиента, скорость работы и оперативность. Промежуточным документом выступает формат «технического задания» для подборки индивидуального тура, обрабатываемого в дальнейшем следующим отделом оформления. От точности его формирования зависит оперативность и предоставление качественной услуги в оговоренный срок. Результат проведенной работы менеджера - договор об оказании услуг ООО «Туризм-Европа».

После завершения оформления менеджером комплекта документов, этот комплект поступает на согласование к юристу, который либо одобряет его, либо передает обратно в коммерческий отдел на изменение, внесенных им замечаний и правок. После проведения соответствующих правок маркетинговый отдел проводит и размещает туристическую путевку со всеми

прилегающими договорами. В большей перспективе задачами маркетинговых исследований являются контроль реализованных туров, учёт результата предоставленной услуги, формирование базы постоянных клиентов и их предпочтений.

Целью деятельности ООО «Туризм Европа» является извлечение и максимизация прибыли при высокоэффективном и рентабельном использовании ресурсов. На данный момент анализируемая организация располагает узким продуктовым портфелем готовых туристических пакетов на базе MS Office, слабой диверсификацией услуг, устаревшими технологическими решениями, что повышает риск неконкурентоспособности и устойчивости его деятельности, а также несёт потери в размере упущенной прибыли.

В настоящее время для развития деятельности ООО «Туризм-Европа» жизненно необходимо внедрять технологические решения с индивидуальным подходом к каждому потенциальному клиенту [2]. Это динамическая обработка туристических предпочтений, индивидуальный подход к клиенту при бронировании, средства централизованной автоматической работы с клиентами, быстрая и безошибочная комплектация пакета документов.

## **1.2 Концептуальное моделирование предметной области**

В анализируемый период деятельность ООО «Туризм-Европа» представляет собой ручной процесс документооборота и обработки заказов потенциальных клиентов без внедрения индивидуального подхода к выбору тура ввиду невозможности оперирования большими объёмами входной информации. Однако, установленная процедура работы занимает в разы больше времени, нежели если бы за данные процессы отвечала АИС. К тому же, возрастает вероятность ошибок, связанных с человеческим фактором.

Отсюда возникает необходимость в устранении такого рода рисков и снижении временных затрат на оформление и обработку заказов клиентов,

составление комплекта необходимых документов, что в дальнейшем приведет к минимизации издержек и увеличению продаж ООО «Туризм-Европа».

Таким образом, для создания автоматического учёта входных данных и электронного документооборота необходимо разработать АИС, которая отвечать следующим функциям:

- автоматическое заполнение шаблонов документов на основании существующих данных;
- добавление клиентов и их персональных данных;
- централизованный учет клиентов и состояния их оформления;
- учет сопроводительной документации по каждому клиенту и сроков выполнения;
- автоматическое составление комплекта документов;
- гибкий параметризованный поиск по заданным критериям;
- импорт существующей в MS Excel базы данных.

Технологией создания модели в данном случае выступит структурный подход по причинам простоты обучения и точности описания.

В свою очередь, анализ и моделирование автоматизируемых бизнес-процессов будет осуществляться на основании технологии IDEF0 [4].

### **1.2.1 Разработка и анализ модели бизнес-процесса «КАК ЕСТЬ»**

На основе структурного подхода к созданию концептуальной модели была составлена модель бизнес-процесса «КАК ЕСТЬ», представленная на рисунке 2, с использованием наиболее популярной и доступной методологии IDEF0 [9].

Входную информацию представляют хранящиеся в базе данных обработанные заявки, полученные в ходе обращения потенциального клиента к менеджеру коммерческого отдела.

Выходной информацией являются, полученные в результате поиска по выбранным критериям, индивидуальные туристические путёвки,

сформированные на их основании и, соответственно, их вывод на печать вместе с договорами на оказание и реализацию услуг.

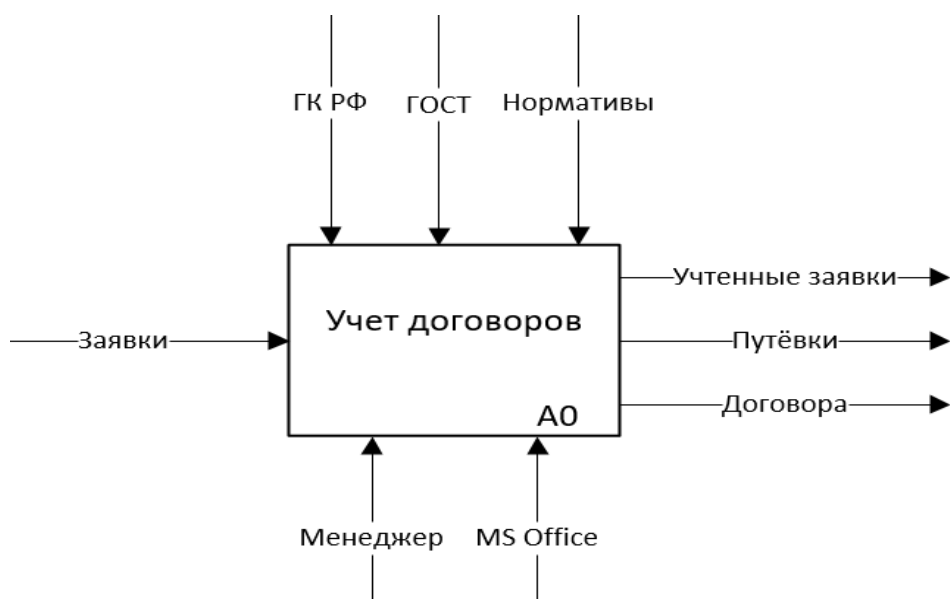


Рисунок 2 – Контекстная IDEF0-диаграмма бизнес-процесса управление договорами «КАК ЕСТЬ» (0-й уровень)

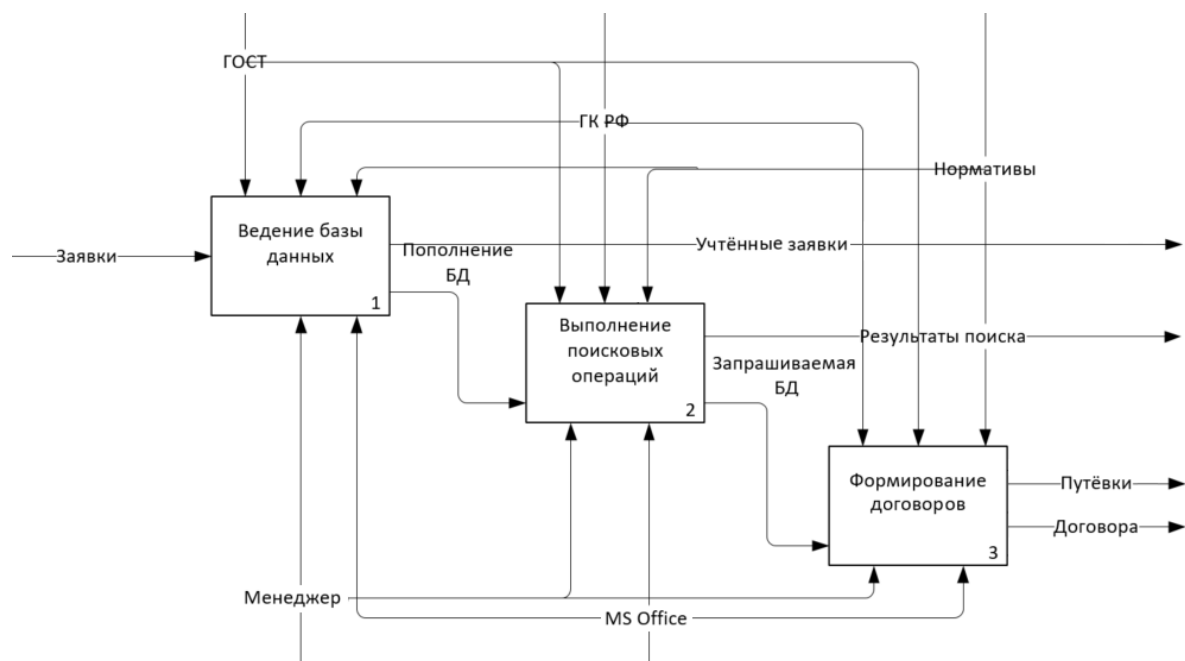


Рисунок 3 – IDEF0-диаграмма декомпозиции бизнес-процесса управление договорами «КАК ЕСТЬ» (1-й уровень)

К управляющим воздействиям относятся нормативы по работе с принятыми заявками и ведению их учета – ГОСТ и ГК РФ.

В роли исполнителя бизнес-процесса выступает менеджер коммерческого отдела, использующий MS Office при работе с входными данными.

Анализ данной модели показал, что существующая система не обладает необходимой функциональностью для оперативного оформления клиента от существующего заказа до формирования двухстороннего договора индивидуального тура.

### **1.2.2 Анализ существующих разработок и обоснование выбора технологии проектирования**

На рынке программных продуктов существует достаточный выбор решений автоматизации бизнес-процессов. В основном, разработчики размещённых вариантов автоматизации используют универсальный вид и способ учёта данных, что подойдёт не каждому потребителю готовой программы [10]. Создание программного решения под индивидуальные цели у сторонней организации потребует весомых вложений, изымаемых из оборота организации, что экономически целесообразно лишь крупным компаниям. Поэтому ООО «Туризм-Европа» может попробовать внедрить существующие решения или разрабатывать собственный проект автоматизации процессов.

В целях оптимизации вложений в разработку наиболее эффективного и отвечающего требованиям пользователя инструмента учёта заказов клиентов и оформления документации с последующей записью в реестр, проведем анализ размещённых на рынке программных продуктов и определим наиболее оптимальный способ разработки [11].

Сформируем критерии отбора эффективного решения разработки:

- готовое решение автоматизации должно соответствовать максимально возможному числу поставленных требований;

- должна быть возможность доработки программного решения под специфические требования, не учтённые по умолчанию. При этом, стоит учесть и дешевизну данной доработки;
- конечная стоимость использования готового решения автоматизации, учитывающая стоимость официального оформления программы по автоматизации и вспомогательных программ, требуемых для потоковой и бесперебойной работы самой системы, а также стоимость услуг установки и дальнейшему обслуживанию.

В процессе выбора максимально выполняющего требования программного обеспечения был изучен ряд отечественных программных решений, представленных в таблице 1.

Аналитика: Документооборот – система, выполняющая функцию автоматизации документооборота на основе платформы 1С: Предприятие 8.

Конфигурация системы управления документооборотом «Аналитика: Документооборот» представлена на рисунке 4.

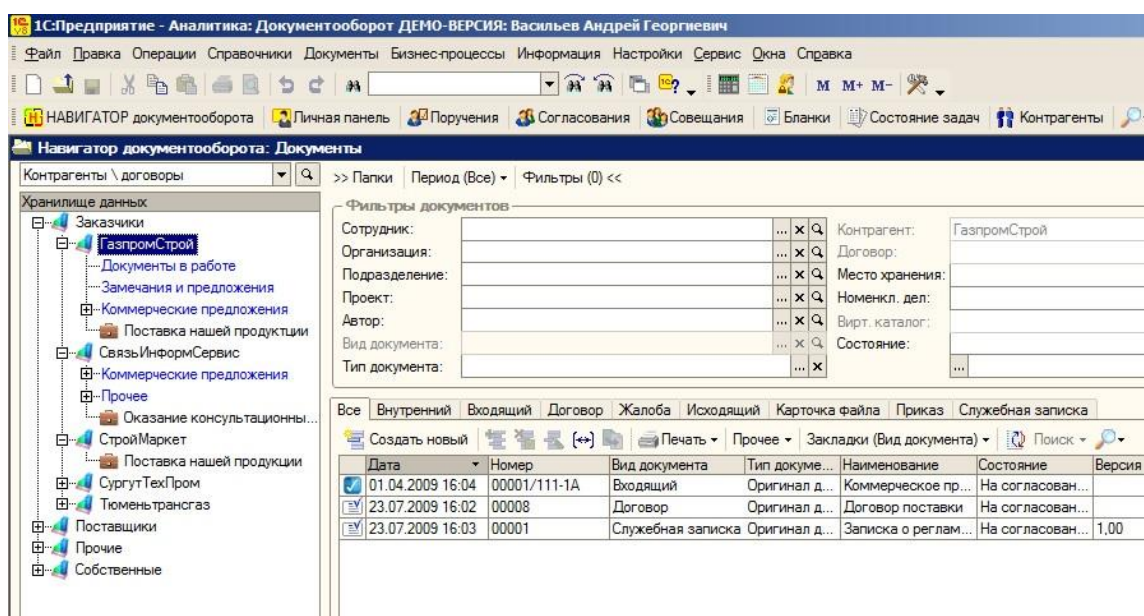


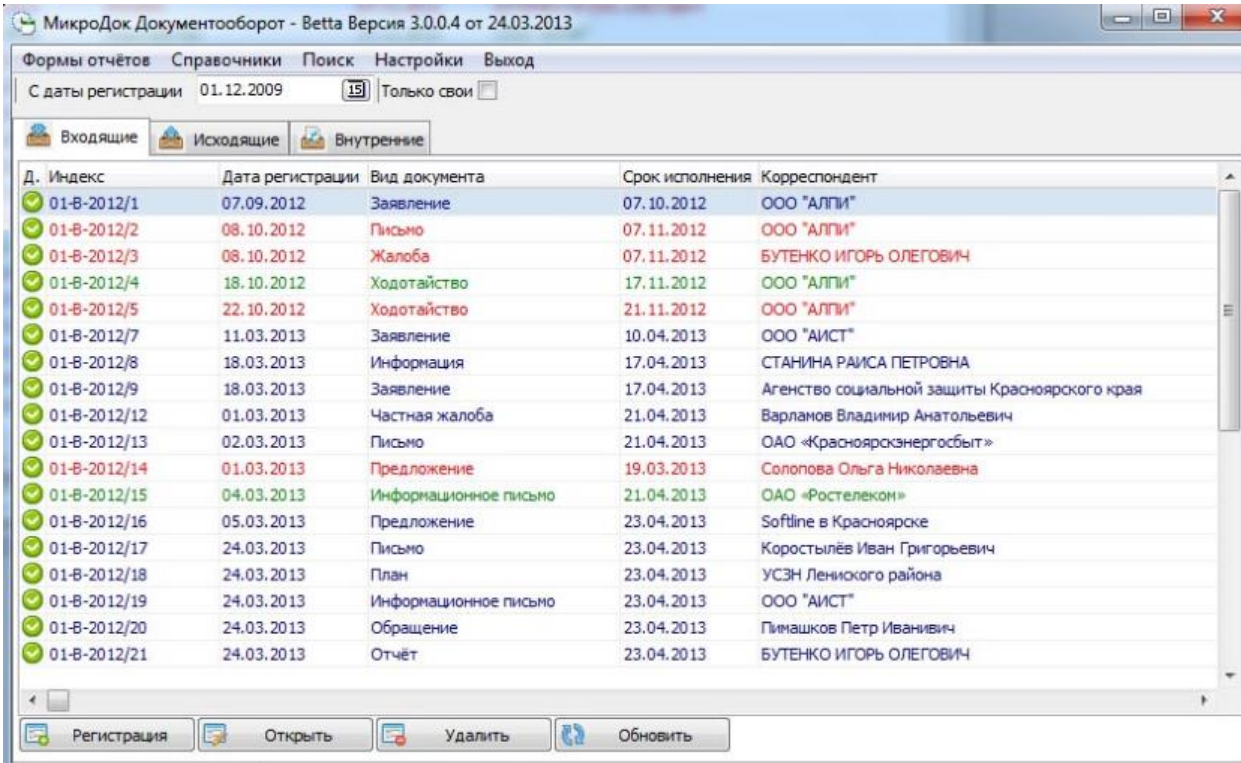
Рисунок 4 – Вид документооборота на базе «Аналитика: Документооборот» на основе платформы 1С: Предприятие 8

Плюсы данной программы автоматизации бизнес-процессов:

- учет документов и ведение архива, нормативно – справочной информации;
- согласование документа ответственными лицами;
- ведение номенклатуры;
- интеграция с 1С, MS Office.

Недостатками данного варианта автоматизации выступают узкий, с одной стороны функционал, ограничивающийся исключительно работой с документооборотом и излишний в этой же области, так как помимо работы с договорами на предприятии существует еще ряд бизнес-процессов, нуждающихся в автоматизации.

Система «МикроДок» – представляет собой систему для регистрации и контроля документооборота в организации. Конфигурация данного программного решения представлена на рисунке 5.



МикроДок Документооборот - Beta Версия 3.0.0.4 от 24.03.2013

Формы отчётов Справочники Поиск Настройки Выход

С даты регистрации 01.12.2009 [15] Только свои

Входящие Исходящие Внутренние

Д. Индекс	Дата регистрации	Вид документа	Срок исполнения	Корреспондент
01-В-2012/1	07.09.2012	Заявление	07.10.2012	ООО "АЛПИ"
01-В-2012/2	08.10.2012	Письмо	07.11.2012	ООО "АЛПИ"
01-В-2012/3	08.10.2012	Жалоба	07.11.2012	БУТЕНКО ИГОРЬ ОЛЕГОВИЧ
01-В-2012/4	18.10.2012	Ходотайство	17.11.2012	ООО "АЛПИ"
01-В-2012/5	22.10.2012	Ходотайство	21.11.2012	ООО "АЛПИ"
01-В-2012/7	11.03.2013	Заявление	10.04.2013	ООО "АИСТ"
01-В-2012/8	18.03.2013	Информация	17.04.2013	СТАНИНА РАИСА ПЕТРОВНА
01-В-2012/9	18.03.2013	Заявление	17.04.2013	Агентство социальной защиты Красноярского края
01-В-2012/12	01.03.2013	Частная жалоба	21.04.2013	Варланов Владимир Анатольевич
01-В-2012/13	02.03.2013	Письмо	21.04.2013	ОАО «Красноярскэнергообл»
01-В-2012/14	01.03.2013	Предложение	19.03.2013	Солопова Ольга Николаевна
01-В-2012/15	04.03.2013	Информационное письмо	21.04.2013	ОАО «Ростелеком»
01-В-2012/16	05.03.2013	Предложение	23.04.2013	Softline в Красноярске
01-В-2012/17	24.03.2013	Письмо	23.04.2013	Коростылёв Иван Григорьевич
01-В-2012/18	24.03.2013	План	23.04.2013	УСЗН Ленинского района
01-В-2012/19	24.03.2013	Информационное письмо	23.04.2013	ООО "АИСТ"
01-В-2012/20	24.03.2013	Обращение	23.04.2013	Пимашков Петр Иванович
01-В-2012/21	24.03.2013	Отчёт	23.04.2013	БУТЕНКО ИГОРЬ ОЛЕГОВИЧ

Регистрация Открыть Удалить Обновить

Рисунок 5 – Вид документооборота на базе «МикроДок»

Плюсы данной программы автоматизации бизнес-процессов:

- возможность использования программы несколькими пользователями;
- контроль входящих и исходящих документов;
- возможность назначения ответственных к каждому процессу.

Недостатками данного варианта автоматизации выступают автоматизация конечного этапа оформления клиента, что говорит о частичной автоматизации необходимого функционала проектного варианта разработки АИС на ООО «Туризм Европа».

Система «Канцлер» – это стандартизированное решение автоматизации документооборота и архивной обработки входящих и исходящих данных. Конфигурация представлена на рисунке 6.

Регистрационный №	Дата регистрации (создания)	Краткое содержание	Приложение
24			
6			
1			
4			
2			
3			
5			
12			
4			

Рисунок 6 – Вид документооборота на базе «Канцлер»

Плюсы данной программы автоматизации бизнес-процессов:

- возможность использования программы несколькими пользователями;
- централизованное хранение баз данных;
- возможность заказной разработки под индивидуальные цели и сопровождения.



Данная система выступает наиболее подходящей для целей АИС управления договорами, так как имеет возможность разработки по индивидуальному техническому заданию, что удовлетворит объёму функционала будущей АИС управления договорами, но в то же время будет заключать в себе стоимость, в разы превышающую стоимость, выделенного на разработку бюджета.

### **1.2.3 Определение критериев анализа**

Для выявления требований при проектировании информационной системы необходимо выявить недостатки существующих процессов, подлежащих автоматизации [5].

Бизнес-процессы верхнего уровня, входящие в состав бизнес-процесса «Управление договорами», обладают следующими недостатками:

- риск допуска ошибок и неточностей;
- высокая трудоемкость обработки информации;
- высокие временные затраты;
- низкая оперативность, снижающая качество управления объектом.

Данные недостатки существуют за счет необходимости ручного составления документации, отчетности, а также выполнении итоговых расчетов в подобного рода документах. Что влечет за собой определенные риски, которые в дальнейшем будут отражаться в низком качестве документов, неточностях и ошибках при проведении расчетов и составлении документации.

Недостатки могут быть устранены, а риски снижены путем внедрения новой автоматизированной информационной системы.

В итоге, имея в своей базе данных информацию за определенный период, АИС управления договорами сможет формировать итоговые отчеты с безошибочно подсчитанными итоговыми значениями. Такая возможность компенсирует недостаток составления статистики вручную, а также устраняет риск любых неточностей и расхождений в данных.

Таким образом, все вышеописанные недостатки являются существенной причиной для автоматизации бизнес-процесса «Управление договорами».

После выявления недостатков, были сформулированы требования, представленные в таблице 1, которым должна соответствовать будущая АИС управления договорами.

Таблица 1 – Требования к АИС управления договорами ООО «Туризм Европа»

№	Описание
1	Система должна иметь возможность централизованного хранения данных и осуществления гибкого поиска по реестру договоров, а также возможность согласования документа с юристом.
2	Система должна иметь возможность вывода на печать туристической путевки и договоров в установленной форме.
3	Система должна иметь возможность автоматически заполнять соответствующие поля документов на основании существующих данных в системе.
4	Система должна позволять предоставлять актуальный курс валют для расчёта стоимости путёвки.
5	Система должна иметь возможность регистрации и авторизации сотрудника в системе.
6	Система должна позволять осуществлять подбор индивидуальных туристических путёвок по следующим критериям: Страна; Курорт; Ценовой диапазон; Период пребывания; Класс отеля; Питание; Дата вылета/прилета.
7	Система должна позволять пользователю добавлять, изменять и удалять информацию в системе о: Договорах; Забронированных путевках; Операторах; Курортах; Авиакомпаниях; Услугах отеля; Клиентах; Сотрудниках.
8	Элементы интерфейса системы должны представлять собой четкую структуру и быть интуитивно понятны любому пользователю.

После того, как требования были сформулированы, необходимо провести анализ уже существующих информационных решений с целью выявления целесообразности разработки данной проектируемой АИС управления договорами.

#### 1.2.4 Сравнительная характеристика существующих разработок

После проведения анализа основных аналогов проектируемого автоматизированного программного решения, необходимо провести сравнительный анализ на предмет соответствия требованиям к системе, сформулированным в п.1.2.3. Результаты сравнения представлены в таблице 2.

Таблица 2 – Сравнительная характеристика основных отечественных фирм-разработчиков АИС электронного автоматического документооборота

Требование/Аналог	«Аналитика: документооборот»	«МикроДок»	«Канцлер»
Гибкий поиск по реестру договоров, согласование с юристом	+	+	+
Вывод на печать документов	+	+	+
Автоматическое заполнение полей	+	+	+
Актуальный курс валют для расчёта стоимости	–	–	–
Регистрация и авторизация сотрудника	–	–	–
Подбор индивидуальных туров по критериям	–	–	–
Манипуляции с данными пользователем в системе	–/+	–/+	+
Чёткая интуитивно понятная структура	–/+	–/+	+
Итого	4 из 9	4 из 9	5 из 9

Исходя из проведенного анализа, следует вывод, что приобретение одной из описанных выше систем является нецелесообразной в рамках решения поставленных задач автоматизации.

Таким образом, наиболее выигрышным решением является разработка программного обеспечения с нуля, которое будет учитывать все особенности и включать в себя весь функционал, необходимый для автоматизации

основных бизнес-процессов и в полной мере отвечающий выявленным требованиям.

Принимая во внимание итоги проведенного анализа, необходима детальная постановка задач для целенаправленного проектирования и разработки АИС управления договорами.

### **1.3 Постановка задачи**

Исходя из анализа, проведенного в предыдущем разделе, можно с уверенностью сказать, что ни одна из существующих информационных систем не удовлетворяет всем требованиям, представленным к разработке данной информационной системы. Следовательно, разработка автоматизированной информационной системы управления договорами туристической фирмы является целесообразным в силу того, что, данная информационная система будет включать в себя весь необходимый для организации функционал.

Назначением АИС управления договорами является:

- централизованное хранение, учёт и оформление персональных данных о клиентах;
- автоматический электронный документооборот;
- автоматический расчёт стоимости тура;
- возможность отслеживания сроков оформления документов;
- упрощение идентификации клиента и сопутствующих данных по личному номеру клиента;
- возможность простого поиска при минимально достаточной информации;
- бронь заявок внутри программы;
- импорт существующей в Excel базы данных.

В результате формализации общих функциональных требований к проектируемой АИС управления договорами, была поставлена следующая задача: автоматизировать основной бизнес-процесс «Управления договорами», и, как следствие, оптимизировать время, затрачиваемое на поиск наиболее подходящего туристического пакета, ведение статистики и соответствующей документации, а также минимизировать количество ошибок и рисков, связанных с человеческим фактором.

Реализация проектируемой информационной системы будет производиться на платформе .NET Framework с использованием технологии WPF (Windows Presentation Foundation).

Разрабатываемая информационная система должна обладать всеми функциями, описанными в пункте 1.2.

Кроме того, система обязана соответствовать требованиям, описанным в пункте 1.2.3.

Целью выпускной квалификационной работы является разработка информационной системы, которая будет автоматизировать деятельность управления договорами туристического предприятия на выбранной платформе.

Для хранения и работы с данными в АИС управления договорами будет использована СУБД RDBMS.

ИС разрабатывается на платформе WPF (Windows Presentation Foundation), являющейся частью .NET Framework, на языке программирования C# с использованием расширяемого языка разметки XAML для описания пользовательского интерфейса [19].

#### **1.4 Разработка модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ»**

Предполагается, что после внедрения проектируемой АИС управления договорами будет реализована автоматизация процесса «Управление договорами».

Следовательно, система будет выступать в роли механизма-исполнителя рассматриваемого бизнес-процесса.

Контекстная модель «КАК ДОЛЖНО БЫТЬ» («ТО-ВЕ»), является основой для технического создания на проектирование и реализацию АИС управления договорами, а также отображает предполагаемое состояние предметной области после внедрения [21].

Основываясь на анализе, проведенном в данной главе, были сформулированы общие бизнес-требования к проектируемой АИС управления договорами:

- автоматическое заполнение шаблонов документов на основании существующих данных;
- добавление клиентов и их персональных данных;
- централизованный учет клиентов и состояния их оформления;
- учет сопроводительной документации по каждому клиенту и сроков выполнения;
- автоматическое составление комплекта документов;
- гибкий параметризованный поиск по заданным критериям;
- осуществление вывода на печать сформированных путевок и договоров;
- экспорт существующей документации.

С учетом вышеописанных требований, была разработана концептуальная модель «КАК ДОЛЖНО БЫТЬ» («ТО-ВЕ»), представленная на рисунке 7.

Все новые элементы и изменения обозначены красным цветом. А также, все операции, совершаемые ранее менеджером вручную, в дальнейшем будут производиться на базе АИС управления договорами.

Сравнивая её с моделью «КАК ЕСТЬ» («AS-IS»), можно заметить добавление нового механизма – «АИС».

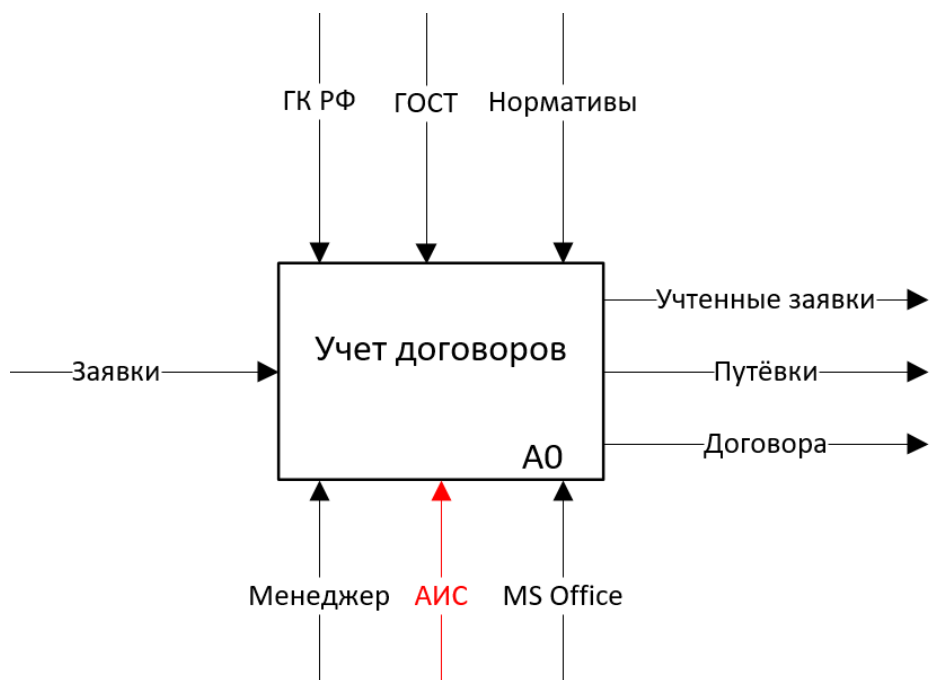


Рисунок 7 – Контекстная IDEF0-диаграмма бизнес-процесса управления договорами «КАК ДОЛЖНО БЫТЬ» (0-й уровень)

На рисунке 8 представлена декомпозиция данной модели.

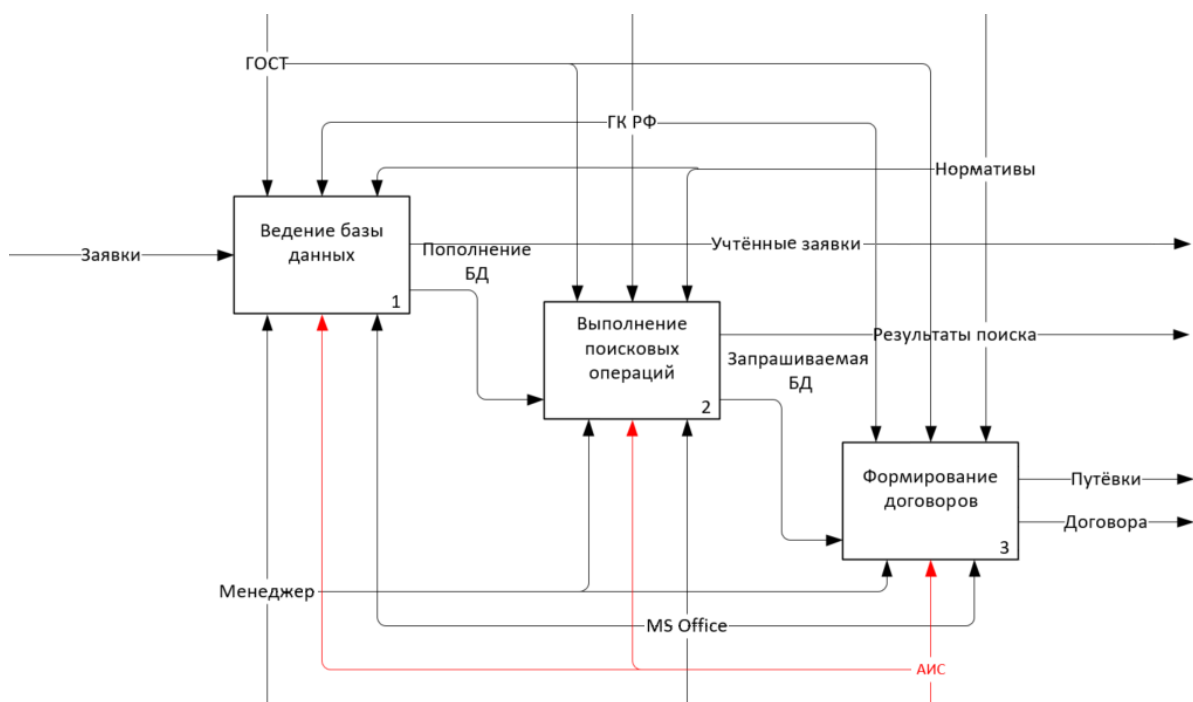


Рисунок 8 – IDEF0-диаграмма декомпозиции бизнес-процесса управления договорами «КАК ДОЛЖНО БЫТЬ» (1-й уровень)

Таким образом, оптимизация автоматического электронного документооборота происходит с помощью разработки и последующего внедрения АИС управления договорами, которая полностью удовлетворяет заявленным требованиям.

### **Выводы по главе 1**

В первой главе был произведен анализ туристической деятельности ООО «Туризм-Европа» и анализ существующих бизнес-процессов. Была составлена модель IDEF0 и её декомпозиция, отображающие процессы обработки заказа клиента и управления договорами, на основе чего были выявлены недостатки существующих бизнес-процессов, которые послужили обоснованием для разработки проектируемой АИС управления договорами.

Были сформулированы требования к проектируемой АИС управления договорами и проведена сравнительная характеристика существующих систем, в ходе которой было выявлено, что основным недостатком известных решений по автоматизации бизнес-процессов является их высокая стоимость, а также избыточный или, наоборот, отсутствующий функционал.

После полного анализа туристической деятельности ООО «Туризм-Европа», можно приступать к логическому и физическому проектированию автоматизированной системы управления договорами ООО «Туризм-Европа».



## Глава 2 Разработка и реализация проектных решений

### 2.1 Логическое моделирование деятельности турфирмы

#### 2.1.1 Логическая модель и ее описание

Взяв за основу структурно-функциональную модель IDEF0 была сформирована диаграмма прецедентов UML бизнес-процесса «Управление договорами», представленная на рисунке 9, которая позволит получить более подробное описание взаимодействия проектируемой системы с любыми внешними или внутренними объектами, такими как – пользователи или другие системы [12].

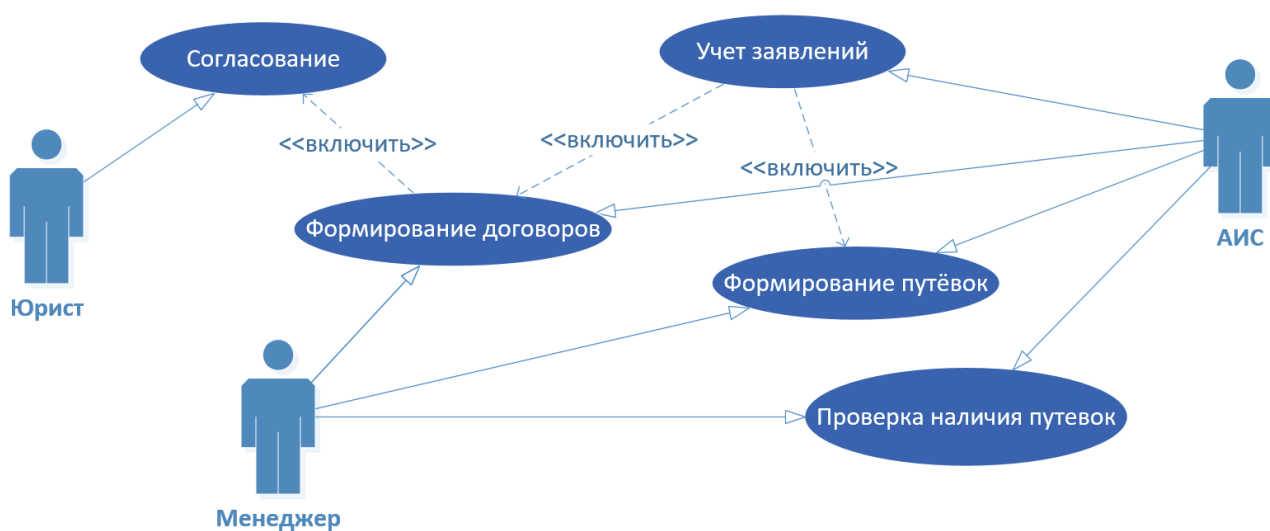


Рисунок 9 – Диаграмма прецедентов

Ниже представлено описание вариантов использования со стороны актёра «Менеджер».

Вариант использования: Формирование договоров.

Краткое описание. Даёт успешно авторизовавшемуся в системе сотруднику с соответствующим уровнем доступа возможность сформировать нужный ему документ на основе учтенных заявок.

Предусловие. Успешная авторизация пользователя в системе и требуемый уровень доступа.

Основной поток событий:

- а) на экране главное меню со сгруппированными вкладками,
- б) менеджер переходит во вкладку «Клиент», выбирает нужного клиента и нажимает на кнопку «Добавить бронь»;
- в) осуществляется переход на вкладку «Бронь»,
- г) данные выбранного клиента переносятся автоматически,
- д) менеджер заполняет оставшиеся поля и пользуется выпадающими списками, которые отображают данные в соответствии с предыдущим выбором. Пример такого заполнения можно увидеть на рисунке 27;
- е) при нажатии на кнопку «Сохранить» система сохраняет обработанный заказ в базе данных и отображает её на вкладке «Заказы»:
  - 1) менеджер выводит полученную путевку на печать, после чего программа перенаправляет менеджера в Excel, в котором данные печатного варианта путевки уже заполнены автоматически. После осуществления печати, Менеджер сохраняет документ и возвращается к программе;
  - 2) менеджер создает договор на оказание услуг клиенту, после чего программа также перенаправляет менеджера в Excel, в котором автоматически заполнены данные договора на оказание услуг для клиента. После создания договора он помечается статусом «Открыт» и отправляется юристу для дальнейшего согласования деталей по договору, после чего он либо меняет статус на «Подготовлен» и выводится на печать для дальнейшего его подписания, либо «На изменении» и ожидается внесение правок. После осуществления печати,

менеджер сохраняет документ и имеет возможность вернуться к работе АИС управления договорами;

3) менеджер создает договор на оказание услуг оператором – проделывает те же самые действия, но уже с договором на оказание услуг оператором;

ж) вариант использования успешно завершен:

4) менеджер переходит на форму печати документа.

Таким образом, было описано взаимодействие сотрудника с учетными заявками на бронирование туров.

Следующим этапом разработки логической модели системы является составление и описание диаграммы классов [11].

Результат построения ER-диаграммы классов проектируемой АИС управления договорами представлен на рисунке 10.

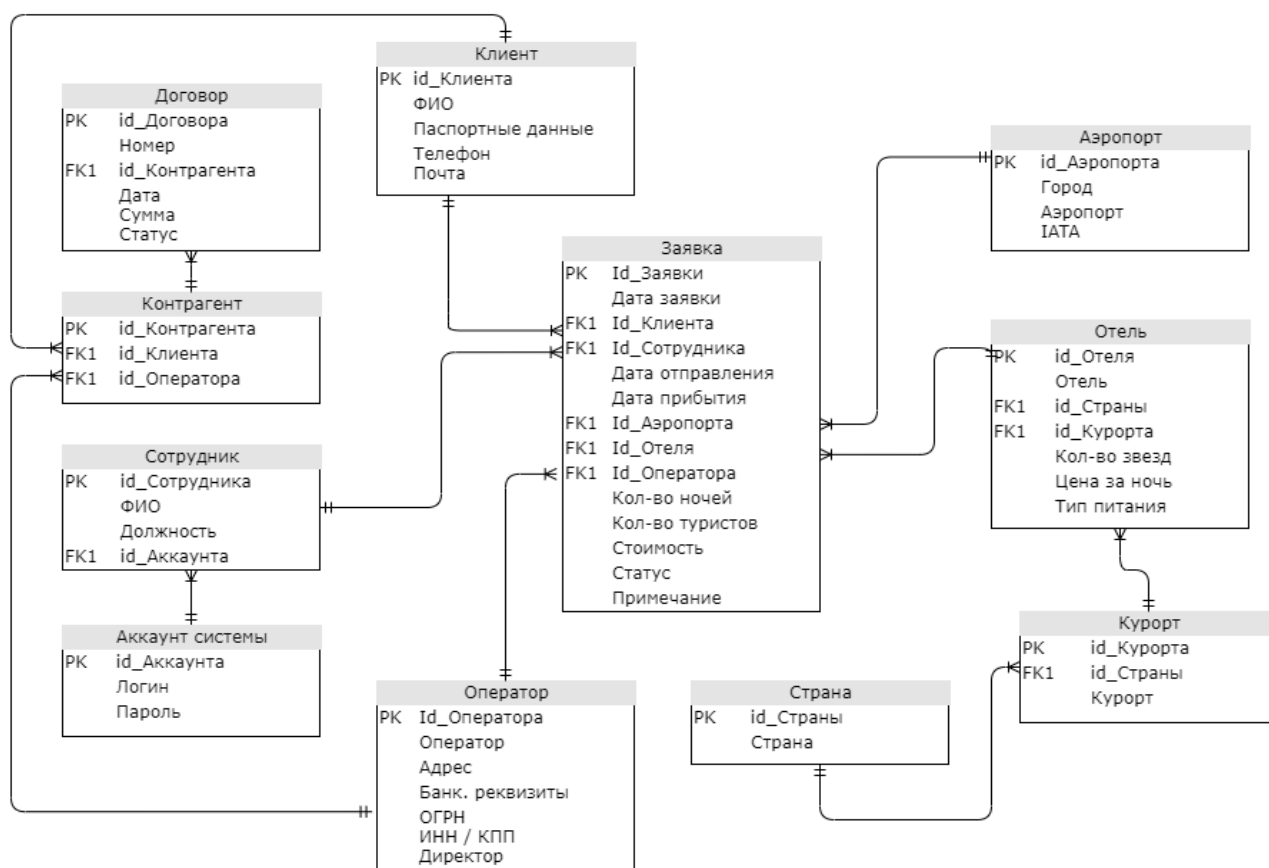


Рисунок 10 – ER-диаграмма классов

Диаграмма состоит из 11 классов сущностей, каждый из которых соответствует таблицам в базе данных, тем самым характеризуя определенный объект своими атрибутами, связями и операциями между другими объектами.

Рассмотрим спецификацию представленной ранее диаграммы классов в отдельности:

- «заявка» – ключевой класс объектов, в состав которого входит вся необходимая информация о выбранном туре клиентом, которую можно править, а также данный класс позволяет получить расчет итоговой стоимости;
- «сотрудник» – класс объектов-сотрудников туристической фирмы, отвечающий за ведение данных в базе данных АИС управления договорами;
- «аккаунт системы» – класс, хранящий данные для авторизации сотрудников в системе;
- «отель» – представляет основную информацию об отеле, также связан с классами «Курорт» и «Страна», в которых отражены соответствующие данные, которые можно менять;
- «аэропорт» – соответственно хранит данные об аэропортах, а также коммерческий код IATA для более удобного поиска;
- «клиент» и «оператор» – хранят общие, а также паспортные и реквизитные данные для дальнейшего формирования документации;
- «договор» и «контрагент» – договор формируется на основе данных контрагентов, которые в свою очередь могут содержать данные как юридических лиц (туроператоров), так и физических (туристов).

Самым распространенным типом связи между классами является тип «один-ко-многим».

Диаграмма классов является основой для моделирования отношений между прецедентами и их атрибутами.

Впоследствии на основании данной диаграммы классов была построена диаграмма последовательности, отображённая на рисунке 11.

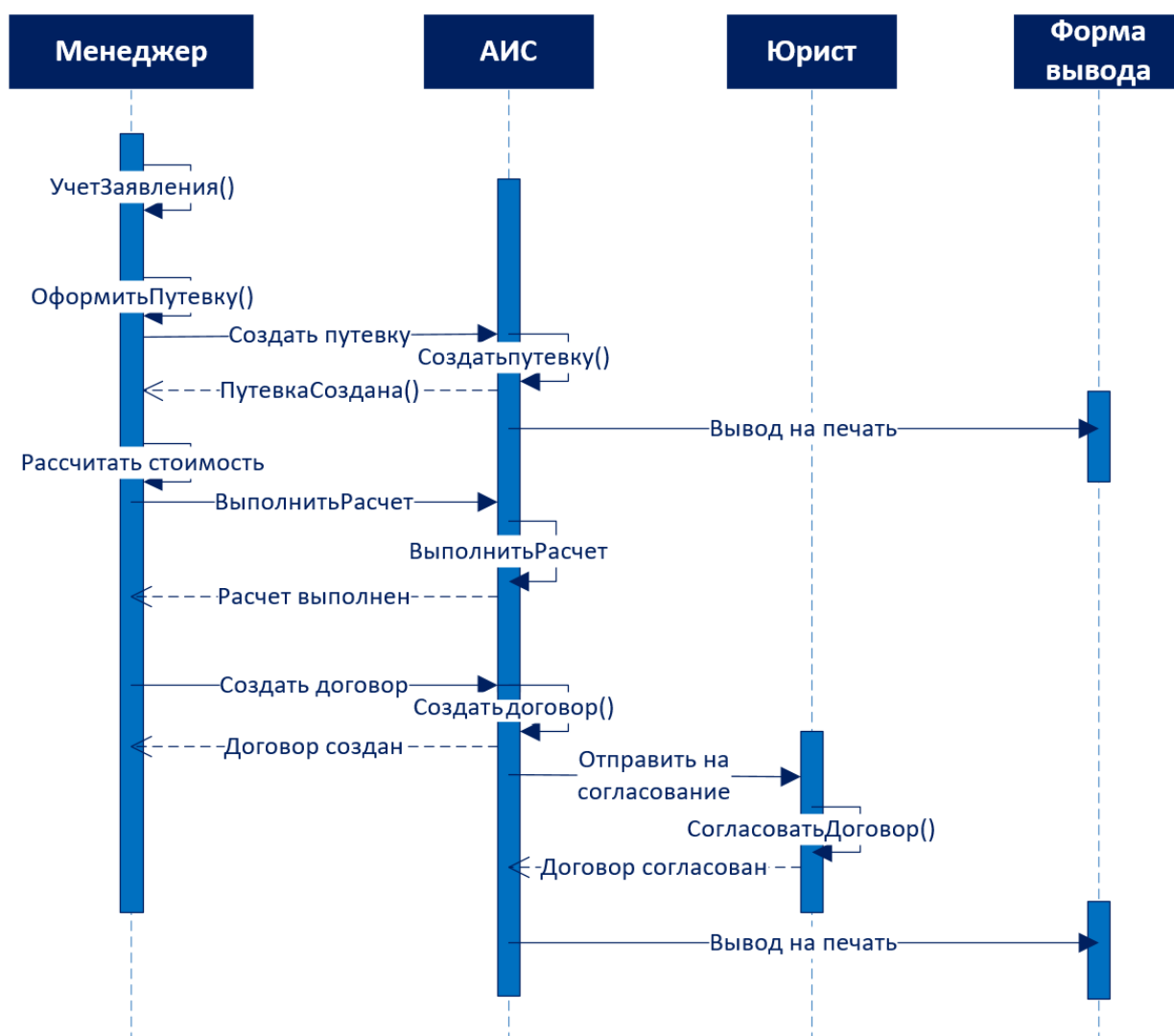


Рисунок 11 – Диаграмма последовательности

Описание процесса:

В любой отрезок времени объект Менеджер обращается к объекту АИС для оформления путевки по учтенной заявке, после чего, АИС заносит паспортные и контактные данные о клиенте, сведения о бронируемом отеле, количестве проживающих, дате отправления и выбранном операторе в базу данных, формирует на основании полученных данных путевку, получает при

этом автоматический расчет стоимости с учетом актуального курса доллара/евро и количества отдыхающих, и далее уже формирует договор об оказании услуг для клиента, и договор на реализацию туристического продукта туроператора.

Только что созданные договора имеют статус «Открыт» и отправляются юристу на согласование, где, после проверки, договор помечается статусом «Подготовлен», в случае если он не нуждается в доработке, иначе если требуются исправления - статусом «На изменение». После исправления замечаний юриста, если таковые имеются, договор отправляется в реестр договоров и выводится на печать для дальнейшего подписания.

Таким образом, были проанализированы и определены способы, характер и последовательность взаимодействия объектов в процессе вариантов использования. Следующим этапом является анализ информационного обеспечения системы.

### **2.1.2 Характеристика нормативно-справочной и входной оперативной информации**

Так как для организации хранения данных была выбрана реляционная база данных SQLite, а для визуальной работы с ней инструмент – Database Browser, нельзя не упомянуть о возможности импорта БД.

В случае с турфирмой ООО «Туризм Европа» это особенно актуально по причине того, что сотрудникам организации, отвечающим за процесс «Управления договорами» не придется переносить существующие данные из Excel.

Для этого достаточно иметь файл формата .csv, как представлено на рисунке 12.

После необходимо выбрать опцию импорта в DB Browser, дать таблице название, как отражено на рисунке 13.

В результате чего данные будут успешно перенесены в общий файл БД для взаимодействия с разрабатываемой АИС управления договорами.

	A	B	C	D	E	F	G	H
1	1.	Анапа	<a href="#">Витязево</a>	AAQ				
2	2.	Абакан	<a href="#">Абакан</a>	ABA				
3	3.	Ачинск	<a href="#">Ачинск</a>	ACS				
4	4.	Алдан	<a href="#">Алдан</a>	ADH				
5	5.	Адлер	<a href="#">Адлер - Сочи</a>	AER				
6	6.	Амдерма	<a href="#">Амдерма</a>	AMV				

Рисунок 12 – Пример таблицы БД в Excel

Импортировать файл в формате CSV

Имя таблицы:

Имена столбцов в первой строке:

Разделитель полей:

Символ кавычки:

Кодировка:

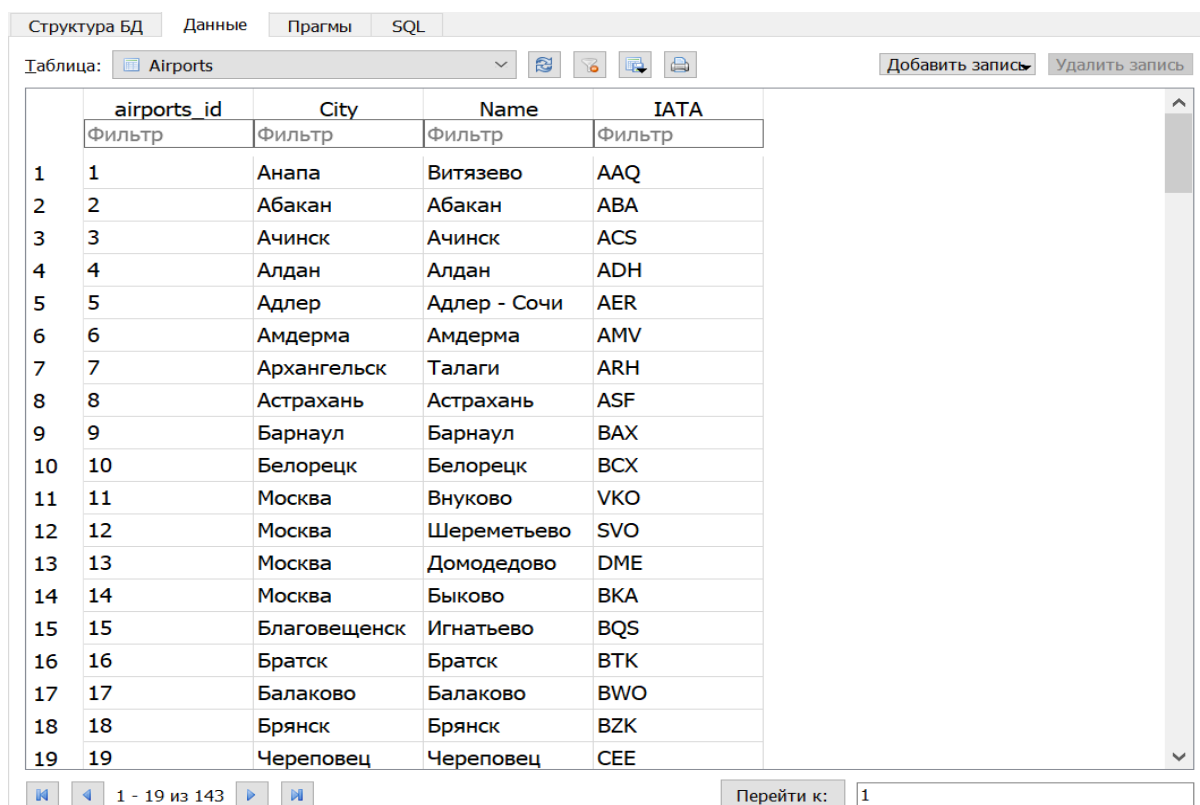
Обрезать поля:

	field1	field2	field3	field4
1	1.	Анапа	Витязево	AAQ
2	2.	Абакан	Абакан	ABA
3	3.	Ачинск	Ачинск	ACS
4	4.	Алдан	Алдан	ADH
5	5.	Адлер	Адлер - Сочи	AER
6	6.	Амдерма	Амдерма	AMV

Рисунок 13 – Импорт таблицы из Excel в DB Browser

Таким образом, был описан один из вариантов входной информации, теперь перейдем, непосредственно, к самим таблицам проектируемой БД, предназначение которых состоит в структурировании и систематизации входящей информации. Данное описание отображено на рисунках 16 – 25.

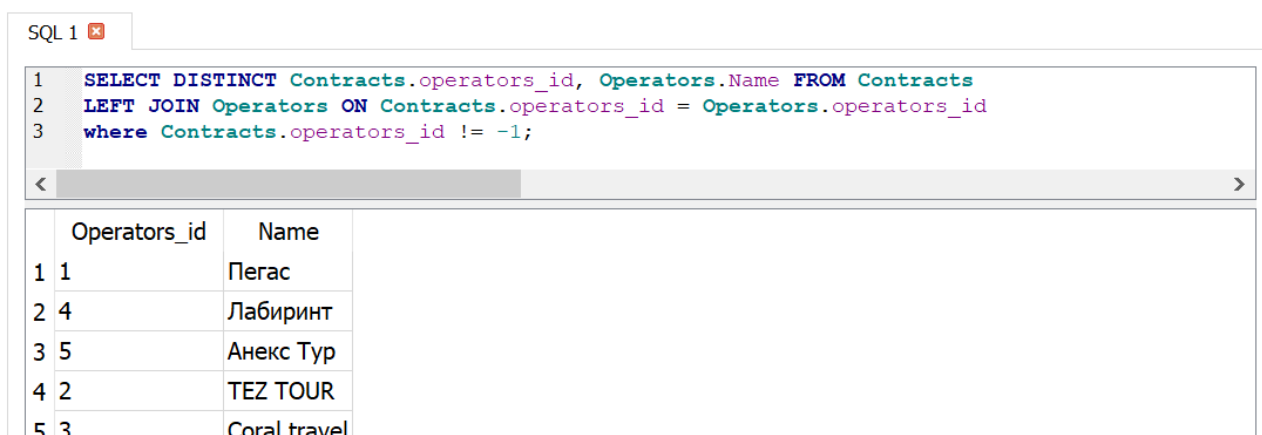
Результат импорта представлен на рисунке 14.



	airports_id	City	Name	IATA
	Фильтр	Фильтр	Фильтр	Фильтр
1	1	Анапа	Витязево	AAQ
2	2	Абакан	Абакан	ABA
3	3	Ачинск	Ачинск	ACS
4	4	Алдан	Алдан	ADH
5	5	Адлер	Адлер - Сочи	AER
6	6	Амдерма	Амдерма	AMV
7	7	Архангельск	Талаги	ARH
8	8	Астрахань	Астрахань	ASF
9	9	Барнаул	Барнаул	BAX
10	10	Белорецк	Белорецк	BCX
11	11	Москва	Внуково	VKO
12	12	Москва	Шереметьево	SVO
13	13	Москва	Домодедово	DME
14	14	Москва	Быково	BKA
15	15	Благовещенск	Игнатьево	BQS
16	16	Братск	Братск	BTK
17	17	Балаково	Балаково	BWO
18	18	Брянск	Брянск	BZK
19	19	Череповец	Череповец	CEE

Рисунок 14 – Импортируемая таблица в DB Browser

На рисунке 15 представлен запрос для возвращения, непосредственно, содержимого столбцов, а не их идентификаторов.



```
SQL 1 ✕
1 SELECT DISTINCT Contracts.operators_id, Operators.Name FROM Contracts
2 LEFT JOIN Operators ON Contracts.operators_id = Operators.operators_id
3 where Contracts.operators_id != -1;
```

	Operators_id	Name
1	1	Пегас
2	4	Лабиринт
3	5	Анекс Тур
4	2	TEZ TOUR
5	3	Coral travel

Рисунок 15 – Пример запроса в БД



Так как в таблице указаны именно идентификаторы, для того, чтобы запрос к БД возвращал не просто числовой id интересующего поля, а именно его текстовую составляющую, был составлен запрос, представленный на рисунке 15.

На рисунке 16 отображена таблица «Аэропорты», содержащая список аэропортов, осуществляющих вылет.

Имя	Тип	Схема
<b>Airports</b>		CREATE TABLE "Airports" ( "airports_id" integer primary key autoincrement not null , "city" varchar not null , "name" varchar not null , "IATA" integer not null )
airports_id	integer	"airports_id" integer NOT NULL PRIMARY KEY AUTOINCREMENT
city	varchar	"city" varchar NOT NULL
name	varchar	"name" varchar NOT NULL
IATA	integer	"IATA" integer NOT NULL

Рисунок 16 – Описание таблицы «Аэропорты»

На рисунке 17 представлена таблица «Договоры», которая включает в себя идентификационные номера контрагентов – клиенты и операторы, по которым в дальнейшем можно будет производить фильтрацию в реестре договоров. В его состав также входит параметр даты заключения договора и статус его готовности, по которому также можно производить отбор.

Имя	Тип	Схема
<b>Contracts</b>		CREATE TABLE "Contracts" ( "Contracts_id" INTEGER, "Clients_id" INTEGER, "Operators_id" INTEGER, "Date" TEXT, "Status" TEXT, PRIMARY KEY("Contracts_id") )
Contracts_id	INTEGER	"Contracts_id" INTEGER
Clients_id	INTEGER	"Clients_id" INTEGER
Operators_id	INTEGER	"Operators_id" INTEGER
Date	TEXT	"Date" TEXT
Status	TEXT	"Status" TEXT

Рисунок 17 – Описание таблицы «Договоры»

На рисунке 18 отражена таблица «Страны», которая включает в себя список стран, доступных для осуществления туристических путешествий.

Имя	Тип	Схема
<b>Country</b>		CREATE TABLE "Country" ( "country_id" INTEGER, "Name" INTEGER, PRIMARY KEY("country_id") )
country_id	INTEGER	"country_id" INTEGER
Name	INTEGER	"Name" INTEGER

Рисунок 18 – Описание таблицы «Страны»

На рисунке 19 представлена таблица «Сотрудники», содержащая список сотрудников, где атрибуты – ФИО, должность, и внешний ключ с ссылкой на аккаунт для входа в систему и разграничения прав доступа [6].

Имя	Тип	Схема
<b>Employee</b>		CREATE TABLE "Employee" ( "employee_id" INTEGER, "Lastname" TEXT, "Name" TEXT, "Middlename" TEXT, "Position" TEXT, "log_id" INTEGER, PRIMARY KEY("employee_id") )
employee_id	INTEGER	"employee_id" INTEGER
Lastname	TEXT	"Lastname" TEXT
Name	TEXT	"Name" TEXT
Middlename	TEXT	"Middlename" TEXT
Position	TEXT	"Position" TEXT
log_id	INTEGER	"log_id" INTEGER

Рисунок 19 – Описание таблицы «Сотрудники»

На рисунке 20 отражена таблица «Логин (аккаунты)», которая содержит список аккаунтов, включающий в себя логин, пароль для входа в систему и разделения прав доступа между сотрудниками.

Имя	Тип	Схема
<b>Log_in</b>		CREATE TABLE "Log_in" ( "log_id" INTEGER, "Login" TEXT, "Password" TEXT, PRIMARY KEY("log_id") )
log_id	INTEGER	"log_id" INTEGER
Login	TEXT	"Login" TEXT
Password	TEXT	"Password" TEXT

Рисунок 20 – Описание таблицы «Логин (аккаунты)»

На рисунке 21 представлена таблица «Операторы», содержащая список операторов со всеми необходимыми реквизитами для заключения договоров на куплю/продажу, предоставляемых ими, туров.

Имя	Тип	Схема
<b>Operators</b>		CREATE TABLE "Operators" ( "operators_id" INTEGER, "Name" TEXT, "Adress" TEXT, "Bank" TEXT, "OGRN" INTEGER, "INN_KPP" TEXT, "Director" TEXT, PRIMARY KEY("operators_id") )
operators_id	INTEGER	"operators_id" INTEGER
Name	TEXT	"Name" TEXT
Adress	TEXT	"Adress" TEXT
Bank	TEXT	"Bank" TEXT
OGRN	INTEGER	"OGRN" INTEGER
INN_KPP	TEXT	"INN_KPP" TEXT
Director	TEXT	"Director" TEXT

Рисунок 21 – Описание таблицы «Операторы»

На рисунке 22 представлена таблица «Заказы», которая содержит список заявок, составленных на основе интереса клиента.

Имя	Тип	Схема
<b>Orders</b>		CREATE TABLE "Orders" ( "orders_id" INTEGER, "Date_order" TEXT, "employee_id" INTEGER, "clients_id" INTEGER, "Date_flight" TEXT, "Date_arrived" TEXT, "airports_id" INTEGER, "country_id" INTEGER, "operators_id" INTEGER, "count_night" INTEGER, "count_people" INTEGER, "Bill" INTEGER, "Status_bill" TEXT, PRIMARY KEY("orders_id") )
orders_id	INTEGER	"orders_id" INTEGER
Date_order	TEXT	"Date_order" TEXT
employee_id	INTEGER	"employee_id" INTEGER
clients_id	INTEGER	"clients_id" INTEGER
Date_flight	TEXT	"Date_flight" TEXT
Date_arrived	TEXT	"Date_arrived" TEXT
airports_id	INTEGER	"airports_id" INTEGER
otels_id	INTEGER	"otels_id" INTEGER
operators_id	INTEGER	"operators_id" INTEGER
count_night	INTEGER	"count_night" INTEGER
count_people	INTEGER	"count_people" INTEGER
Bill	INTEGER	"Bill" INTEGER
Status_bill	TEXT	"Status_bill" TEXT

Рисунок 22 – Описание таблицы «Заказы»

Таблица «Заказы» включает в себя: дату обработки заказа, даты вылета/прилета, количество туристов и ночей пребывания в отеле, расчетный счет и статус оплаты, а также внешние ключи, ссылающиеся на клиента, заказавшего тур, сотрудника его оформляющего, аэропорт, из которого будет отправляться турист, и, соответственно, оператора, предоставляющего услуги отеля, и сам отель.

На рисунке 23 представлена таблица «Отели», включающая в себя список отелей, доступных для бронирования. В таблицу входит: название отеля, количество звезд, тип питания, стоимость проживания за ночь, а также внешние ключи, ссылающиеся на курорт страны, в котором он расположен.

Имя	Тип	Схема
<b>Otels</b>		CREATE TABLE "Otels" ( "otels_id" INTEGER, "Name" TEXT, "country_id" INTEGER, "resorts_id" INTEGER, "Stars" INTEGER, "Bill" TEXT, "Eating" TEXT, PRIMARY KEY("otels_id") )
otels_id	INTEGER	"otels_id" INTEGER
Name	TEXT	"Name" TEXT
country_id	INTEGER	"country_id" INTEGER
resorts_id	INTEGER	"resorts_id" INTEGER
Stars	INTEGER	"Stars" INTEGER
Bill	TEXT	"Bill" TEXT
Eating	TEXT	"Eating" TEXT

Рисунок 23 – Описание таблицы «Отели»

На рисунке 24 представлена таблица «Курорты».

Имя	Тип	Схема
<b>Resorts</b>		CREATE TABLE "Resorts" ( "resorts_id" INTEGER, "Name" TEXT, "country_id" INTEGER, PRIMARY KEY("resorts_id") )
resorts_id	INTEGER	"resorts_id" INTEGER
Name	TEXT	"Name" TEXT
country_id	INTEGER	"country_id" INTEGER

Таблица 24 – Описание таблицы «Курорты»

Таблица «Курорты» включает в себя список курортов, доступных к

приобретению туров. Помимо названия курорта содержится также и внешний ключ, который отображает то, в какой стране расположен курорт.

На рисунке 25 представлена таблица «Клиенты», которая содержит список клиентов, когда-либо заказывавших туры текущей турфирмы или недавно подавших заявление. Таблица содержит следующие атрибуты: фамилия, имя, отчество, телефон, адрес, электронная почта, паспортные данные.

Имя	Тип	Схема
<b>clients</b>		CREATE TABLE "clients" ( "clients_id" INTEGER, "Lastname" TEXT, "Name" TEXT, "MiddleName" TEXT, "Phone" TEXT, "Adress" TEXT, "Email" TEXT, "Birthday" TEXT, "Zagran" TEXT, "Zagran_date" TEXT, "Ru_pass" TEXT, PRIMARY KEY("clients_id") )
clients_id	INTEGER	"clients_id" INTEGER
Lastname	TEXT	"Lastname" TEXT
Name	TEXT	"Name" TEXT
MiddleName	TEXT	"MiddleName" TEXT
Phone	TEXT	"Phone" TEXT
Adress	TEXT	"Adress" TEXT
Email	TEXT	"Email" TEXT
Birthday	TEXT	"Birthday" TEXT
Zagran	TEXT	"Zagran" TEXT
Zagran_date	TEXT	"Zagran_date" TEXT
Ru_pass	TEXT	"Ru_pass" TEXT

Рисунок 25 – Описание таблицы «Клиенты»

Следующим этапом разработки логической модели системы, будет описание выходной информации.

### 2.1.3 Характеристика выходной информации

Проектируемая АИС управления договорами включает в себя следующие типы выходной информации:

- путевка;
- договор на оказание услуг клиенту;
- договор на оказание услуг оператора;
- отчет продаж за день/неделю/месяц/квартал;
- статистика продаваемости туров.

Документ «Путевка» – это ничто иное как представление таблицы «Заявки», приведенное в соответствие с требуемым оформлением отчетной документации. Печатная форма путевки приведена в Приложении А на рисунке А.1.

Договор между сторонами турагента и заказчика турпродукта (туроператора и турагента) выступает «гарантом» и основанием для соглашения о взаимных обязательствах, где под подпись двух сторон обозначаются границы прав и обязанностей, их состав, продукт взаимовыгодных отношений, а также гарантии при их невыполнении.

Система автоматически заполняет все необходимые поля в договоре на основании полученных входных данных, после чего остается только вывести документ на печать. Печатная форма договоров для клиента и оператора приведены в Приложении А, рисунки А.2 – А.3 и А.5 – А.6.

Отчеты продаж за выбранный промежуток времени, а также статистика продаваемых туров представлены в системе в виде графической интерпретации, то есть в виде графиков и диаграмм. Пример такого отчета представлен в главе 3, рисунок 33. Также эти отчеты можно экспортировать в Excel для дальнейшей печати. Примеры этих отчетов представлены в приложении А, рисунки А.4 и А.7.

## **2.2 Физическое моделирование АИС управления договорами на ООО «Туризм Европа»**

### **2.2.1 Выбор архитектуры АИС управления договорами на ООО «Туризм Европа»**

Для выбора наиболее подходящей архитектуры стоит рассмотреть две распространенные концепции архитектур: файловая и клиент-серверная.

**Файловый** вариант работы с базами данных предназначен для обеспечения работы небольшого круга пользователей по локальной сети. Данный подход значительно минимизирует накладные расходы, время

отклика и делает программу более простой за счет того, что не нужно дополнительно разворачивать сервер, так как все данные содержатся в одном файле, именуемом файловой базой данных [7].

Немаловажным фактором является то, что во время исполнения транзакций, выбранная ранее SQLite, хранит дополнительную информацию в запасном файле, именуемом журналом отката. Что означает, в случае отключения компьютера или отказа приложения во время выполнения транзакции, данные файлы будут содержать всю необходимую информацию для восстановления [18]. А также существенным плюсом является низкая стоимость такого подхода, его простота в ходе реализации и эксплуатации.

**Клиент-серверный** вариант работы с базами данных предназначен для более масштабных организаций, рассчитанных на одновременное взаимодействие большого количества пользователей. Он реализован на основе трехуровневой «клиент-серверной» архитектуры и является наиболее накладным и дорогим вариантом работы. Схематика данного способа представлена на рисунке 26.

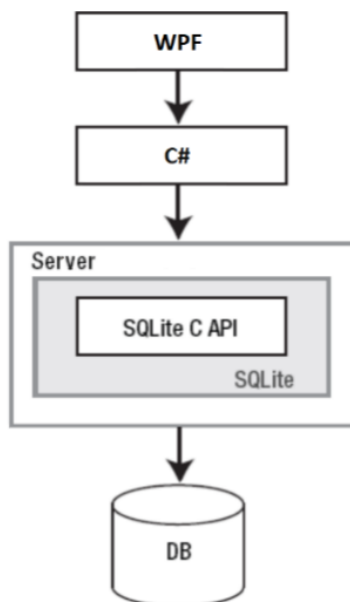


Рисунок 26 – Клиент-серверная архитектура

Из вышесказанного следует, что, в связи с небольшим штатом сотрудников рассматриваемой турфирмы, более оптимальным решением является использование файл-серверной архитектуры, но в случае расширения предприятия и увеличения штата все же наиболее целесообразно использовать клиент-серверную, поэтому файл базы данных должен находиться на отдельном сервере.

Для реализации данного способа разрабатываемая АИС должна иметь настройку, позволяющую вводить url-адрес сервера с хранящейся базой данных. Данная настройка представлена на рисунке 30.

В конечном счете была выбрана клиент-серверная архитектура. Следующим этапом предстоит описать функциональную составляющую проекта.

### 2.2.2 Функциональная схема проекта

В процессе проектирования АИС управления договорами были выделены два вида функций: основные и служебные. На основе данного разделения было построено дерево функций, изображенное на рисунке 27.

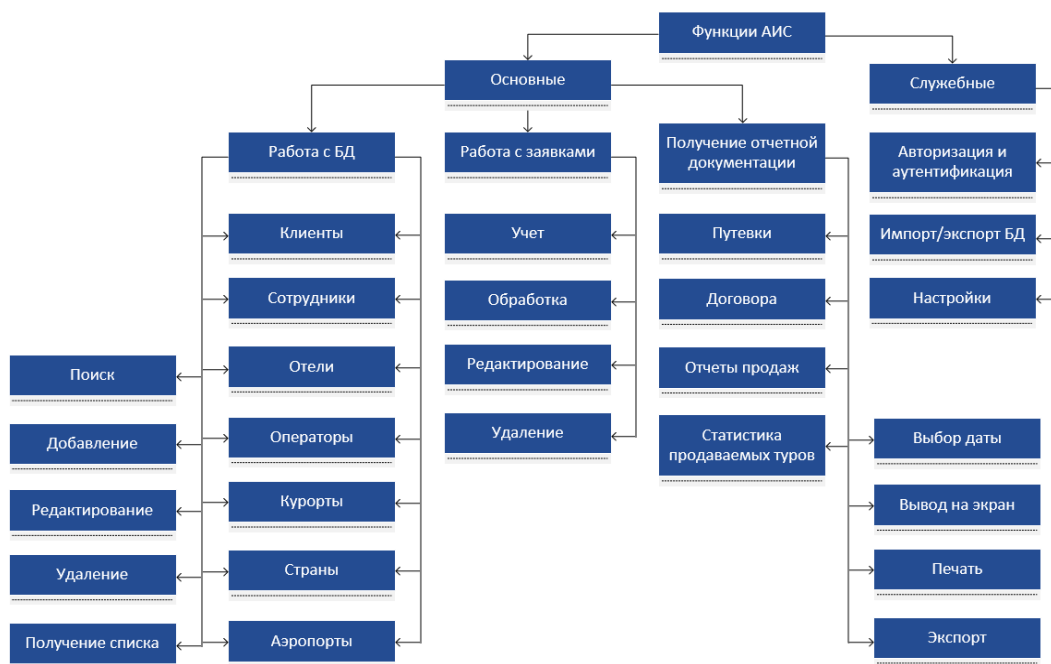


Рисунок 27 – Дерево функций АИС управления договорами



Служебные отвечают за проверку входных данных, а также за настройку обеспечения связи с БД. Основные же делятся еще на подвиды: работа с таблицами баз данных, работа с заявками и получение отчетной документации.

Полученное дерево представляет собой иерархию описанных ранее функций.

Для более детального представления также была разработана структура сценария диалога, представленная на рисунке 28.

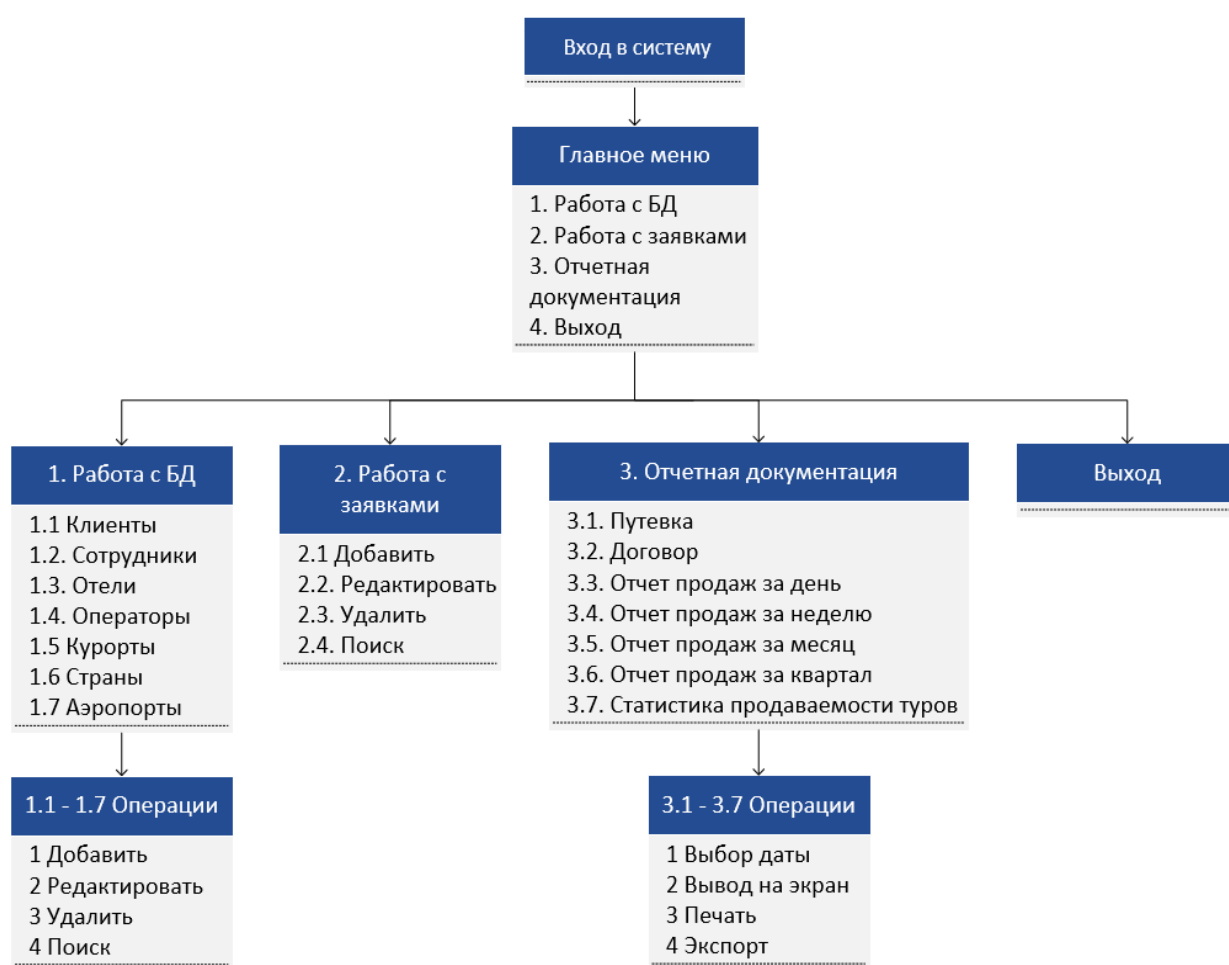


Рисунок 28 – Структура диалога системы

Таким образом, были определены уровни, на которых пользователь принимает решение относительно следующего действия.

### 2.2.3 Структурная схема проекта и описание программных модулей

По результатам выстроенной структуры функциональной схемы проекта, а также с помощью среды программирования Visual Studio 2017, было построено дерево модулей будущей программы, составляющих взаимосвязанный иерархичный макет, включающий в себя модули таких классов как:

- отвечающие за графическую составляющую АИС;
- ответственные за работу служебных функций;
- управляющие, а именно отвечающие за загрузку меню и взаимосвязь при передаче управления от одного модуля к другому;
- модули, связанные обработкой данных.

На рисунке 29 отражен непосредственный структурный макет проекта, являющийся наглядным представлением иерархии модулей программы будущей АИС управления договорами [26, 29].

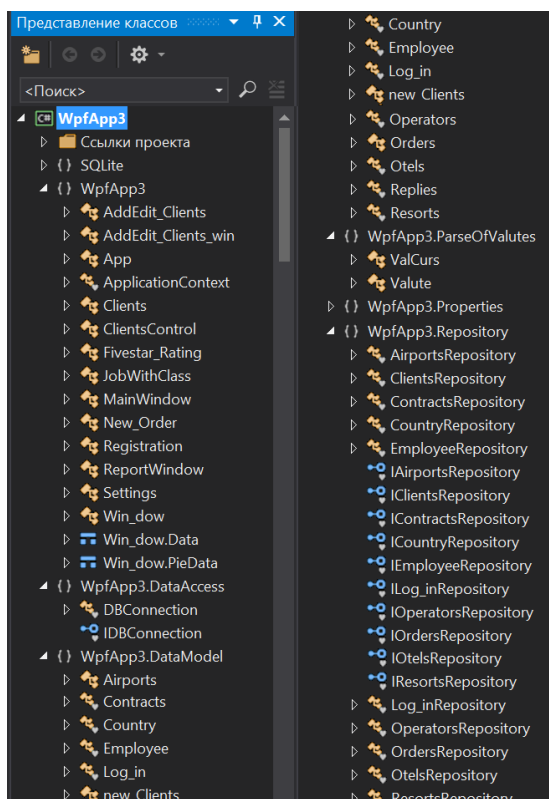


Рисунок 29 – Структурная схема основного пакета

В разделе WPFApp3 содержатся графические формы приложения, а также настройки для сохранения пути к базе данных и пользовательские элементы интерфейса [1].

Раздел DataAccess содержит функционал для подключения к базе данных.

Раздел DataModel содержит модель данных – классы, связанные с таблицами баз данных.

Раздел ParseOfValues отвечает за парсинг актуальной валюты с официального сайта ЦБ.

Раздел Repository содержит репозитории и их интерфейсы, включающие в себя функции, выполняющие CRUD-операции (добавление, удаление, редактирование), а также необходимые запросы к базе данных [18].

Далее необходимо описать модули, в указанных выше разделах, и выполняемые ими функции.

Описание этих модулей представлено в таблице 2.

Таблица 2 – Описание программных модулей проекта

<b>Программный модуль</b>	<b>Описание программного модуля</b>
Модуль данных (DataAccess)	Модуль данных содержит класс DBConnection и интерфейс IDBConnection для удобства разделения функций для обеспечения связи с базой данных, хранящейся в таблицах SQLite. Связь с базой данных осуществлена с помощью библиотеки SQLite и вызова функции SQLiteAsyncConnection.
Модуль главной формы (Win_dow)	Модуль главной формы представляет собой навигационное меню для работы с входными/выходными данными, подлежащими редактированию и хранящимися в таблицах базы данных SQLite. Визуализация данных из БД происходит за счет их привязки с помощью Binding к графическим элементам ListView и вызова соответствующих функций из behind кода. Например, функция Update_ListView_Orders() отвечает за заполнение данными визуализированных таблиц ListView, а также элементов выпадающего списка ComboBox.
Модуль данных (DataModel)	Модуль данных содержит классы, которые являются аналогами таблиц базы данных SQLite, связь с которой обеспечивается также за счет библиотеки SQLite и табличного атрибута Table.

## Продолжение таблицы 2

Модуль данных (ParseOfValues)	Модуль данных содержит класс Valute для хранения данных, полученных с сайта ЦБ и класс ValCurs, содержащий поля для сериализации, которые будут заполнены при загрузке главной формы после успешной авторизации пользователя.
Модуль данных (Repository)	Модуль данных содержит классы и их интерфейсы, хранящие функционал, предназначенный, для работы с классами, олицетворяющими таблицы БД, описанные выше. Каждый класс содержит функцию обеспечения связи с БД, например OrdersRepository() и функции выполнения CRUD-операций – insert, update, delete и конечно же запросы на выборку данных из таблиц, либо всех имеющихся SelectAll(), либо выборочно из разных таблиц Select(query).
Класс (Settings)	Содержит обращение к сохраненному пути к БД при выборе его пользователем при первом запуске АИС управления договорами.
Структуры (Data и PieData)	Содержат поля для определения текущей даты и сравнения её с данными, хранящимися в таблице «Заказы» для корректного отображения отчетных графиков продаж за выбранный промежуток времени.
Модуль формы (ReportWindow)	Модуль формы отвечает за экспорт в Excel для дальнейшей печати сформированной отчетности.

Ознакомиться с листингом программного кода описанных в таблице программных модулей можно в Приложении В.

### 2.3 Контрольный пример реализации проекта и его описание

При первом запуске программы пользователя первым делом попросят указать путь к базе данных с помощью соответствующих диалоговых окон, а также задать url-адрес сервера в настройках в окне авторизации. Данные диалоговые окна представлены на рисунке 30. В случае, если по указанному пути файла БД не будет, программа создаст его автоматически за счет классов, расположенных в разделе DataModel программного кода, описанного в пункте 2.2.3.

В случае, если файл БД обнаружен – программа создаст файл с настройками Settings.xml, чтобы при следующих запусках программы пользователю не приходилось каждый раз указывать путь к БД.

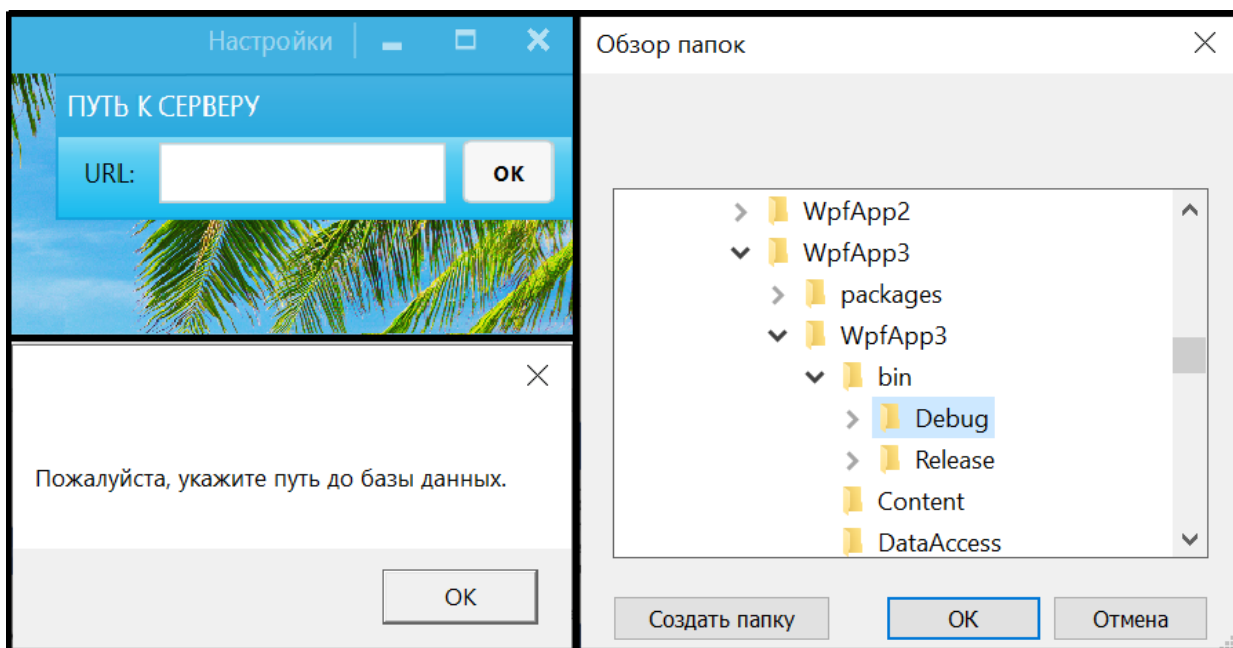


Рисунок 30 – Диалоговое окно выбора пути к базе данных

Далее появляется окно входа в систему на рисунке 31. В случае ввода неверных данных внизу появится соответствующее предупреждение, что тоже видно исходя из указанного рисунка.

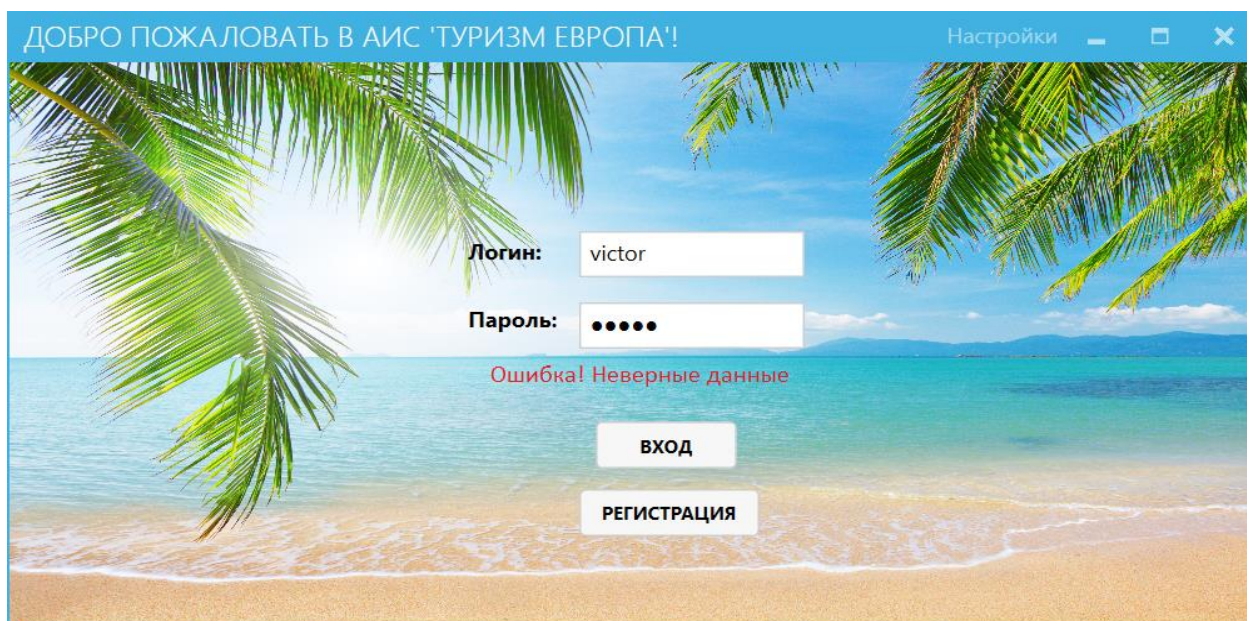


Рисунок 31 – Вход в систему



Окно регистрации представлено на рисунке 32.

РЕГИСТРАЦИЯ

Фамилия: Викторов

Имя: Виктор

Отчество: Викторович

Должность: Глав. менеджер

Логин: victor

Пароль: ●●●●

РЕГИСТРАЦИЯ

Рисунок 32 – Окно регистрация

При регистрации сотрудник должен указать свою должность для дальнейшего разграничения прав доступа к вкладке «Управление/Сотрудники».

После успешного входа в систему откроется главное меню с 6 главными вкладками: Бронь, Заказы, Клиенты, Отели, Статистика и Управление, что отражено на рисунке 33.

ГЛАВНОЕ МЕНЮ

Бронь Заказы Клиенты Отели Статистика Управление \$: 72.73 €: 79.12

Поиск... Выбор даты 14 Менеджер: Все Статус: поиск

ДАТА	МЕНЕДЖЕР	КЛИЕНТ	ОТКУДА	КУДА	ВЫЛЕТ	ВОЗВРАТ	ОПЕРАТОР	СТОИМОСТЬ	СТАТУС	ПРИМЕЧАНИЕ
01.03.2020	Иванов	Бузова	Пулково	Иордания	10.03.20	24.03.20	Перас	60000	Оплачено	
02.03.2020	Иванов	Лакшери	Курумоч	Абхазия	05.03.20	10.03.20	Перас	6000	Оплачено	
02.03.2020	Новиков	Казимиров	Домодедово	Абхазия	09.03.20	12.03.20	Перас	14500	Оплачено	
06.03.2020	Иванов	Мишунин	Домодедово	Сингапур	15.03.20	20.03.20	Лабиринт	22000	Оплачено	
12.03.2020	Новиков	Измайлов	Саратов	Абхазия	18.03.20	26.03.20	Анекс Тур	43000	Оплачено	
20.03.2020	Новиков	Лабода	Домодедово	Боливия	19.04.20	25.04.20	TEZ TOUR	81000	Не оплачен	до 21.04
01.04.2020	Новиков	Святошин	Барнаул	Каймановы ост	21.04.20	23.04.20	Coral travel	9700	Отменен	Болезнь

ДОБАВИТЬ ИЗМЕНИТЬ УДАЛИТЬ

Рисунок 33 – Вкладка «Заказы»

На вкладке «Заказы» располагается таблица с историей заказов, фильтры для её сортировки: по дате обращения, по оформившему заявку сотруднику и статусу; а также поиск по фамилии клиента и кнопки для редактирования, при нажатии одной из которых произойдет переход на вкладку «Бронь», которая представлена на рисунке 36.

Пример применения фильтрации приведен на рисунке 34.

В правом верхнем углу расположены поля, отвечающие за отображение актуального курса доллара и евро вне зависимости от открытой вкладки.

За функционирование данных полей отвечают классы ValCurs.cs и Valute.cs, а также метод MetroWindow\_Loaded(), который срабатывает при загрузке формы, посылая запрос на получение данных с официального сайта Центрального банка РФ.

Таким образом, программа отображает актуальный курс на момент запуска.

Для просмотра и редактирования клиентов переходим во вкладку «Клиенты», приведённой на рисунке 35.

ДАТА	МЕНЕДЖЕР	КЛИЕНТ	ОТКУДА	КУДА	ВЫЛЕТ	ВОЗВРАТ	ОПЕРАТОР	СТОИМОСТЬ	СТАТУС	ПРИМЕЧАНИЕ
01.03.2020	Иванов	Бузова	Пулково	Иордания	10.03.20	24.03.20	Перас	60000	Оплачено	
02.03.2020	Иванов	Лакшери	Курумоч	Абхазия	05.03.20	10.03.20	Перас	6000	Оплачено	
06.03.2020	Иванов	Мишунин	Домодедово	Сингапур	15.03.20	20.03.20	Лабиринт	22000	Оплачено	

Рисунок 34 – Применение фильтрации на вкладке «Заказы»

ГЛАВНОЕ МЕНЮ

Бронь Заказы Клиенты Отели Статистика Управление \$: 72.73 €: 79.12

ОБЩИЕ ДАННЫЕ		КОНТАКТНЫЕ ДАННЫЕ		ПАСПОРТНЫЕ ДАННЫЕ	
Фамилия: *	Святошин	Дата рождения:	25.11.1989	Загран: *	425 34540921
Имя: *	Дмитрий	Телефон:	89346371671	Действителен: *	19.03.2025
Отчество:	Юрьевич	Почта:	dimon@yandex.ru	Паспорт РФ: *	4610 821724

ФИО	ТЕЛЕФОН	АДРЕС	EMAIL	BIRTHDAY	ЗАГРАН.ПАС	СРОК	ПАСПОРТ РФ
Лакшери Василиса Антоновна	89379901355	Московский пр.	Luhary@mail.ru	03.02.1990	585 31545811	28.05.2021	4601 729721
Казимиров Антон Петрович	89347242364	Невский пр.	anton1985@mail.r	15.03.1985	315 81525810	14.01.2022	4611 809723
Мишунин Олег Александрович	89346327621	Васильевский пр	Misha@gmail.com	01.07.1982	212 11345891	02.05.2024	4502 449703
Измайлов Алексей Михайлович	89213672613	Приморский пр.	Alejandro@yande	12.05.1988	134 81585119	17.06.2021	4600 824128
Бузова Ольга Дмитриевна	89234671687	Волжские зори	olis91@mail.ru	05.07.1991	985 21575851	09.11.2023	4601 434725
Святошин Дмитрий Юрьевич	89346371671	Льва Яшина 19	dimon@yandex.ru	25.11.1989	425 34540921	19.03.2025	4610 821724
Лабода Светлана Ильина	89124653661	Псковский пр.	sun@gmail.com	14.02.1978	585 11555800	14.06.2022	4602 729423

ДОБАВИТЬ    ИЗМЕНИТЬ    УДАЛИТЬ    Поиск...    ДОБАВИТЬ БРОНЬ

Рисунок 35 – Вкладка «Клиенты»

Вкладка «Клиенты», помимо описанного выше функционала, также включает в себя дополнительные поля ввода и выбора дат. К тому же, при выборе клиента его данные автоматически отобразятся в соответствующих полях, что позволяет редактировать их, не переключаясь между вкладками.

Выбрав нужного клиента переходим к добавлению брони, представленной на рисунке 36.

При добавлении брони из вкладки «Клиент» данные выбранного клиента также заполняются автоматически. К тому же, при выборе страны отдыха данные выпадающего списка «Туроператор» отображает только операторов, предоставляющих свои услуги в выбранной стране, тоже самое и с выпадающими списками «Курорт» и «Отель».

Пример можно увидеть на рисунке 37.

Также при выборе отеля автоматически заполняется количество звезд и тип питания.



ГЛАВНОЕ МЕНЮ

Бронь Заказы Клиенты Отели Статистика Управление \$: 72.73 €: 79.12

ОБЩИЕ ДАННЫЕ	КОНТАКТНЫЕ ДАННЫЕ	ПАСПОРТНЫЕ ДАННЫЕ
Фамилия: Святошин	Дата рождения: 25.11.1989	Загран: 425 34540921
Имя: Дмитрий	Телефон: 89346371671	Действителен: 19.03.2025
Отчество: Юрьевич	Почта: dimon@yandex.ru	Паспорт РФ: 4610 821724

ДАННЫЕ О ВЫЛЕТЕ	ДАННЫЕ О КУРОРТЕ	ДАННЫЕ ОБ ОПЛАТЕ
Вылет: 02.04.2020	Туроператор: Coral travel	Статус: В обработке
Откуда: Казань	Курорт: Гагры	Примечание: оплата до 30.03
Куда: Абхазия	Отель: Аквамарин	Рейтинг отеля: ★★★★★
Кол-во ночей: 6	Питание: ВВ - Только завтрак	Стоимость: 73 200
Туристы: 2		<input type="radio"/> RUB <input type="radio"/> USD <input type="radio"/> EUR

ДОГОВОР С ТУРИСТОМ   ДОГОВОР С ОПЕРАТОРОМ   ПУТЕВКА   СОХРАНИТЬ

Рисунок 36 – Вкладка «Бронь»

При добавлении брони из вкладки «Клиент» данные выбранного клиента также заполняются автоматически. К тому же, при выборе страны отдыха данные выпадающего списка «Туроператор» отображает только операторов, предоставляющих свои услуги в выбранной стране, тоже самое и с выпадающими списками «Курорт» и «Отель». Пример можно увидеть на рисунке 37. Также при выборе отеля автоматически заполняется количество звезд и тип питания.

ДАННЫЕ О ВЫЛЕТЕ	ДАННЫЕ О КУРОРТЕ
Вылет: Выбор даты	Туроператор: Coral travel
Откуда: Казань	Курорт: [выпадающий список]
Куда: Абхазия	Отель: Гагры, Гудаута, Мюссера, Новый Афон, Пицунда, Сухум, Цандрипш
Кол-во ночей: 1	Рейтинг отеля: [звезды]
Туристы: 1	Питание: [тип питания]

Рисунок 37 – Пример связанных выпадающих списков

Элемент «Выбор даты» включает в себя встроенный календарь, представленный на рисунке 38.

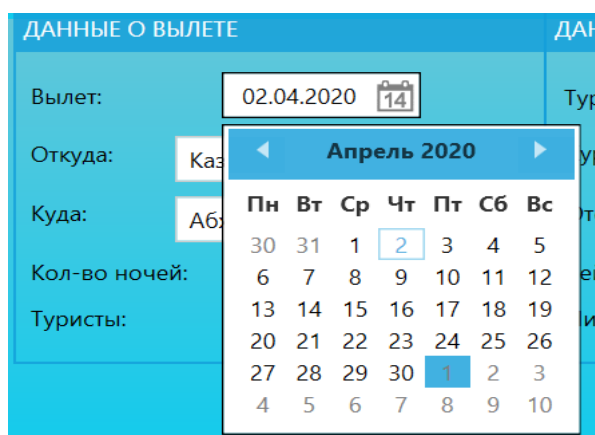


Рисунок 38 – Элемент «Календарь»

Также существует контроль ввода данных, поэтому, если заполнены не все поля, или при создании договора на основании данных клиента и его путевки, сам клиент из списка не выбран, то высветится соответствующее окно, пример которого представлен на рисунке 39.

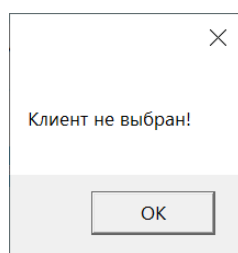


Рисунок 39 – Пример окна контроля ввода данных

Во вкладке «Отели» можно производить расчет стоимости проживания в отеле в зависимости от количества ночей и числа проживающих, а также учитывая текущий курс доллара/евро, конвертировать конечную стоимость в нужной валюте. Вкладка «Отели» отображена на рисунке 40.

ГЛАВНОЕ МЕНЮ

Бронь Заказы Клиенты **Отели** Статистика Управление \$: 72.73 €: 79.12

РАСПОЛОЖЕНИЕ	ДАННЫЕ ОБ ОТЕЛЕ	ДАННЫЕ ДЛЯ РАСЧЕТА ОПЛАТЫ
Страна: ОАЭ	Отель: Armani Hotel Dubai	Кол-во ночей: 5
Курорт: Дубаи	Рейтинг отеля: ★★★★★	Туристы: 2
	Питание: FB - Полный пансион	Стоимость: 3 122,95 \$ USD

НАЗВАНИЕ	СТРАНА	КУРОРТ	КОЛ-ВО ЗВЕЗД	ПИТАНИЕ	ЦЕНА ЗА НОЧЬ
The Oberoi Udaivilas	Индия	Удайпур	5	НВ - Завтрак и ужин	27 144
Baros Maldives	Мальдивы	Мале	5	UAI - Ультра всё включен	41 290
<b>Armani Hotel Dubai</b>	<b>ОАЭ</b>	<b>Дубаи</b>	<b>5</b>	<b>FB - Полный пансион</b>	<b>22 712</b>
Mriya Resort & Spa	Россия	Крым	5	НВ - Завтрак и ужин	14 600
Hotel The Peninsula Paris	Франция	Париж	5	AI - Всё включено	54 057
Finch Bay Galapagos Hotel	Эквадор	Пуэрто-Айора	5	AI - Всё включено	26 403
The Venetian Macao Resort Hotel	Китай	Макао	5	FB - Полный пансион	13 837

ДОБАВИТЬ    ИЗМЕНИТЬ    УДАЛИТЬ

Рисунок 40 – Вкладка «Отели» и расчет стоимости

Содержимое вкладки «Управление» имеет разный уровень доступа для определенного круга пользователей, например, юрист может работать во вкладке «Договора», но ему не доступны данные сотрудников и аэропортов, а главный менеджер может как отслеживать статусы готовности договоров, так и раздавать права доступа во вкладке «Сотрудники», директор же имеет доступ и к тому и к другому.

Содержимое вкладки «Управление» на примере вкладки «Сотрудники» с уровнем доступа главного менеджера представлено на рисунке 41.

Содержимое вкладки «Договора» с уровнем доступа юриста представлена на рисунке 42.

Вкладка «Договора» представляет собой реестр договоров, по которому можно производить параметризованный поиск с применением фильтрации по контрагентам, статусу, а также по дате создания договора, и производить все интересующие пользователя операции по управлению этим реестром.

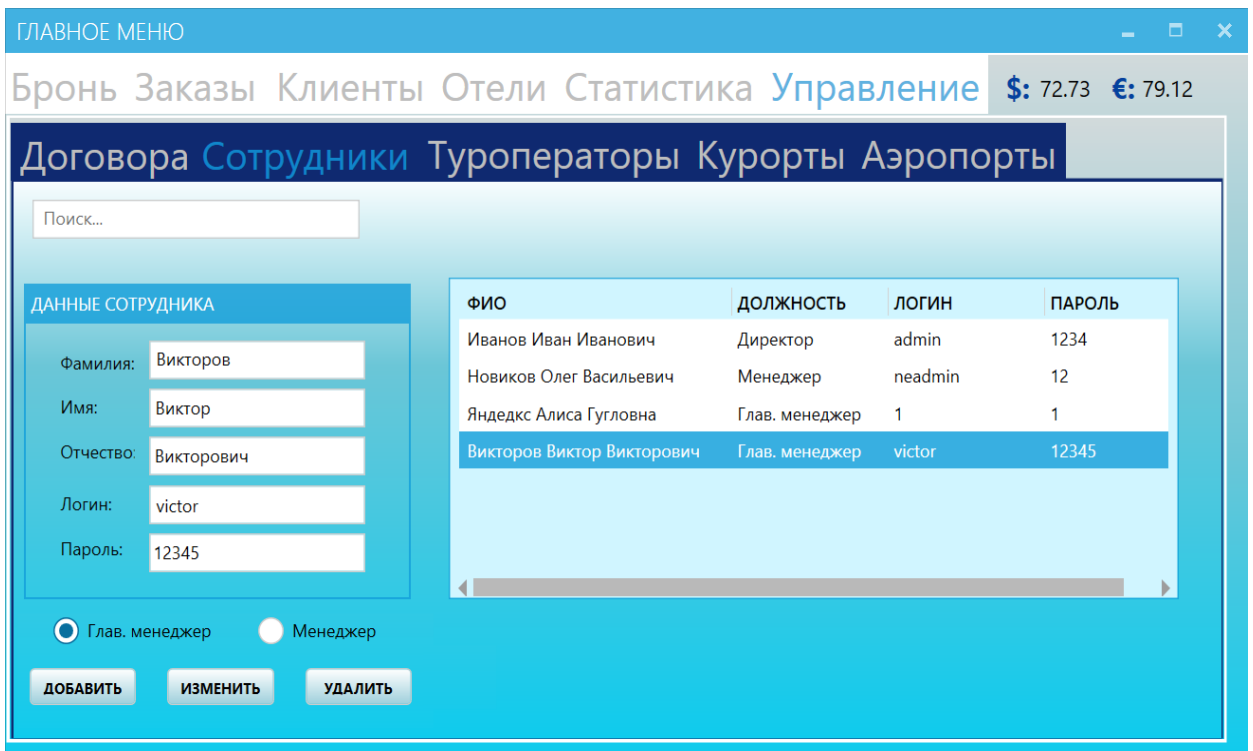


Рисунок 41 – Вкладка «Управление» с уровнем доступа главного менеджера

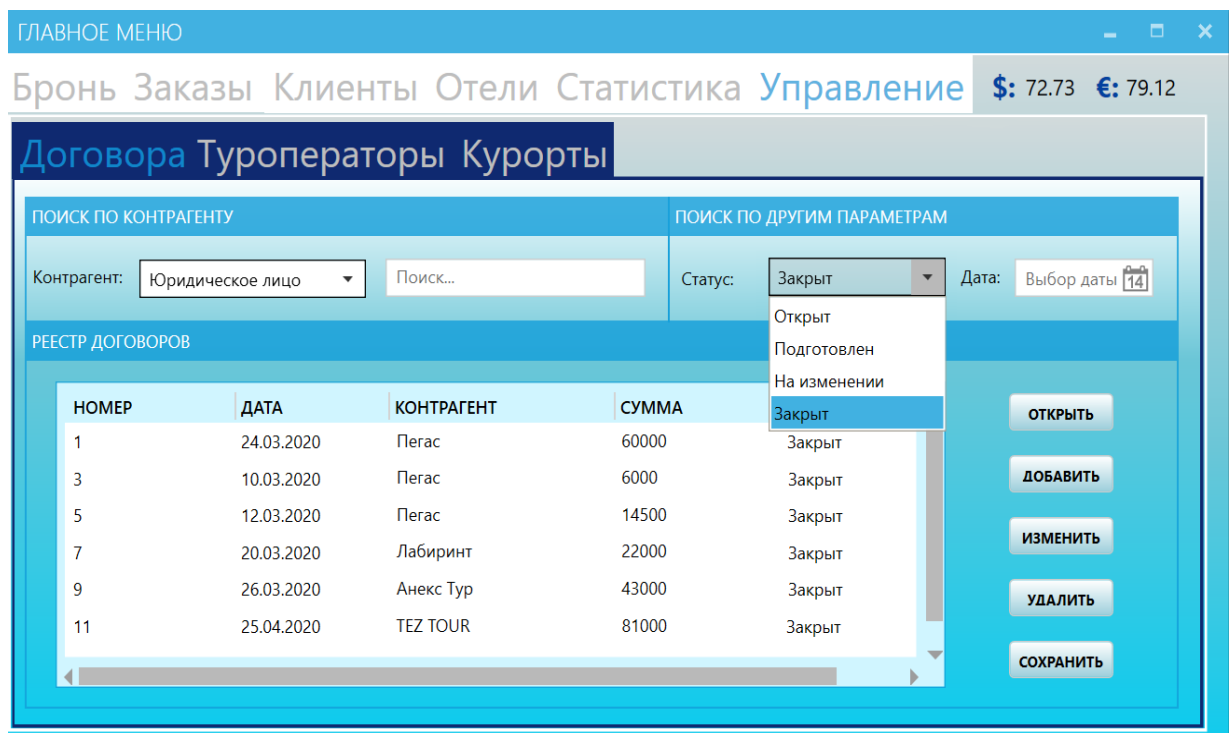


Рисунок 42 – Вкладка «Договора» с уровнем доступа юриста

Остальное содержимое вкладки «Управление» представлено в приложении А.

Вид главного меню с ограниченным доступом на примере вкладки «Статистика» со сформированным отчетом продаж за месяц представлен на рисунке 43.

Статистика продаваемости туров, а также печатные формы отчетности приведены в приложении Б.



Рисунок 43 – Отчет о работе турфирмы за месяц

Таким образом, были представлены и описаны примеры реализации АИС управления договорами.

## **Выводы по главе 2**

В ходе логического проектирования были построены и описаны диаграммы вариантов использования, классов и последовательности проектируемой системы. Тем самым, была отражена специфика деятельности туристической фирмы, для которой разрабатывается АИС управления договорами.

В ходе физического проектирования была выбрана архитектура АИС, разработаны и описаны функциональные и структурные схемы проекта, а также продемонстрированы и описаны примеры реализации конечной АИС управления договорами.

## **Глава 3 Оценка и обоснование экономической эффективности проекта**

### **3.1 Выбор и обоснование методики расчета экономической эффективности**

Для оценки экономической эффективности автоматизации процесса управления договорами на ООО «Туризм Европа» посредством АИС необходимо определить прямой и косвенный эффект от внедрения технологического решения по сравнению с существующим вариантом обработки информации в организации.

Основными факторами оценки экономического эффекта выступают:

- показатели прямого эффекта (трудовые и стоимостные показатели, срок окупаемости технического решения);
- показатели косвенного эффекта (гибкость доработки ИС в процессе неизбежной модернизации, лёгкость управляемости непосредственными пользователями, качество и скорость обработки входящих и исходящих данных и отдельных операций);
- выгодность и необходимость внедрения АИС управления договорами;
- целесообразность капитальных затрат на разработку АИС управления договорами.

Также необходимо определить очередность этапов автоматизации процесса управления договорами на ООО «Туризм Европа».

В настоящее время ООО «Туризм Европа» для управления договорами, а также входящей и исходящей информации использует пакеты MS «Office» в силу небольшого масштаба деятельности. Данный выбор способа работы с информацией имеет ряд недостатков:

- высокий риск допущения ошибок;

- низкая скорость обработки информации;
- децентрализованное хранение данных и возможность их потери;
- низкий уровень скорости принятия решения высшим звеном управления;
- отсутствие дополнительных возможностей работы с имеющейся базой телефонии клиентов, как каналом рекламных кампаний.

Представленные выше недостатки существующей системы обработки данных на ООО «Туризм Европа» свидетельствуют о неэффективном использовании ресурсов, что ведёт за собой упущение возможной прибыли, неконкурентоспособность и слабую устойчивость организации.

По результатам сравнительного анализа основных аналогов, проектируемого автоматизированного программного решения на предмет соответствия требованиям к системе было принято решение о разработке программного обеспечения с нуля, которое будет учитывать все особенности включать в себя весь функционал, необходимый для автоматизации основных бизнес-процессов и в полной мере отвечающий требованиям к АИС управления договорами.

Основная часть стоимости разработки и внедрения АИС управления договорами состоит из трудовых затрат работы приглашённого специалиста – разработчика С# и социальных отчислений во внебюджетные фонды (30%): Пенсионный фонд (ПФР) – 22%, Фонд обязательного медицинского страхования (ФФОМС) – 5,1%, Фонд социального страхования (ФСС) – 2,9%. Непосредственное вознаграждение за разработку АИС управления договорами будет состоять из объёма трудозатрат и часовой ставки в размере 500 рублей\час. На ООО «Туризм Европа» в настоящее время в штате сотрудников числится системный администратор, отвечающий за бесперебойную работу технической стороны организации. В целях последующего обслуживания АИС системный администратор организации будет участвовать в формировании требований к АИС, технического задания,



а также пройдет обучение у приглашенного специалиста. Стоимость машинных мощностей не учитывается, так как будет входить в почасовую оплату специалиста.

В таблице 3 представлен бюджет разработки и внедрения АИС управления договорами ООО «Туризм Европа» по этапам разработки АИС.

Таблица 3 – Капитальные затраты разработки и внедрения АИС управления договорами на ООО «Туризм Европа»

№п\п	Виды затрат	Объем ч\рубль
1	Формирование требований к АИС	3 \ 1500
2	Тех. задание	4 \ 2000
3	Определение бюджета разработки и внедрения АИС	1 \ 500
4	Эскизный проект	15 \ 7500
5	Технический проект	30 \ 15 000
6	Рабочий проект	30 \ 15 000
7	Внедрение АИС	5 \ 2500
8	Оформление пояснительной документации	4 \ 2000
9	Обучение 1 специалиста в виде курса 20часов	20 \ 10 000
10	Трудозатратность разработки АИС	112 \ 56 000
11	Отчисления во внебюджетные фонды (30%)	16 800
12	Затраты на разработку и внедрение АИС	72 800

Таким образом, капитальные затраты (КП) на внедрение и разработки АИС управления договорами составит 72 800 рублей.

В настоящее время в ООО «Туризм Европа» основной объём работы заключается в ручной обработке и занесении данных в БД MS «Excel». При этом общее время оформления клиента и предоставления ему полной и удовлетворяющей его требования информации занимает 90 минут: от начала бронирования на поиск тура по желаемым критериям до заключения официального двухстороннего договора, что является «узким» местом в работе туристической организации и требует совершенствования.

Рассмотрим оперативность работы базового варианта обработки данных на ООО «Туризм Европа» и в сравнении с экономией времени при проектном варианте автоматизации в таблице 4.

Таблица 4 – Сравнительная таблица базового и проектного варианта трудоёмкости на 1 сотрудника

№п\п	Вид работ	До автоматизации, мин	Экономия времени, мин	После автоматизации, мин
1	Поиск тура по критериям	20	15	5
2	Бронирование	15	10	5
3	Занесение личных данных о клиенте в базу данных	10	10	0
4	Проведение расчётов стоимости тура	15	15	0
5	Оформление и печать договора	15	15	0
6	Подготовка отчёта и аналитики	15	5	10
7	Общее время оформления	90	70	20

Расчёт экономии от повышения производительности труда рассчитывается по формуле (1):

$$Pr = \left( \frac{\Delta T}{T_0 - \Delta T} \right) * 100\%, \quad (1)$$

где Pr – Повышение производительности труда, %;

$\Delta T$  – Экономия времени на 1 клиента в проектном варианте, минута;

$T_0$  – Время, затраченное на оформление 1 клиента в базовом варианте, минута.

$$\text{Пр} = \left(\frac{70}{90 - 70}\right) * 100\% = 350\%$$

Таким образом, экономия от прироста производительности труда составила 350% (3,5раз).

При этом, часть операций будет исключена из функционала пользователя АИС, а именно занесение личных данных о клиенте в базу данных, проведение расчёта стоимости тура, оформление и печать договора. Экономия времени после внедрения АИС составит 70 минут.

При увеличении производительности целесообразно в целях сокращения затрат уменьшить штат сотрудников. Фонд оплаты труда по управлению договорами на ООО «Туризм Европа» в месяц рассчитывается по формуле (2):

$$C = Ч * Ок * \left(1 + \frac{Вн}{100\%}\right) * \left(1 + \frac{П}{100\%}\right), \quad (2)$$

где С – Фонд оплаты труда, рубль;

Ч – численность сотрудников, количество;

Ок – Оклад сотрудника, рубль;

Вн – процент отчислений во внебюджетные фонды (30%), %;

П – процент премий (25%), %.

Фонд оплаты труда по управлению договорами на ООО «Туризм Европа» в месяц рассчитывается по формуле (3):

$$CГ = C \times м, \quad (3)$$

где м – количество месяцев в периоде, месяц.

Фонд оплаты труда специалистов по управлению договорами в настоящее время в месяц составляет:

$$C_0 = 7 * 20\,000 * \left(1 + \frac{30\%}{100\%}\right) * \left(1 + \frac{25\%}{100\%}\right) = 227\,500 \text{ (месяц)}$$

Фонд оплаты труда специалистов по управлению договорами в настоящее время в год составляет:

$$C_{0г} = 227\,500 * 12 = 2\,730\,000 \text{ (год)}$$

При увеличении производительности на 350% (3,5 раза) целесообразно сократить штат до 2 человек, но, чтобы исключить человеческий фактор (выход на больничные, отпуска и т.п.), необходимо оставить в штате 3 человек.

Внедрение АИС сократит количество операций каждого пользователя и упростит функционал, что также является положительным эффектом.

Фонд оплаты труда специалистов по управлению договорами после внедрения АИС и увеличения производительности составит в месяц:

$$C_1 = 3 * 20\,000 * \left(1 + \frac{30\%}{100\%}\right) * \left(1 + \frac{25\%}{100\%}\right) = 97\,500 \text{ (месяц)}$$

Фонд оплаты труда специалистов по управлению договорами после внедрения АИС и увеличения производительности составит в год:

$$C_{1г} = 97\,500 * 12 = 1\,170\,000 \text{ (год)}$$

Автоматизация управления договорами посредством АИС на ООО «Туризм Европа» позволит снизить трудоёмкость расчётов, повысить

оперативность управления, а именно сократить время выполнения операций, что увеличит производительность труда. Также другими показателями косвенного эффекта от автоматизации выступят повышение качества данных, что снизит человеческий фактор при обработке данных.

### **3.2 Расчет показателей экономической эффективности проекта**

Основополагающим фактором экономической эффективности разработки технического решения является прямой эффект от внедрения, заключающийся в количественных и качественных показателях трудовых и стоимостных затратах, а также сроке окупаемости капитальных затрат.

Величину и целесообразность трудовых затрат отразят следующие экономические показатели:

1. Снижение трудовых затрат в абсолютном выражении оценивается по формуле (4):

$$\Delta T = T_0 - T_1, \quad (4)$$

где  $T_0$  – ограниченное время для проведения операций по полному оформлению 1 клиента в базовом варианте, минута;

$T_1$  – ограниченное время для проведения операций по полному оформлению 1 клиента в проектном варианте, минута.

$$\Delta T = 90 - 20 = 70$$

Трудовые затраты сократятся на 70 минут или 1,1(6) часа. И в проектном варианте составит 20 минут или 1, (3) часа. Полная автоматизация части операций (занесение личных данных о клиенте в базу данных, проведение расчёта стоимости тура, оформление и печать договора) и значительное сокращение времени на каждую операцию позволит ООО

«Туризм Европа» максимально снизить риск ошибок и переделок, увеличить скорость и качество обработки информации и увеличить масштаб обслуживания большего количества потенциальных клиентов. Также это позволит облегчить работу непосредственным пользователям АИС управления договорами.

2. Оценка коэффициента снижения трудовых затрат в относительном выражении КТ (%) производится по формуле (5):

$$КТ = \left(\frac{\Delta T}{T_0}\right) * 100\%, \quad (5)$$

$$КТ = \left(\frac{70}{90}\right) * 100\% = 77, (7)\%$$

Снижение трудовых затрат в относительном выражении в соотношении трудовых затрат в базовом варианте составит 77, (7) %, что также отражает эффективные качественные изменения в показателе трудоёмкости пользователя АИС.

3. Для определения индекса снижения трудовых затрат используется следующая формула (6):

$$Y_T = \frac{T_0}{T_1}, \quad (6)$$

$$Y_T = \frac{90}{20} = 4,5$$

Индекс снижения трудовых затрат составил 4,5, что свидетельствует о значительном увеличении эффективности и производительности труда в проектном варианте АИС управления договорами в 4,5 раз.

Стоимостные показатели оценки экономической эффективности разработки и внедрения АИС для управления договорами отражают непосредственное получение выгоды от снижения затрат:

1. Снижение стоимостных затрат в абсолютном выражении в данном случае отображает изменение ФОТ, как самой значительной статьи расходов любой организации, в том числе ООО «Туризм Европа».

Изменение ФОТ после внедрения АИС будет рассчитываться по формуле (7):

$$\Delta C = C_0 - C_1, \quad (7)$$

где  $\Delta C$  – величина изменения ФОТ специалистов по управлению договорами после внедрения АИС, рубль;

$C_0$  – стоимостные затраты на обработку информации по базовому варианту (ФОТ специалистов по управлению договорами в базовом варианте), рубль;

$C_1$  – стоимостные затраты на обработку информации по проектному варианту (ФОТ специалистов по управлению договорами в проектном варианте), рубль.

$$\Delta C = 227\,500 - 97\,500 = 130\,000 \text{ (месяц),}$$

$$\Delta C = 130\,000 * 12 = 1\,560\,000 \text{ (год)}$$

Внедрение проектного варианта АИС управления договорами позволит высвободить 1 560 000 рублей в год и 130 000 рублей ежемесячно, которые в базовом варианте можно считать упущенной прибылью без учёта возможно упущенных потенциальных клиентов и прибыли от взаимовыгодного сотрудничества с ними.

2. Коэффициент снижения стоимостных затрат в относительном выражении КС (%) производится по формуле (8):

$$КС = \left( \frac{\Delta C}{C_0} \right) * 100\%, \quad (8)$$

$$КС = \left( \frac{130\,000}{227\,500} \right) * 100\% = 57,14$$

Качественный показатель снижения стоимостных затрат отражает эффективность проектного решения АИС управления договорами в соотношении с базовым вариантом ООО «Туризм Европа» на 57%.

3. Индекс снижения стоимостных затрат рассчитывается по формуле (9):

$$УС = \frac{C_0}{C_1}, \quad (9)$$

$$УС = \frac{227\,500}{97\,500} = 2,3$$

Индекс снижения стоимостных затрат в проектном варианте АИС управления договорами на ООО «Туризм Европа» отражает целесообразность проектного варианта АИС, поскольку снижает стоимостные затраты после внедрения АИС ежемесячно в 2,3 раза.

Показатель срока окупаемости отразит целесообразность и вес капитальных вложений в объёме затрат ООО «Туризм Европа» и приобретаемой выгоды от внедрения АИС управления договорами.

Срок окупаемости производится по формуле (10) (ТОК):

$$ТОК = (КП/\Delta C) \times п, \quad (10)$$



где КП – капитальные затраты на создание проекта, рубль;

п – число периодов, количество.

Показатель числа периодов – дни. Рабочих календарных дней в месяце 21 день.

$$\text{ТОК} = \frac{72\,800}{130\,000} \times 21 = 11,76$$

Проектное решение АИС управления договорами на ООО «Туризм Европа» полностью окупится за полных 12 дней (11,76 дня) и высвободит неэффективно используемые денежные средства для целей организации и получения прибыли.

Снижение стоимостных затрат в день после внедрения АИС (снижение ФОТ специалистов по управлению договорами на ООО «Туризм Европа») рассчитывается по формуле (11):

$$\Delta C_d = \frac{\Delta C}{n}, \quad (11)$$

где  $\Delta C_d$  – снижение стоимостных затрат в день после внедрения АИС управления договорами, рубль.

$$\Delta C_d = \frac{130\,000}{21} = 6\,190,47$$

Снижение стоимостных затрат в день после внедрения АИС управления договорами на ООО «Туризм Европа» составит 6 190, 47 рублей. Данный показатель позволит рассчитать наращиваемый денежный поток после капитальных вложений в размере 72 800 рублей, которые на рисунке 3 отображены отрицательным результатом в 0 день, как единовременное вложение.

В первый день капитальные вложения постепенно будут сокращаться на сумму снижения стоимостных затрат в день после внедрения АИС, а именно на постоянную величину 6 190, 47 рублей. Как видно из рисунка 44, полная окупаемость капитальных затрат произойдет на 12 день, и, далее, будет генерировать доход.

Собрав рассчитанные показатели экономической эффективности проектного варианта АИС управления договорами на ООО «Туризм Европа» и базового варианта в таблице 5 можно сделать вывод об экономической обоснованности внедрения АИС.

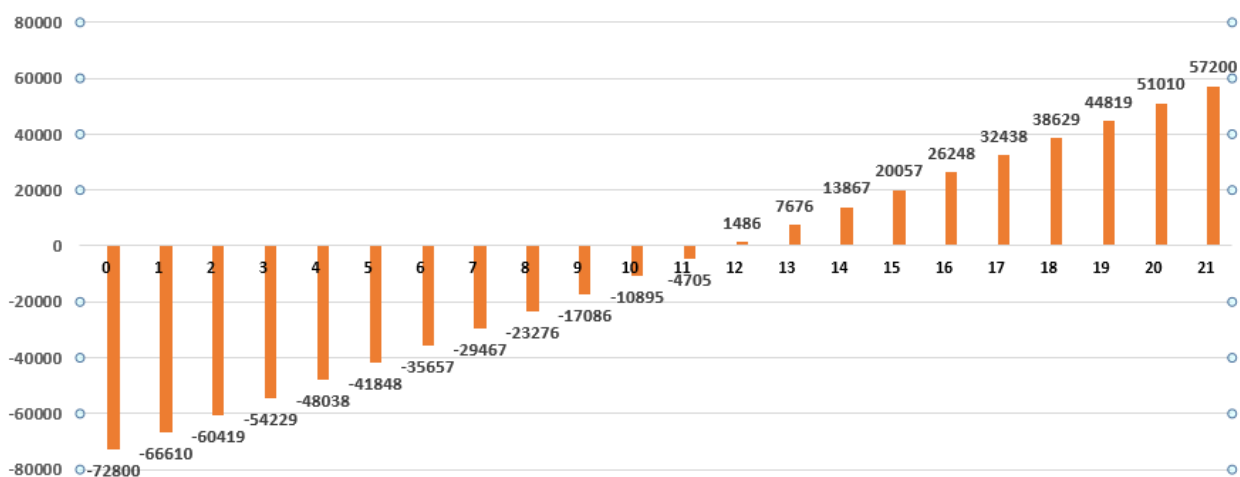


Рисунок 44 – Нарастиваемый денежный поток после КП

Для наглядности данные о стоимостных затратах представлены за период месяца и года.

Трудовые затраты в проектом варианте АИС сократятся в 4,5 раз, в соотношении с базовым вариантом сократятся на 77, (7) %, количественное изменение – 70 минут, а трудоёмкость обработки и оформления 1 клиента составит 20 минут. Экономия от повышения производительности труда – 350%.

Таблица 5 – Показатели эффективности от внедрения АИС управления договорами ООО «Туризм Европа»

	Затраты		Абсолютное изменение затрат	Коэффициент изменения затрат	Индекс изменения затрат
	Базовый вариант	Проектный вариант			
Трудоёмкость	1,5ч (90мин)	1,(3)ч (20мин)	1,1(6)ч (70 мин)	77,(7)%	4,5
Стоимость (месяц), рубли	227 500	97 500	130 000	57,14%	2,3
Стоимость (год), рубли	2 730 000	1 170 000	1 560 000	57,14%	2,3

Стоимостные затраты в проектном варианте АИС управления договорами составят 1 170 000 рублей в год, что на 1 560 000 рублей или 57,14 % меньше по сравнению с базовым вариантом на ООО «Туризм Европа» в размере 2 730 000 рублей. Данная разница достигнута за счёт увеличения производительности труда и сокращения штата сотрудников с 7 человек до 3 человек. Ежемесячная экономия составит 130 000 рублей. Капитальные затраты на полную оплату работы приглашённого специалиста и обучение штатного системного администратора для целей последующего обслуживания АИС составляет 72 800 рублей. Срок окупаемости составляет 12 полных дней (11,76 дня), что говорит о полной окупаемости в пределах отчётного месяца.

Трудоёмкость сократится в 4,5 раз, а стоимостные затраты в 2,3 раза. Автоматизацию операций в стоимостном выражении можно оценить относительно, поскольку технологические решения не являются прямым источником дохода. Автоматизация выступает вспомогательным ресурсом получения прибыли и сокращения затрат.

Косвенный эффект от внедрения АИС управления договорами по управлению договорами на ООО «Туризм Европа»: создание собственной разработки под цели организации, гибкость АИС при доработке,

возможность обслуживать АИС штатным системным администратором за счёт обучения у приглашённого специалиста, централизованная база данных клиентов для последующей работы с ней как с каналом рекламы, повышение оперативности управления и увеличения скорости принятия решений вышестоящим звеном управления, увеличение качества и скорости при обработке данных, лёгкость использования АИС управления договорами непосредственными пользователями.

### **Выводы по главе 3**

Оценка и обоснование экономической эффективности проектного варианта АИС управления договорами на ООО «Туризм Европа» было произведено за счет решения следующих задач:

- расчёта прямого эффекта;
- оценки косвенного эффекта;
- оценки выгоды и необходимости внедрения АИС;
- оценки целесообразности капитальных затрат на разработку АИС.

Трудовые затраты в проектом варианте АИС управления договорами сократятся в 4,5 раз, в соотношении с базовым вариантом сократятся на 77, (7) %, количественное изменение - 70мин, а трудоёмкость обработки и оформления 1 клиента составит 20 минут.

Стоимостные затраты в проектом варианте АИС сократятся за год на 1 560 000 рублей или 57,14 % по сравнению с базовым вариантом на ООО «Туризм Европа». Данная разница достигнута за счёт увеличения производительности труда и сокращения штата сотрудников по обработке учтённых заявок с 7 человек до 3 человек.

Капитальные затраты на полную оплату работы приглашённого специалиста и обучение системного администратора для целей последующего технического обслуживания АИС управления договорами

составит 72 800 рублей. Срок окупаемости составит 12 полных дней (11,76 дня), что говорит о полной окупаемости в пределах отчётного месяца и низкой степени риска вложений в разработку АИС.

Косвенный и прямой эффекты от внедрения АИС управления договорами на ООО «Туризм Европа» достигнуты, что улучшит экономические показатели хозяйственной деятельности организации.

Разработка и внедрение АИС управления договорами ООО «Туризм Европа» экономически эффективны, целесообразны и необходимы.

## Заключение

Итогом проделанной работы является разработанная информационная система управления договорами туристического предприятия, которая полностью удовлетворяет основным требованиям, предъявленным в задании, а также реализует заявленные функции.

Для достижения цели по реализации программного продукта был произведен анализ туристической деятельности и бизнес процессов ООО «Туризм-Европа», составлены модели IDEF0 и их декомпозиции, с помощью которых были выявлены недостатки существующих бизнес-процессов, послужившие обоснованием для разработки АИС управления договорами.

В ходе логического проектирования были построены и описаны все необходимые диаграммы, отражающие специфику деятельности туристической фирмы, для которой разрабатывается АИС.

В ходе физического проектирования была выбрана наиболее подходящая архитектура АИС, а также разработаны и описаны функциональные и структурные схемы проекта, с последующей демонстрацией примеров реализации конечной АИС с их подробным описанием.

В заключении были произведены экономические расчеты для обоснования экономической эффективности проекта, результаты которых подтвердили необходимость и эффективность разработки АИС.

Разработанная информационная система управления договорами автоматизирует неэффективные процессы учёта и управления данными ООО «Туризм Европа», что исключит фактор ошибок при формировании документации, позволит минимизировать временные затраты процесса «Управление договорами». Система АИС управления договорами полностью автоматизирует процесс расчёта стоимости тура, в том числе в иностранной валюте, который ранее выполнялся вручную.

Всё это было достигнуто за счёт инструментов SQLite, Visual Studio и платформы .NET Framework, которые были использованы в процессе разработки АИС управления договорами ООО «Туризм Европа».

## Список используемой литературы и источников

1. Албахари. С# 5.0. Справочник. Полное описание языка: Учебное пособие / Албахари, Джозеф, Бен : Пер. с англ. - М. : ООО «И.Д. Вильямс», 2016. – 1008с.
2. Бодров, О.А. Предметно-ориентированные экономические информационные системы: Учебник для вузов / О.А. Бодров. - М.: Гор. линия-Телеком, 2016. – 244 с.
3. Буч Гради, Введение в UML от создателей языка / Гради Буч, Джеймс Рамбо, Ивар Якобсон. - М.: ДМК Пресс, 2015. – 496 с.
4. Варзунов А. В. Анализ и управление бизнес-процессами [Электронный ресурс]: учеб.пособие / А. В. Варзунов, Е. К. Торосян, Л. П. Сажнева. – Санкт-Петербург: Университет ИТМО, 2016. – 114 с.
5. Варфоломеева, А.О. Информационные системы предприятия: Учебное пособие / А.О. Варфоломеева, А.В. Коряковский, В.П. Романов. - М.: НИЦ ИНФРА-М, 2017. – 283 с.
6. Васильков, А.В. Информационные системы и их безопасность: Учебное пособие / А.В. Васильков, А.А. Васильков, И.А. Васильков. - М.: Форум, 2013. – 528 с.
7. Голицына, О. Л. Системы управления базами данных : учеб. пособие / О. Л. Голицына, Т. Л. Партыка, И. И. Попов. - Гриф МО. – М. : ФОРУМ - ИНФРА-М, 2011. – 431 с.
8. Гринберг А.С. Информационные технологии управления [Электронный ресурс]: учебное пособие для вузов / А.С. Гринберг, Н.Н. Горбачев, А.С. Бондаренко. – Электрон.текстовые данные. – М.: ЮНИТИ-ДАНА, 2017. – 478 с. – 5-238-00725-6.
9. Елиферов В.Г. Бизнес-процессы: регламентация и управление: учебное пособие [для студентов вузов] / В. Г. Елиферов, В. В. Репин - Институт экономики и финансов "Синергия". - Москва : ИНФРА-М, 2016. – 319 с.



10. Золотов С. Ю. Проектирование информационных систем [Электронный ресурс] : учеб.пособие / С. Ю. Золотов ; Томский гос. ун-т систем управления и радиоэлектроники. - Томск : Эль Учебное пособие Контент, 2013. – 86 с. – ISBN 978-5-4332-0083-8.
11. Коцюба, И.Ю. Основы проектирования информационных систем. Учебное пособие / И.Ю.Коцюба, А.В. Чунаев, А.Н. Шиков. – СПб: Университет ИТМО, 2015. – 206 с.
12. Ларман Крэг, Применение UML 2.0 и шаблонов проектирования. Введение в объектно-ориентированный анализ, проектирование и итеративную разработку / Крэг Ларман. – М.: Вильямс, 2013. – 736 с.
13. Мезенцев, К.Н. Автоматизированные информационные системы: Учебник для студентов учреждений среднего профессионального образования / К.Н. Мезенцев. - М.: ИЦ Академия, 2013. – 176 с.
14. Рихтер Дж., CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд. / Дж. Рихтер — СПб.: Питер, 2015. – 896 с.
15. Рыбников А. И. Информационные системы управления производственной компанией. Учебник - М.: Юрайт, 2016. – 356 с.
16. Сайт CyberForum.ru Форум программистов. C# .NET URL: <https://www.cyberforum.ru/csharp-net/> (дата обращения 08.03.2019).
17. Сайт MENANIT.COM Сайт о программировании. Все операции с БД в графическом приложении URL: <https://metanit.com/sharp/adonet/3.5.php> (дата обращения 01.04.2020).
18. Сайт MENANIT.COM Сайт о программировании. Работа с SQLite URL: <https://metanit.com/sharp/wpf/21.1.php> (дата обращения 05.04.2020).
19. Сайт MENANIT.COM Сайт о программировании. Руководство по WPF URL: <https://metanit.com/sharp/wpf/> (дата обращения 10.03.2020).
20. Самуйлов К.Е. Основы формальных методов описания бизнес-процессов. Учебное пособие / К.Е. Самуйлов, Н.В. Серебренникова, А.В. Чукарин – Москва: РУНД, 2013 – 130с.

21. Самуйлов К.Е. Основы формальных методов описания бизнес-процессов. Учебное пособие / К.Е. Самуйлов, Н.В. Серебренникова, А.В. Чукарин – Москва: РУНД, 2013 – 130с.
22. Троелсен Эндрю C# 6.0 and the .NET 4.6 Framework / Эндрю Троелсен / М.: Вильямс, 2016. – 1440 с.
23. Уотсон Карли Beginning Visual C# 2016 Programming / Карли Уотсон, Кристиан Нейгел, Якоб Хаммер Педерсен / М.: Вильямс, 2016. – 960 с.
24. Хетагуров Я. А. Проектирование автоматизированных систем обработки информации и управления (АСОИУ). Учебник - М.: Бином. Лаборатория знаний, 2015. – 240 с.
25. Albahari J., Albahari B., C# 6.0 in a Nutshell: The Definitive Reference. Moscow: O'Reilly, 2018.
26. Anderson C. Essential Windows Presentation Foundation (WPF) 1st Edition. Moscow: Don Box and Chris Sells, 2015.
27. Blethyn S. G. Designing Information Systems / S. G. Blethyn, C. Y. Parker. – London. : Butterworth-Heinemann, 2014.
28. Davies A. Async in C# 5.0. Moscow: O'Reilly, 2018.
29. Nathan A. Windows Presentation Foundation Unleashed (WPF) 1st Edition. Moscow: Sams, 2016.



**ДОГОВОР № 1**  
**о реализации туристического продукта**

город Тольятти 22.05.2020

**Турагент:**  
Общество с ограниченной ответственностью "Туризм Европа" (сокращенное наименование ООО "Туризм Европа" \_\_\_\_\_,  
в лице: директора Викторова Виктора Викторовича \_\_\_\_\_,  
действующего на основании Устава \_\_\_\_\_,  
именуемого в дальнейшем Турагентство,  
с одной стороны,  
и, именуемый далее Заказчик: Казимиров Антон Петрович \_\_\_\_\_,  
действующего на основании \_\_\_\_\_,  
с другой стороны, вместе именуемые Стороны, заключили настоящий договор (далее - Договор) о нижеследующем:

**1. ПРЕДМЕТ ДОГОВОРА**

1.1. В соответствии с Договором Турагент обязуется обеспечить оказание Заказчику комплекса услуг по перевозке и размещению и иных услуг (далее - Туристский продукт) и несёт перед Заказчиком ответственность за неоказание или ненадлежащее оказание услуг, входящих в туристический продукт, независимо от того кем должны были оказаться или оказывались эти услуги, а Заказчик обязуется оплатить Туристский продукт.

1.2. Информация о потребительских свойствах Туристического продукта:

Страна: Абхазия	Класс отеля: 4
Курорт: Новый Афон	Питание: RO - Только комната
Дата начала: 43899	Кол-во туристов: 2
Дата окончания: 43902	Кол-во ночей: 3
Отель: Царская Аллея	Стоимость: 14500

1.3. Турист, совершающий путешествие на основании настоящего договора:

№	ФИО	Дата рождения	Данные паспорта	Адрес	Телефон
1	Казимиров Антон Петрович	15.03.1985	4611 809723, 14.01.2022	Пирогова 6, кв.1	89347242364

**2. СТОИМОСТЬ ТУРПРОДУКТА, ПОРЯДОК РАСЧЁТА**

2.1. Полная стоимость турпродукта:  
на 2 человек 14500 руб. (четырнадцать тысяч пятьсот рублей).

2.2. Оплата осуществляется Заказчиком путем перечисления денежных средств на расчётный счёт Турагента или вносится наличными в кассу Турагента не позднее 14 дней до начала поездки. Оплата производится в российских рублях по курсу ЦБ на день оплаты. Моментом оплаты считается поступление денежных средств в кассу или на расчётный счёт Турагента.

2.3. В случае, если поездка отменена по решению Турагента, Заказчику выплачивается вся уплаченная им сумма в течение 14 календарных дней. Полный возврат этой суммы освобождает Турагентство от всех обязательств по настоящему договору и никакая компенсация при этом не предусматривается.

1 из 30

Рисунок А.2 – Печатная форма договора между турагентом и туристом

**6. АДРЕСА, РЕКВИЗИТЫ И ПОДПИСИ СТОРОН**

**ТУРАГЕНТ:**  
 Полное наименование: Туризм Европа  
 Адрес местонахождения: г. Тольятти Полякова 16  
 Почтовый адрес: 125319  
 ОГРН:  
 ИНН: 7826669006  
 КПП: 783901001  
 р\с: 40702810600000030409  
 к\с: 30101810100000000716  
 БИК: 044525716

Генеральный директор  
 " " 202\_года (подпись)  
 М.П.

**ЗАКАЗЧИК:**  
 ФИО: Казимиров Антон Петрович  
 Дата рождения: 15.03.1985  
 Серия и номер паспорта: 4611 809723, 14.01.2022  
 Адрес регистрации: Пирогова 6, кв.1

" " 202\_года (подпись)

30 из 30

Рисунок А.3 – Продолжение договора между турагентом и туристом

<b>Время/Дата</b>	<b>Стоимость</b>	<b>Страна</b>
12.03.2020	63500	Абхазия
20.03.2020	81000	Беларусь
01.03.2020	60000	Израиль
06.03.2020	22000	Испания
02.03.2020	9700	Россия



Рисунок А.4 – Печатная форма отчета продаж за месяц

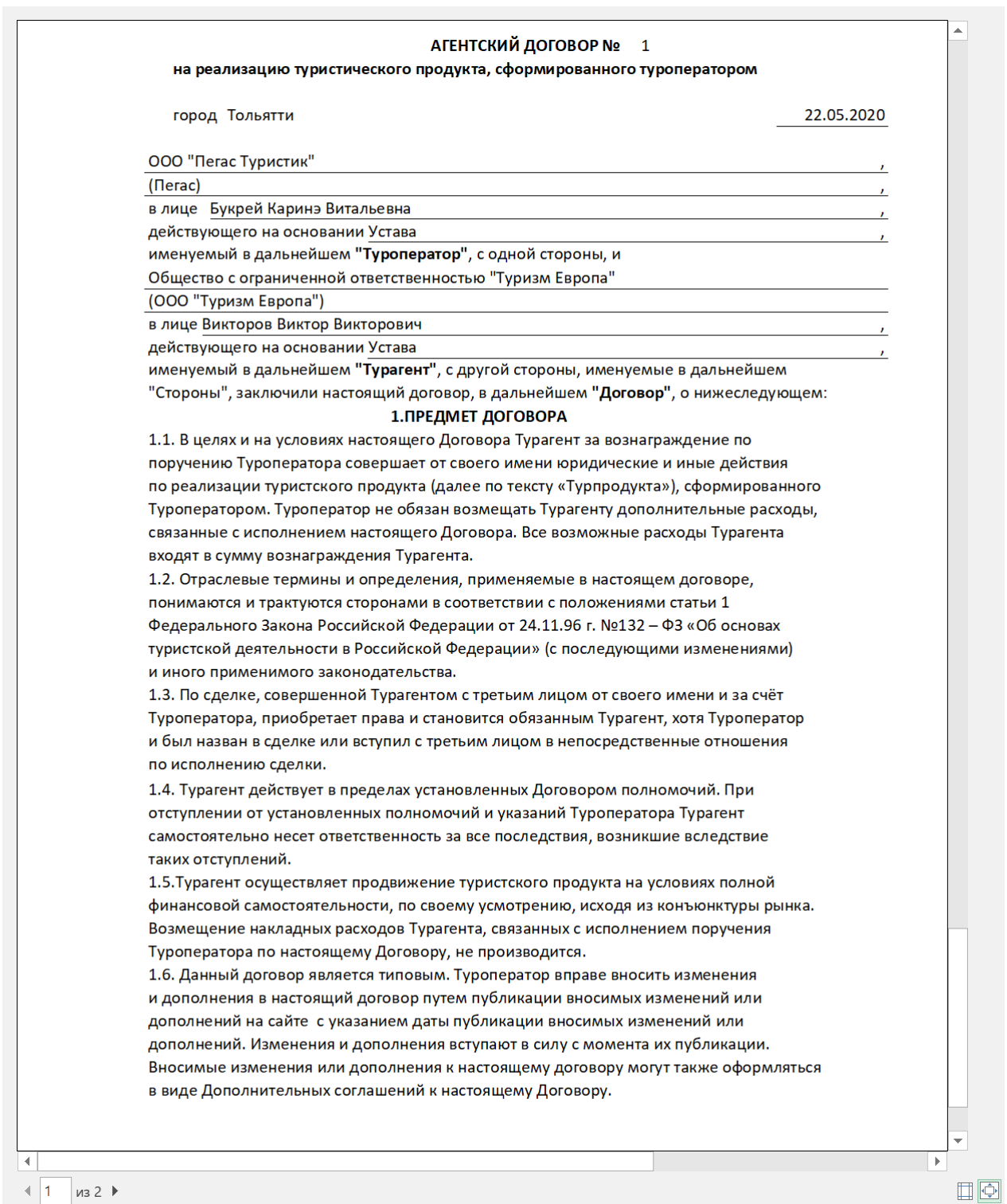


Рисунок А.5 – Печатная форма договора между турагентом и туроператором

**11. ЮРИДИЧЕСКИЕ АДРЕСА И БАНКОВСКИЕ РЕКВИЗИТЫ СТОРОН**

**ТУРОПЕРАТОР:**  
 ООО "Пегас Туристик"  
 Юр. Адрес: г. Москва, ул. Коккинаки, д. 4, пом. IV  
 Почтовый адрес 125319  
 ИНН: 7714126996  
 КПП: 771401001  
 р\с: 40702810516000912601  
 к\с: 30101810000000000920  
 БИК: 044030920

**ТУРАГЕНТ:**  
 Общество с ограниченной ответственностью "Туризм Европа"  
 Юр. Адрес: г. Тольятти Полякова 16  
 Почтовый адрес \_\_\_\_\_  
 ИНН: 7826669006  
 КПП: 783901001  
 р\с: 40702810600000030409  
 к\с: 30101810100000000716  
 БИК: 044525716

**12. ПОДПИСИ СТОРОН**

Туроператор  
 " " \_\_\_\_\_ 202\_года  
 (подпись) \_\_\_\_\_

Турагент  
 " " \_\_\_\_\_ 202\_года  
 (подпись) \_\_\_\_\_

**М.П.**  
 \_\_\_\_\_  
**М.П.**  
 \_\_\_\_\_

2 из 3

Рисунок А.6 – Продолжение договора между турагентом и туроператором

Страна	Путевки
Абхазия	3
Испания	1



Рисунок А.7 – Отчет по наиболее/наименее продаваемым за месяц странам

## Приложение Б

### Макеты и программная реализация продукта

	A	B	C	D
1				
2		<b>Договор №</b>	<b>1</b>	
3			<b>Туроператор</b>	
4		Полное наименование	ООО "Пегас Туристик"	
5		Сокращённое	Пегас	
6		и.о.	Букрей Каринэ Витальевна	
7		основание	Устава	
8		Юр. Адрес	г. Москва, ул. Коккинаки, д. 4, пом. IV	
9		Почтовый адрес	125319	
10		ИНН	7714126996	
11		КПП	771401001	
12		р\с	40702810516000912601	
13		к\с	3010181000000000920	
14		БИК	044030920	
15			<b>Турагент</b>	
16		Полное наименование	Общество с ограниченной ответственностью "Туризм Европа"	
17		Сокращённое	ООО "Туризм Европа"	
18		и.о.	Викторов Виктор Викторович	
19		основание	Устава	
20		Юр. Адрес	г. Тольятти Полякова 16	
21		Почтовый адрес		
22		ИНН	7826669006	
23		КПП	783901001	
24		р\с	40702810600000030409	
25		к\с	30101810100000000716	
26		БИК	044525716	
27				

Рисунок Б.1 – Данные макета договора между турагентством и туроператором

B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1																	
2				<b>АГЕНТСКИЙ ДОГОВОР № 1</b>						<b>11.ЮРИДИЧЕСКИЕ АДРЕСА И БАНКОВСКИЕ РЕКВИЗИТЫ СТОРОН</b>							
3				на реализацию туристического продукта, сформированного туроператором						<b>ТУРОПЕРАТОР:</b>							
4				город Тольятти				22.05.2020		ООО "Пегас Туристик"							
5										Юр. Адрес: г. Москва, ул. Коккинаки, д. 4, пом. IV							
6										Почтовый адрес 125319							
7				ООО "Пегас Туристик"						ИНН: 7714126996							
8				(Пегас)						КПП: 771401001							
9				в лице Букрей Каринэ Витальевна						р\с: 40702810516000912601							
10				действующего на основании Устава						к\с: 3010181000000000920							
11				именуемый в дальнейшем "Туроператор", с одной стороны, и						БИК: 044030920							
12				Общество с ограниченной ответственностью "Туризм Европа"						<b>ТУРАГЕНТ:</b>							
13				(ООО "Туризм Европа")						Общество с ограниченной ответственностью "Туризм Европа"							
14				в лице Викторов Виктор Викторович						Юр. Адрес: г. Тольятти Полякова 16							
15				действующего на основании Устава						Почтовый адрес							
16				именуемый в дальнейшем "Турагент", с другой стороны, именуемые в дальнейшем						ИНН: 7826669006							
17				"Стороны", заключили настоящий договор, в дальнейшем "Договор", о нижеследующем:						КПП: 783901001							
18				<b>1.ПРЕДМЕТ ДОГОВОРА</b>													
19				1.1. В целях и на условиях настоящего Договора Турагент за вознаграждение по													
20				поручению Туроператора совершает от своего имени юридические и иные действия													
21				по реализации туристского продукта (далее по тексту «Турпродукта»), сформированного													
22				Туроператором. Туроператор не обязан возмещать Турагенту дополнительные расходы,													
23				связанные с исполнением настоящего Договора. Все возможные расходы Турагента													
24				входят в сумму вознаграждения Турагента.													
25				1.2. Отраслевые термины и определения, применяемые в настоящем договоре,													
26				понимаются и трактуются сторонами в соответствии с положениями статьи 1													
27				Туроператор													

Рисунок Б.2 – Макет договора между турагентством и туроператором



	A	B	C	D	E	F
1						<b>Турист</b>
2		<b>Договор №</b>	1		Фамилия	Казимиров
3			<b>Директор</b>		Имя	Антон
4					Отчество	Петрович
5		ФИО	Викторова Виктора Викторовича		Дата рождения	15.03.1985
6		должность	директора		паспорт	4611 809723, 14.01.2022
7		основание	Устава		Адрес	Пирогова 6, кв.1
8					Телефон	89347242364
9			<b>Заказчик</b>		Адрес регистрации	Невский пр. 5
10		Фамилия	Казимиров			<b>Турагент</b>
11		Имя	Антон		Полное наименование	Туризм Европа
12		Отчество	Петрович		Адрес местонахождения	г. Тольятти Полякова 16
13		Страна	Абхазия		Почтовый адрес	125319
14		Курорт	Новый Афон		ОГРН	
15		Дата начала	09.03.2020		ИНН	7826669006
16		Дата окончания	12.03.2020		КПП	783901001
17		Отель	Царская Аллея		р\с	40702810600000030409
18		Класс отеля	4		к\с	30101810100000000716
19		Питание	RO - Только комната		БИК	044525716
20		Кол-во туристов	2			
21		Кол-во ночей	3			
22		Стоимость	14500			
23		Пропись	четырнадцать тысяч пятьсот рублей			
24						
25						
26						
27						
28						
29						
30						

Рисунок Б.3 – Макет договора между турагентством и туристом

	A	B	C	D	E	F	G
1		<b>Дата расчёта</b>		22.05.2020	22 мая	2020	
2		<b>Дата приказа</b>		22.11.2015	22 ноября	2015	
3		<b>№</b>		1985			
4		<b>Турагент</b>			<b>Заказчик</b>		
5		Полное наименование	Общество с ограниченной ответственностью "Туризм Европа"		Фамилия	Казимиров	
6		Сокращённое	ООО "Туризм Европа"		Имя	Антон	
7		Место нахождения	г. Тольятти Полякова 16		Отчество	Петрович	
8		Почтовый адрес	г. Тольятти Полякова 16		паспорт:	4611 809723	
9		Телефон	8 (8482) 36 20 40		Место жительства:	Невский пр. 5	
10		ИНН	7826669006		Место нахождения:	Невский пр. 5	
11		№ в реестре (для оператора)			ОКПО(для юр.лица):	-	
12		№ путёвки	0002563				
13		Серия	2369		Вид турпродукта	61101	
14					Кол-во дней на 1	3	
15					Стоимость пропись	четырнадцать тысяч пятьсот рублей	
16					Оплачено наличными с карты (прописью)	четырнадцать тысяч пятьсот рублей	
17					Ответственный	Новиков Олег Васильевич	
18							
19							
20							

Рисунок Б.4 – Макет туристической путевки

## Приложение В

### Листинг программного кода модуля главной формы Win\_dow

```
using System;
using MahApps.Metro.Controls;
using Excel = Microsoft.Office.Interop.Excel;
using System.IO;
using System.Collections.Generic;
using System.Data.SQLite;
using System.Windows.Controls.DataVisualization.Charting;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using static WpfApp3.JobWithClass;
using System.Data;
using WpfApp3.DataModel;
using WpfApp3.Repository;
using WpfApp3.DataAccess;
using System.Net;
using System.Xml.Serialization;
using System.Xml;
using WpfApp3.ParseOfValutes;
namespace WpfApp3
{
    /// <summary>
    /// Логика взаимодействия для Window.xaml
    /// </summary>
    public partial class Win_dow : MetroWindow
    {
        DataSet ds = new DataSet(); //Создаем объект класса DataSet
        SQLiteDataAdapter adapter;
        SQLiteCommandBuilder commandBuilder;

        string sql = "Select * From clients"; //Sql запрос (достать все из таблицы customer)
        string path = "Database.db"; //Путь к файлу БД
        string connectionString = @"Data Source=. \Database.db; Version=3"; // @"Data Source=.\ " + path +
"; Version=3;New=True;Compress=True;"; //Строка соединения (sqlite)

        public List<KeyValuePair<string, int>> values;
        public List<KeyValuePair<string, int>> pie_values;
        public List<PieData> list_pie;
        public List<Data> list_day_data;
        public List<Data> list_week_data;
        public List<Data> list_month_data;
        public List<Data> list_quater_data;
        Replies r = new Replies();

        public Win_dow()
        {
            InitializeComponent();

            #region Для отчетов
            grid_day.Visibility = System.Windows.Visibility.Collapsed;
            grid_month.Visibility = System.Windows.Visibility.Collapsed;
            grid_week.Visibility = System.Windows.Visibility.Collapsed;

```

```

grid_quater.Visibility = System.Windows.Visibility.Collapsed;
grid_stat.Visibility = System.Windows.Visibility.Collapsed;
button_createExcel.IsEnabled = false;
button_excel_month.IsEnabled = false;
button_excel_week.IsEnabled = false;
button_excel_quater.IsEnabled = false;
button_excel_stat.IsEnabled = false;
#endregion

SQLiteConnection conn = new SQLiteConnection(ConnectionString); //Создаем соединение

SQLiteDataAdapter da = new SQLiteDataAdapter(sql, conn); //Создаем объект класса
DataAdapter (тут мы передаем наш запрос и получаем ответ)

da.Fill(ds); //Заполняем DataSet содержимым DataAdapter'a
listview_Clients_Copy.ItemsSource = ds.Tables[0].DefaultView;
conn.Close(); //Закрываем соединение

}
#region Структура для работы с датой статистики
public struct Data
{
    public string time { get; set; }
    public int bill { get; set; } //count
    public int country_id { get; set; } // bouquet_id
}

public struct PieData
{
    public int bill { get; set; } // count
    public int country_id { get; set; } //bouquet_id
    public string name { get; set; } //
}
#endregion

/// <summary>
/// Структура для работы с заказом и его клиентом
/// </summary>
struct Clients_Orders
{
    public new_Clients new_clients { get; set; }
    public Orders order { get; set; }
    public Country country { get; set; }
    public Airports airports { get; set; }
    public Employee employee { get; set; }
    public Operators operators { get; set; }
}

new_Clients selected_client; // Храним выделенного клиента.
Otels selected_otel; // Храним выделенный отель.
Employee selected_employee; // Храним выделенного сотрудника.

/// <summary>
/// Структура для работы с отелями и курортами/странами, в которых они расположены
/// </summary>
struct Otels_country_resorts
{
    public Otels otels { get; set; }
    public Country country { get; set; }
    public Resorts resorts { get; set; }
}

```

```

/// <summary>
/// Структура для работы с сотрудниками и их аккаунтами
/// </summary>
struct Employees_Login
{
    public Employee employee { get; set; }
    public Log_in log_in { get; set; }
}

ClientsRepository oClientsRepository; //
OrdersRepository oOrdersRepository; // Контроллеры
AirportsRepository oAirportsRepository; // Таблиц
CountryRepository oCountryRepository; //
EmployeeRepository oEmployeeRepository; //
OperatorsRepository oOperatorsRepository;
OtelsRepository oOtelsRepository; //
ResortsRepository oResortsRepository;
Log_inRepository oLog_InRepository;

```

по ID

```

SQLite.SQLiteAsyncConnection conn; // Прямой коннект к БД для выдачи записей из таблиц

```

```

Clients_Orders oClients_Orders; // Выделенные в списках объекты
//Carts_Country oCarts_Country; //
Otels_country_resorts oOtels_country_resorts; // Выделенные в списках объекты

```

```

Employees_Login oEmployees_Login; // Выделенные в списках объекты

```

```

/// <summary>
/// Загрузка окна, инициализация контроллеров, загрузка в listview данных и сброс управления.
/// </summary>

```

```

private async void MetroWindow_Loaded(object sender, RoutedEventArgs e)
{

```

```

    DBConnection oDBConnection = new DBConnection();

```

```

    await oDBConnection.InitializeDatabase();

```

```

    oClientsRepository = new ClientsRepository(oDBConnection);
    oOrdersRepository = new OrdersRepository(oDBConnection);
    oAirportsRepository = new AirportsRepository(oDBConnection);
    oCountryRepository = new CountryRepository(oDBConnection);
    oEmployeeRepository = new EmployeeRepository(oDBConnection);
    oOperatorsRepository = new OperatorsRepository(oDBConnection);
    oOtelsRepository = new OtelsRepository(oDBConnection);
    oResortsRepository = new ResortsRepository(oDBConnection);
    oLog_InRepository = new Log_inRepository(oDBConnection);
    //oPaymentsRepository = new PaymentsRepository(oDBConnection);

```

```

    conn = oDBConnection.GetAsyncConnection();

```

```

    // Clients_grid_Copy.Items.Clear();

```

```

    await Update_ListView_Orders();
    await Update_Clients_grid_Copy();
    await Update_Airports_combobox();
    await Update_Country_combobox();
    await UpdateListView_Otels();
    await Update_Operators_combobox();
    await UpdateListView_Resorts();
    await UpdateListView_Employees();

```

```

    Clear_Control();

```

```

    try

```

```

{
    using (var client = new WebClient())
    {
        string USDXml = "";
        string EuroXML = "";

        //Индикация процесса работы
        // downloadRatesFileButton.Text = "Обновление данных...";

        //Скачиваем данные с сайта
        var xml = await client.DownloadStringTaskAsync(new
Uri("http://www.cbr.ru/scripts/XML_daily.asp"));
        if (string.IsNullOrEmpty(xml)) return;
        //Сериализация полученной строки в нужный тип данных
        var serializer = new XmlSerializer(typeof(ValCurs));
        using (TextReader reader = new StringReader(xml))
        {
            var valcurs = serializer.Deserialize(reader) as ValCurs;

            if (valcurs == null) return;
            {
                XmlTextReader readerr = new
XmlTextReader("http://www.cbr.ru/scripts/XML_daily.asp");

                while (readerr.Read())
                { //Проверяем тип текущего узла
                    switch (readerr.NodeType)
                    {
                        //Если этого элемент Valute, то начинаем анализировать атрибуты
                        case XmlNodeType.Element:
                            if (readerr.Name == "Valute")
                            {
                                if (readerr.HasAttributes)
                                {
                                    //Метод передвигает указатель к следующему атрибуту
                                    while (readerr.MoveToNextAttribute())
                                    {
                                        if (readerr.Name == "ID")
                                        {
                                            //Если значение атрибута равно R01235, то перед нами информация
                                            о курсе доллара

                                            if (readerr.Value == "R01235")
                                            {
                                                //Возвращаемся к элементу, содержащий текущий узел атрибута
                                                readerr.MoveToElement();
                                                //Считываем содержимое дочерних узлом
                                                USDXml = readerr.ReadOuterXml();
                                            }
                                        }
                                    }
                                    //Аналогичную процедуру делаем для ЕВРО
                                    if (readerr.Name == "ID")
                                    {
                                        if (readerr.Value == "R01239")
                                        {
                                            readerr.MoveToElement();
                                            EuroXML = readerr.ReadOuterXml();
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
        break;
    }
}

```

```

    }
    }
    //Из выдернутых кусков XML кода создаем новые XML документы
    XmlDocument usdXmlDocument = new XmlDocument();
    usdXmlDocument.LoadXml(USDXml);
    XmlDocument euroXmlDocument = new XmlDocument();
    euroXmlDocument.LoadXml(EuroXML);
    //Метод возвращает узел, соответствующий выражению XPath
    XmlNode xmlNode = usdXmlDocument.SelectSingleNode("Valute/Value");

    //Считываем значение и конвертируем в decimal. Курс валют получен
    decimal usdValue = Convert.ToDecimal(xmlNode.InnerText);
    xmlNode = euroXmlDocument.SelectSingleNode("Valute/Value");
    decimal euroValue = Convert.ToDecimal(xmlNode.InnerText);

    //Отображаем данные в dataGridView1 через bindingSource1
    parseUSD.Content = Math.Round(usdValue, 2, MidpointRounding.AwayFromZero);
    parseEUR.Content = Math.Round(euroValue, 2, MidpointRounding.AwayFromZero);
    //Отображаем дату
    //currentDateLabel.Text = valcurs.Date;
    }
    }
    }
    }
    catch (Exception x) { MessageBox.Show(x.ToString()); }
}

```

```

/// <summary>
/// Загрузка и обновление сотрудников
/// </summary>
private async Task UpdateListView_Employees()
{
    List<Employees_Login> IEmployees_Login = new List<Employees_Login>();
    List<Employee> IEmployee = new List<Employee>();
    List<Log_in> ILog_in = new List<Log_in>();

    IEmployee = await oEmployeeRepository.Select_All_Employee_Async();

    foreach (var o in IEmployee)
    {
        Employees_Login employees_login = new Employees_Login();

        employees_login.employee = o;
        employees_login.log_in = await conn.GetAsync<Log_in>(o.log_id);

        IEmployees_Login.Add(employees_login);
    }
    listview_Employees.ItemsSource = IEmployees_Login;
}

```

```

/// <summary>
/// Загрузка и обновление отелей
/// </summary>
private async Task UpdateListView_Otels()
{
    List<Otels_country_resorts> IOtels_country_resorts = new List<Otels_country_resorts>();
    List<Otels> IOtels = new List<Otels>();
    List<Country> ICountry = new List<Country>();
    List<Resorts> IResorts = new List<Resorts>();

    IOtels = await oOtelsRepository.Select_All_Otels_Async();
}

```

```

foreach (var o in IOtels)
{
    Otels_country_resorts otels_country_resorts = new Otels_country_resorts();

    otels_country_resorts.otels = o;
    otels_country_resorts.country = await conn.GetAsync<Country>(o.country_id);
    otels_country_resorts.resorts = await conn.GetAsync<Resorts>(o.resorts_id);

    IOtels_country_resorts.Add(otels_country_resorts);
}
Otels_combobox.ItemsSource = IOtels_country_resorts;
combo_otel.ItemsSource = IOtels_country_resorts;
listview_Otels.ItemsSource = IOtels_country_resorts;
}

/// <summary>
/// Обновление listview с Заказами. Ставим в соответствие каждому заказу его клиента.
/// </summary>
private async Task Update_ListView_Orders()
{
    List<Clients_Orders> IClients_Orders = new List<Clients_Orders>();
    List<Orders> IOrders = new List<Orders>();
    List<Country> ICountry = new List<Country>();
    List<Airports> IAirports = new List<Airports>();
    List<Employee> IEmployee = new List<Employee>();
    List<Operators> IOperators = new List<Operators>();

    IOrders = await oOrdersRepository.Select_All_Orders_Async();

    foreach (var o in IOrders)
    {
        Clients_Orders clients_orders = new Clients_Orders();

        clients_orders.order = o;
        clients_orders.new_clients = await conn.GetAsync<new_Clients>(o.clients_id);
        clients_orders.country = await conn.GetAsync<Country>(o.country_id);
        clients_orders.airports = await conn.GetAsync<Airports>(o.airports_id);
        clients_orders.employee = await conn.GetAsync<Employee>(o.employee_id);
        clients_orders.operators = await conn.GetAsync<Operators>(o.operators_id);

        IClients_Orders.Add(clients_orders);
    }
    listview_Orders.ItemsSource = IClients_Orders;
}

/// <summary>
/// Загрузка и обновление заказов
/// </summary>
private async Task Update_ListView_Orders(string name)
{
    List<Clients_Orders> IClients_Orders = new List<Clients_Orders>();
    List<Orders> IOrders = new List<Orders>();
    List<Country> ICountry = new List<Country>();
    List<Airports> IAirports = new List<Airports>();
    List<Employee> IEmployee = new List<Employee>();
    List<Operators> IOperators = new List<Operators>();

    IOrders = await oOrdersRepository.Select_All_Orders_Async();

    foreach (var o in IOrders)
    {
        Clients_Orders clients_orders = new Clients_Orders();

```

```

        clients_orders.order = o;
        clients_orders.new_clients = await conn.GetAsync<new_Clients>(o.clients_id);
        clients_orders.country = await conn.GetAsync<Country>(o.country_id);
        clients_orders.airports = await conn.GetAsync<Airports>(o.airports_id);
        clients_orders.employee = await conn.GetAsync<Employee>(o.employee_id);
        clients_orders.operators = await conn.GetAsync<Operators>(o.operators_id);
        if
(clients_orders.new_clients.lastname.ToLower().Contains(textbox_Search_Orders.Text.ToLower()))
        {
            IClients_Orders.Add(clients_orders);
        }
    }
    listview_Orders.ItemsSource = IClients_Orders;
}
/// <summary>
/// Обновление listview с договорами. Ставим в соответствие каждому договору его
контрагента.
/// </summary>
private async Task Update_ListView_Contracts()
{
    if (Contragent_box.SelectedIndex == 0) //Фильтр по юридическим лицам
    {
        List<Contracts> result = await oContractsRepository.Select_Contracts_Async("select * from
Contracts where clients_id != -1");// SELECT DISTINCT Contracts.clients_id, clients.Lastname FROM Contracts
LEFT JOIN clients ON Contracts.clients_id = clients.clients_id where Contracts.clients_id != -1;");
        listview_Contracts.ItemsSource = result;
    }
    if (Contragent_box.SelectedIndex == 1) //Фильтр по физическим лицам
    {
        List<Contracts> result = await oContractsRepository.Select_Contracts_Async("SELECT
DISTINCT Contracts.operators_id, Operators.Name FROM Contracts LEFT JOIN Operators ON
Contracts.operators_id = Operators.operators_id where Contracts.operators_id != -1;");
        listview_Contracts.ItemsSource = result;
    }
}
/// <summary>
/// Сброс управления
/// </summary>
private void Clear_Control()
{
    listview_Orders.SelectedIndex = -1;
    listview_Clients_Copy.SelectedIndex = -1;
}
/// <summary>
/// Выделение заказа в listview Заказов ЛКМ. Включение управления соответствующих
элементов. Обновление listview Корзины, соответствующей выделенному заказу
/// </summary>
private async void listview_Orders_MouseLeftButtonUp(object sender, MouseButtonEventArgs e)
{
    try
    {
        oClients_Orders = (Clients_Orders)listview_Orders.SelectedItem;
    }
    catch
    {
        System.Windows.Forms.MessageBox.Show("Клиент не выбран!");
    }
}

/// <summary>
/// Снятие выделение с заказа в listview Заказов ПКМ. Сброс управления
/// </summary>

```



```

private void listView_Orders_MouseRightButtonUp(object sender, MouseButtonEventArgs e)
{
    Clear_Control();
}

/// <summary>
/// ПОИСК ПО ФИО В ЗАКАЗАХ
/// </summary>
private async void textbox_Search_Orders_TextChanged(object sender, TextChangedEventArgs e)
{
    if (textbox_Search_Orders.Text != "")
    {
        await Update_ListView_Orders(textbox_Search_Orders.Text);
    }
    else
    {
        await Update_ListView_Orders();
    }
}

private void Add_Click(object sender, RoutedEventArgs e)    {
    AddEdit_Clients_win add_ed_client = new AddEdit_Clients_win();
    add_ed_client.Show();

    DataRow row = ds.Tables[0].NewRow(); // добавляем новую строку в DataTable
    ds.Tables[0].Rows.Add(row);
}

private void Button_Click(object sender, RoutedEventArgs e)
{
    ReportWindow win2 = new ReportWindow();
    win2.Show();
}

private void Edit_Click(object sender, RoutedEventArgs e)
{
    MainTabControl.SelectedIndex = 0;
}

private void Del_Click(object sender, RoutedEventArgs e)
{
    удаляем выделенные строки из dataGridView1
    foreach (DataGridRow row in Clients_grid_Copy.SelectedItems) //.SelectedRows)
    {
        Clients_grid_Copy.Items.Remove(row); //Rows.Remove(row);
    }
}

private void Save_Click(object sender, RoutedEventArgs e)
{
    using (SQLiteConnection connection = new SQLiteConnection(@"Data Source=. \Database.db"))
    {
        connection.Open();
        adapter = new SQLiteDataAdapter(sql, connection);
        commandBuilder = new SQLiteCommandBuilder(adapter);
        adapter.InsertCommand = new SQLiteCommand("clients", connection);
        adapter.InsertCommand.Parameters.Add(new SQLiteParameter("@lastname", DbType.String,
50, "Lastname"));
        adapter.InsertCommand.Parameters.Add(new SQLiteParameter("@name", DbType.String, 50,
"Name"));
        adapter.InsertCommand.Parameters.Add(new SQLiteParameter("@phone", DbType.String, 50,
"Phone"));
    }
}

```

```

        adapter.InsertCommand.Parameters.Add(new SQLiteParameter("@address", DbType.String, 50,
"Adress"));
        adapter.InsertCommand.Parameters.Add(new SQLiteParameter("@email", DbType.String, 50,
"Email"));
        adapter.InsertCommand.Parameters.Add(new SQLiteParameter("@birthday", DbType.String, 50,
"Birthday"));

        SQLiteParameter parameter = adapter.InsertCommand.Parameters.Add("@clients_id",
DbType.Int32, 0, "clients_id");
        parameter.Direction = ParameterDirection.Output;

        adapter.Update(ds);
    }
}

/// <summary>
/// ПОИСК КЛИЕНТОВ
/// </summary>
private async void textbox_Search_Clients_TextChanged(object sender, TextChangedEventArgs e)
{
    if (textbox_Search_Clients.Text != "")
    {
        await Update_Clients_grid_Copy(textbox_Search_Clients.Text);
    }
    else
    {
        await Update_Clients_grid_Copy();
    }
}

private async Task Update_Clients_grid_Copy(string name)
{
    List<new_Clients> query = await oClientsRepository.Select_All_Clients_Async();

    List<new_Clients> result = new List<new_Clients>();

    foreach (var b in query)
    {
        if (b.lastname.ToLower().Contains(name.ToLower()))
        {
            result.Add(b);
        }
    }
    listview_Clients_Copy.ItemsSource = result;
}

/// <summary>
/// Загрузка клиентов
/// </summary>
private async Task Update_Clients_grid_Copy()
{
    List<new_Clients> result = await oClientsRepository.Select_All_Clients_Async();

    listview_Clients_Copy.ItemsSource = result;
}

/// <summary>
/// Загрузка аэропортов
/// </summary>
private async Task Update_Airports_combobox()
{
    List<Airports> result = await oAirportsRepository.Select_All_Airports_Async();

```

```

        combo_From.ItemsSource = result;
        listview_Airports.ItemsSource = result;
    }

    /// <summary>
    /// Загрузка туроператоров
    /// </summary>
    private async Task Update_Operators_combobox()
    {
        List<Operators> result = await oOperatorsRepository.Select_All_Operators_Async();

        Operators_combobox.ItemsSource = result;
        listview_Operators.ItemsSource = result;
    }

    /// <summary>
    /// Загрузка курортов
    /// </summary>
    private async Task UpdateListView_Resorts()
    {
        List<Otels_country_resorts> lOtels_country_resorts = new List<Otels_country_resorts>();
        List<Country> lCountry = new List<Country>();
        List<Resorts> lResorts = new List<Resorts>();

        lResorts = await oResortsRepository.Select_All_Resorts_Async();

        foreach (var o in lResorts)
        {
            Otels_country_resorts otels_country_resorts = new Otels_country_resorts();

            otels_country_resorts.resorts = o;
            otels_country_resorts.country = await conn.GetAsync<Country>(o.country_id);

            lOtels_country_resorts.Add(otels_country_resorts);
        }
        listview_Resorts.ItemsSource = lOtels_country_resorts;

        Resorts_combobox.ItemsSource = lResorts;
        combo_resort.ItemsSource = lResorts;
    }

    /// <summary>
    /// Загрузка курортных стран/направлений
    /// </summary>
    private async Task Update_Country_combobox()
    {
        List<Country> result = await oCountryRepository.Select_All_Country_Async();

        combo_To.ItemsSource = result;
        combo_country.ItemsSource = result;
        listview_Country.ItemsSource = result;
    }

    #region ОТЧЕТЫ
    private void menu_lv12_day_Click(object sender, RoutedEventArgs e)
    {
        grid_day.Visibility = System.Windows.Visibility.Visible;
        grid_month.Visibility = System.Windows.Visibility.Collapsed;
        grid_week.Visibility = System.Windows.Visibility.Collapsed;
        grid_quater.Visibility = System.Windows.Visibility.Collapsed;
    }

```

```

        grid_stat.Visibility = System.Windows.Visibility.Collapsed;
    }
    /// <summary>
    /// построение графика - отчет за день
    /// </summary>
    private async void button_build_Click(object sender, RoutedEventArgs e)
    {
        for (int i = 0; i < day_chart.Series.Count; i++)
        {
            day_chart.Series.Remove(day_chart.Series[i]);
        }
        values = new List<KeyValuePair<string, int>>();
        List<Orders> list_orders = await oOrdersRepository.Select_All_Orders_Async();
        List<Data> list_data = new List<Data>();
        for (int i = 0; i < list_orders.Count; i++)
        {
            var datetime = DateTime.Parse(list_orders[i].date_order);
            Data d = new Data();
            if (datetime.Date == datepicker.SelectedDate)
            {
                List<Orders> list_carts = await oOrdersRepository.Select_Orders_Async("select
bill,country_id from Orders where orders_id=" + list_orders[i].orders_id);
                d.time = datetime.TimeOfDay.ToString();
                foreach (var c in list_carts)
                {
                    d.bill = (int)c.bill;
                    d.country_id = c.country_id;
                    list_data.Add(d);
                }
            }
        }
        list_day_data = list_data;
        List<Country> list_country = await oCountryRepository.Select_All_Country_Async();
        for (int i = 0; i < list_country.Count; i++)
        {
            values = new List<KeyValuePair<string, int>>();
            for (int j = 0; j < list_data.Count; j++)
            {
                if (list_data[j].country_id == list_country[i].country_id)
                {
                    values.Add(new KeyValuePair<string, int>(list_data[j].time, list_data[j].bill)); //time + bill
                }
            }
            createLineSeries_Day(values, list_country[i].name);
            button_createExcel.IsEnabled = true;
        }
    }

    #region методы отрисовки данных графика
    public void createLineSeries_Day(List<KeyValuePair<string, int>> values, string title)
    {
        ColumnSeries ls = new ColumnSeries();
        ls.Title = title;
        ls.DependentValuePath = "Value";
        ls.IndependentValuePath = "Key";
        ls.IsSelectionEnabled = true;
        ls.ItemsSource = values;
        day_chart.Series.Add(ls);
    }

    public void createColSeries_Month(List<KeyValuePair<string, int>> values, string title)
    {

```

```

        ColumnSeries ls = new ColumnSeries();
        ls.Title = title;
        ls.DependentValuePath = "Value";
        ls.IndependentValuePath = "Key";
        ls.IsSelectionEnabled = true;
        ls.ItemsSource = values;
        month_chart.Series.Add(ls);
    }

public void createColSeries_Week(List<KeyValuePair<string, int>> values, string title)
{
    ColumnSeries ls = new ColumnSeries();
    ls.Title = title;
    ls.DependentValuePath = "Value";
    ls.IndependentValuePath = "Key";
    ls.IsSelectionEnabled = true;
    ls.ItemsSource = values;
    week_chart.Series.Add(ls);
}

public void createColSeries_Quarter(List<KeyValuePair<string, int>> values, string title)
{
    ColumnSeries ls = new ColumnSeries();
    ls.Title = title;
    ls.DependentValuePath = "Value";
    ls.IndependentValuePath = "Key";
    ls.IsSelectionEnabled = true;
    ls.ItemsSource = values;
    quater_chart.Series.Add(ls);
}

public void createPieSeries_Stat(List<KeyValuePair<string, int>> values)
{
    PieSeries ls = new PieSeries();
    ls.DependentValuePath = "Value";
    ls.IndependentValuePath = "Key";
    ls.IsSelectionEnabled = true;
    ls.ItemsSource = values;
    stat_chart.Series.Add(ls);
}
#endregion

private void menu_lvl2_month_Click(object sender, RoutedEventArgs e)
{
    grid_day.Visibility = System.Windows.Visibility.Collapsed;
    grid_month.Visibility = System.Windows.Visibility.Visible;
    grid_week.Visibility = System.Windows.Visibility.Collapsed;
    grid_quater.Visibility = System.Windows.Visibility.Collapsed;
    grid_stat.Visibility = System.Windows.Visibility.Collapsed;
}

/// <summary>
/// отчет за месяц
/// </summary>
private async void button_create_month_Click(object sender, RoutedEventArgs e)
{
    for (int i = 0; i < month_chart.Series.Count; i++)
    {
        month_chart.Series.Remove(month_chart.Series[i]);
    }
    values = null;
    List<Orders> list_orders = await oOrdersRepository.Select_All_Orders_Async();

```

```

List<Data> list_data = new List<Data>();
for (int i = 0; i < list_orders.Count; i++)
{
    var datetime = DateTime.Parse(list_orders[i].date_order);
    Data d = new Data();
    if (DateTime.Parse(list_orders[i].date_order).Month == DateTime.Now.Month)
    {
        List<Orders> list_carts = await oOrdersRepository.Select_Orders_Async("select bill,sum
(bill),country_id from Orders GROUP BY country_id "); //"select bill,country_id from Orders where orders_id=" +
list_orders[i].orders_id);
        d.time = datetime.Date.Day.ToString() + "/" + datetime.Month.ToString() + "/" +
datetime.Year.ToString();
        foreach (var c in list_carts) //график цен
        {
            d.bill = (int)c.bill;
            d.country_id = c.country_id;
            list_data.Add(d);
        }
    }
}
list_month_data = list_data;
List<Country> list_country = await oCountryRepository.Select_Country_Async("SELECT
DISTINCT Orders.country_id, Country.Name FROM Orders LEFT JOIN Country ON Orders.country_id =
Country.country_id;");
for (int i = 0; i < list_country.Count; i++)
{
    List<Orders> listIC_carts = await oOrdersRepository.Select_Orders_Async("select sum
(bill),country_id from Orders where orders_id=" + list_orders[i].orders_id + " GROUP BY country_id "); //"select
bill,country_id from Orders where orders_id=" + list_orders[i].orders_id);

    values = new List<KeyValuePair<string, int>>();
    for (int j = 0; j < list_data.Count; j++)
    {
        if (list_data[j].country_id == list_country[i].country_id)
            values.Add(new KeyValuePair<string, int>(list_data[j].time,
list_data[j].bill)); //listIC_carts[j].bill); //
    }
    createColSeries_Month(values, list_country[i].name);
}
button_create_month.IsEnabled = false;
button_excel_month.IsEnabled = true;
}

private void menu_lvl2_range_Click(object sender, RoutedEventArgs e)
{
    grid_week.Visibility = System.Windows.Visibility.Visible;
    grid_day.Visibility = System.Windows.Visibility.Collapsed;
    grid_month.Visibility = System.Windows.Visibility.Collapsed;
    grid_quater.Visibility = System.Windows.Visibility.Collapsed;
    grid_stat.Visibility = System.Windows.Visibility.Collapsed;
}

/// <summary>
/// отчет за неделю
/// </summary>
private async void button_create_week_Click(object sender, RoutedEventArgs e)
{
    for (int i = 0; i < month_chart.Series.Count; i++)
    {
        month_chart.Series.Remove(month_chart.Series[i]);
    }
}

```

```

values = null;
List<Orders> list_orders = await oOrdersRepository.Select_All_Orders_Async();
List<Data> list_data = new List<Data>();
var cal = System.Globalization.DateTimeFormatInfo.CurrentInfo.Calendar;
for (int i = 0; i < list_orders.Count; i++)
{
    var datetime = DateTime.Parse(list_orders[i].date_order);
    Data d = new Data();
    if (cal.GetWeekOfYear(datetime, System.Globalization.CalendarWeekRule.FirstDay,
System.DayOfWeek.Monday) == cal.GetWeekOfYear(DateTime.Now,
System.Globalization.CalendarWeekRule.FirstDay, System.DayOfWeek.Monday))
    {
        List<Orders> list_carts = await oOrdersRepository.Select_Orders_Async("select
bill,country_id from Orders where orders_id=" + list_orders[i].orders_id);
        d.time = datetime.Day.ToString() + "," + datetime.DayOfWeek.ToString();
        foreach (var c in list_carts)
        {
            d.bill = (int)c.bill;
            d.country_id = c.country_id;
            list_data.Add(d);
        }
    }
}
list_week_data = list_data;
List<Country> list_country = await oCountryRepository.Select_All_Country_Async();
for (int i = 0; i < list_country.Count; i++)
{
    values = new List<KeyValuePair<string, int>>();
    for (int j = 0; j < list_data.Count; j++)
    {
        if (list_data[j].country_id == list_country[i].country_id)
        {
            values.Add(new KeyValuePair<string, int>(list_data[j].time, list_data[j].bill));
        }
    }
    createColSeries_Week(values, list_country[i].name);
}
button_create_week.IsEnabled = false;
button_excel_week.IsEnabled = true;
}

private void menu_lvl2_quater_Click(object sender, RoutedEventArgs e)
{
    grid_quater.Visibility = System.Windows.Visibility.Visible;
    grid_day.Visibility = System.Windows.Visibility.Collapsed;
    grid_month.Visibility = System.Windows.Visibility.Collapsed;
    grid_week.Visibility = System.Windows.Visibility.Collapsed;
    grid_stat.Visibility = System.Windows.Visibility.Collapsed;
}

private async void button_create_quater_Click(object sender, RoutedEventArgs e)
{
    for (int i = 0; i < quater_chart.Series.Count; i++)
    {
        quater_chart.Series.Remove(quater_chart.Series[i]);
    }
    values = null;
    List<Orders> list_orders = await oOrdersRepository.Select_All_Orders_Async();
    List<Data> list_data = new List<Data>();
    for (int i = 0; i < list_orders.Count; i++)
    {
        var datetime = DateTime.Parse(list_orders[i].date_order);

```

```

        Data d = new Data();
        if ((datetime.Month > DateTime.Now.Month - 2) && (datetime.Month < DateTime.Now.Month
+ 2))
        {
            List<Orders> list_carts = await oOrdersRepository.Select_Orders_Async("select
bill,country_id from Orders where orders_id=" + list_orders[i].orders_id);
            d.time = datetime.Date.Day.ToString() + "/" + datetime.Month.ToString() + "/" +
datetime.Year.ToString();
            foreach (var c in list_carts)
            {
                d.bill = (int)c.bill;
                d.country_id = c.country_id;
                list_data.Add(d);
            }
        }
        list_quater_data = list_data;
        List<Country> list_country = await oCountryRepository.Select_All_Country_Async();
        for (int i = 0; i < list_country.Count; i++)
        {
            values = new List<KeyValuePair<string, int>>();
            for (int j = 0; j < list_data.Count; j++)
            {
                if (list_data[j].country_id == list_country[i].country_id)
                {
                    values.Add(new KeyValuePair<string, int>(list_data[j].time, list_data[j].bill));
                }
            }
            createColSeries_Quater(values, list_country[i].name);
        }
        button_create_quater.IsEnabled = false;
        button_excel_quater.IsEnabled = true;
    }

    private void menu_lv11_reply_Click(object sender, RoutedEventArgs e)
    {
        grid_stat.Visibility = System.Windows.Visibility.Visible;
        grid_day.Visibility = System.Windows.Visibility.Collapsed;
        grid_month.Visibility = System.Windows.Visibility.Collapsed;
        grid_week.Visibility = System.Windows.Visibility.Collapsed;
        grid_quater.Visibility = System.Windows.Visibility.Collapsed;
    }

    private async void button_create_stat_Click(object sender, RoutedEventArgs e)
    {
        for (int i = 0; i < stat_chart.Series.Count; i++)
        {
            stat_chart.Series.Remove(stat_chart.Series[i]);
        }
        values = null;
        list_pie = new List<PieData>();
        List<Orders> list_orders = await oOrdersRepository.Select_All_Orders_Async();
        for (int i = 0; i < list_orders.Count; i++)
        {
            PieData pd = new PieData();
            List<Orders> list_carts = await oOrdersRepository.Select_Orders_Async("select bill,country_id
from Orders where orders_id=" + list_orders[i].orders_id);
            foreach (var n in list_carts)
            {
                pd.bill = (int)n.bill;
                pd.country_id = n.country_id;
                Country b = await conn.GetAsync<Country>(n.country_id);
            }
        }
    }

```



```

        pd.name = b.name;
        list_pie.Add(pd);
    }
}

int min_id = 0;
int max = 0;
int min = 1000000;
int max_id = 0;
for (int i = 0; i < list_pie.Count; i++)
{
    if (list_pie[i].bill > max)
    {
        max = list_pie[i].bill;
        max_id = i;
    }
}
for (int i = 0; i < list_pie.Count; i++)
{
    if (list_pie[i].bill < min)
    {
        min = list_pie[i].bill;
        min_id = i;
    }
}

List<Country> list_bouq = await oCountryRepository.Select_All_Country_Async();

pie_values = new List<KeyValuePair<string, int>>();
string min_name = "";
string max_name = "";
for (int j = 0; j < list_pie.Count; j++)
{
    if (j == min_id || j == max_id)
    {
        pie_values.Add(new KeyValuePair<string, int>(list_pie[j].name, list_pie[j].bill));
        if (j == min_id)
        {
            min_name = list_pie[j].name;
        }
        if (j == max_id)
        {
            max_name = list_pie[j].name;
        }
    }
}
createPieSeries_Stat(pie_values);
button_create_stat.IsEnabled = false;
button_excel_stat.IsEnabled = true;
}

private void button_createExcel_Click(object sender, RoutedEventArgs e)
{
    r.Create_Excel_Doc(list_day_data, this, conn, list_day_data, list_week_data, list_month_data,
list_quater_data);
}

private void button_excel_month_Click(object sender, RoutedEventArgs e)
{
    r.Create_Excel_Doc(list_month_data, this, conn, list_day_data, list_week_data, list_month_data,
list_quater_data);
}

```

```

private void button_excel_week_Click(object sender, RoutedEventArgs e)
{
    r.Create_Excel_Doc(list_week_data, this, conn, list_day_data, list_week_data, list_month_data,
list_quater_data);
}

private void button_excel_quater_Click(object sender, RoutedEventArgs e)
{
    r.Create_Excel_Doc(list_quater_data, this, conn, list_day_data, list_week_data, list_month_data,
list_quater_data);
}

private void button_excel_stat_Click(object sender, RoutedEventArgs e)
{
    r.Create_Excel_Doc(pie_values);
}
#endregion

/// <summary>
/// Выделение данных клиента при нажатии ЛКМ и заполнение ими текстовых. Запоминает
выбранного клиента.
/// </summary>
private async void listview_Clients_Copy_MouseLeftButtonUp(object sender,
MouseButtonEventArgs e)
{
    try
    {
        selected_client = listview_Clients_Copy.SelectedItem as new_Clients;
        ClientsControl clientsControl = new ClientsControl();

        var fio = selected_client.lastname.Split(' ');
        edit_lastname.Text = fio[0].ToString();
        edit_name.Text = fio[1].ToString();
        edit_otch.Text = fio[2].ToString();
        edit_birthday.Text = selected_client.birthday;
        edit_phone.Text = selected_client.phone;
        edit_mail.Text = selected_client.email;
        edit_zagran.Text = selected_client.zagran;
        edit_zagranpass_date.Text = selected_client.zagran_date;
        edit_pass.Text = selected_client.ru_pass;
    }
    catch
    {
        System.Windows.Forms.MessageBox.Show("Клиент не выбран!");
    }
}

private void Add_Order_Click(object sender, RoutedEventArgs e)
{
}

/// <summary>
/// Выделение данных отеля при нажатии ЛКМ и заполнение ими текстовых. Запоминает
выбранного клиента.
/// </summary>
private void listview_Otels_MouseLeftButtonUp(object sender, MouseButtonEventArgs e)
{
    try
    {
        oOtels_country_resorts = (Otels_country_resorts)listview_Otels.SelectedItem;

```

```

combo_country.SelectedIndex = oOtel_country_resorts.country.country_id - 1;
combo_resort.SelectedIndex = oOtel_country_resorts.resorts.resorts_id - 1;
combo_otel.SelectedIndex = oOtel_country_resorts.otels.otels_id - 1;

foreach (var item in combo_eating.Items)
{
    if (item.ToString() == oOtel_country_resorts.otels.eating)
    {
        combo_eating.SelectedIndex = selected_otel.eating.IndexOf(item.ToString());
    }
}
Bill_box_Copy1.Text = oOtel_country_resorts.otels.bill;

selected_otel = listview_Otels.SelectedItem as Otel;

SolidColorBrush blue = new SolidColorBrush(Color.FromRgb(0, 155, 255));
SolidColorBrush clear = new SolidColorBrush(Color.FromRgb(255, 255, 255));

switch (oOtel_country_resorts.otels.stars)
{
    case 1:
    {
        star_raiting.s1.Fill = blue;
        star_raiting.s2.Fill = clear;
        star_raiting.s3.Fill = clear;
        star_raiting.s4.Fill = clear;
        star_raiting.s5.Fill = clear;
    }
    break;
    case 2:
    {
        star_raiting.s1.Fill = blue;
        star_raiting.s2.Fill = blue;
        star_raiting.s3.Fill = clear;
        star_raiting.s4.Fill = clear;
        star_raiting.s5.Fill = clear;
    }
    break;
    case 3:
    {
        star_raiting.s1.Fill = blue;
        star_raiting.s2.Fill = blue;
        star_raiting.s3.Fill = blue;
        star_raiting.s4.Fill = clear;
        star_raiting.s5.Fill = clear;
    }
    break;
    case 4:
    {
        star_raiting.s1.Fill = blue;
        star_raiting.s2.Fill = blue;
        star_raiting.s3.Fill = blue;
        star_raiting.s4.Fill = blue;
        star_raiting.s5.Fill = clear;
    }
    break;
    case 5:
    {
        star_raiting.s1.Fill = blue;
        star_raiting.s2.Fill = blue;
        star_raiting.s3.Fill = blue;
        star_raiting.s4.Fill = blue;
    }
}

```

```

        star_raiting.s5.Fill = blue;
    }
    break;
}
}
catch
{
    MessageBox.Show("Отель не выбран!");
}
}

/// <summary>
/// Выделение данных сотрудников при нажатии ЛКМ и заполнение ими текстовых.
Запоминает выбранного сотрудника.
/// </summary>
private void listview_Employees_MouseLeftButtonUp(object sender, MouseButtonEventArgs e)
{
    try
    {
        oEmployees_Login = (Employees_Login)listview_Employees.SelectedItem;

        var fio = oEmployees_Login.employee.lastname.Split(' ');

        lastname_emp.Text = fio[0].ToString();
        name_emp.Text = fio[1].ToString(); //selected_employee.name;
        middlename_emp.Text = fio[2].ToString(); //selected_employee.middlename;

        login_emp.Text = oEmployees_Login.log_in.login;
        password_emp.Text = oEmployees_Login.log_in.password;

        if (oEmployees_Login.employee.position == "Глав. менеджер" ||
oEmployees_Login.employee.position == "Директор")
        {
            main_manager.IsChecked = true; manager.IsChecked = false;
        }
        else { manager.IsChecked = true; main_manager.IsChecked = false; }
    }
    catch { MessageBox.Show("Сотрудник не выбран!"); }
}

/// <summary>
/// Добавление договора агент-оператор
/// </summary>
private void Add_agent_operator_Click(object sender, RoutedEventArgs e)
{
    Create_Order_Excel(pie_values);
}

/// <summary>
/// Добавление договора агент-турист
/// </summary>
private void Add_agent_turist_Click(object sender, RoutedEventArgs e)
{
    Create_Order_Excel(pie_values);
}

/// <summary>
/// Добавление туристической путевки
/// </summary>
private void Print_order_Click(object sender, RoutedEventArgs e)
{
    Create_Order_Excel(pie_values);
}

```

```

public void Create_Order_Excel(List<KeyValuePair<string, int>> pie_values)
{
    if (pie_values != null)
    {
        Excel.Application exapp = new Excel.Application();
        exapp.SheetsInNewWorkbook = 1;
        Excel.Workbook workbooks =
exapp.Workbooks.Open(@"C:\Users\13ann\OneDrive\Документы\Фриланс\Договор.xlsx");
        exapp.DisplayAlerts = true;
        exapp.Visible = true;
        Excel.Workbooks exappworkbooks = exapp.Workbooks;
        Excel.Workbook exappworkbook = exappworkbooks[1];
        exappworkbook.Saved = false;
        Excel.Sheets excelsheets = exappworkbook.Worksheets;
        Excel.Worksheet excellworksheet = (Excel.Worksheet)excelsheets.get_Item(1);
        excellworksheet.Activate();
        Excel.Range excelcells;

        selected_client = listview_Clients_Copy.SelectedItem as new_Clients;
        var selected_country = combo_To.SelectedItem as Country;
        var selected_resort = Resorts_combobox.SelectedItem as Resorts;
        var fio = selected_client.lastname.Split(' ');

        //ДАННЫЕ ЗАКАЗЧИКА
        exapp.Cells[10, 3] = fio[0].ToString(); //"Фамилия";
        exapp.Cells[11, 3] = fio[1].ToString(); //"Имя";
        exapp.Cells[12, 3] = fio[2].ToString(); //"Отчество";
        exapp.Cells[13, 3] = selected_country.name; //"Страна";
        exapp.Cells[14, 3] = selected_resort.name;
//Resorts_combobox.SelectedValue;/.Items.CurrentItem; (Resorts_combobox.SelectedItem); //"Курорт";
        exapp.Cells[15, 3] = dateTo.Text; //"Дата начала";
        exapp.Cells[16, 3] =
DateTime.Parse(dateTo.Text).AddDays(int.Parse(numUpDown_Nights.Text)); //"Дата окончания";
        exapp.Cells[17, 3] = Otels_combobox.SelectedItem; //"Отель";
        exapp.Cells[18, 3] = combo_From.SelectedItem; //"Класс отеля";
        exapp.Cells[19, 3] = combo_eating_Copy.SelectedItem; //"Питание";
        exapp.Cells[20, 3] = combo_From.SelectedItem; //"Кол-во туристов";
        exapp.Cells[21, 3] = numUpDown_Nights.Text; //"Кол-во ночей";
        exapp.Cells[22, 3] = numUpDown_count_people.Text; //"Стоимость";
        ////ДАННЫЕ ТУРИСТА
        exapp.Cells[5, 6] = selected_client.birthday; //"Дата рождения";
        exapp.Cells[6, 6] = selected_client.ru_pass; //"Паспорт";
        exapp.Cells[7, 6] = selected_client.adress; //"Адрес";
        exapp.Cells[8, 6] = selected_client.phone; //"Телефон";
        exapp.Cells[9, 6] = selected_client.adress; //"Адрес регистрации";
        ////ДАННЫЕ ТУРАГЕНТА «ТУРИЗМ ЕВРОПА»
        exapp.Cells[11, 6] = "Туризм Европа";
        exapp.Cells[12, 6] = "г. Тольятти Полякова 16"; //"Адрес";
        exapp.Cells[13, 6] = "125319"; //"Индекс";
        exapp.Cells[14, 6] = ""; //"ОГРН";
        exapp.Cells[15, 6] = "7826669006"; //"ИНН";
        exapp.Cells[16, 6] = "783901001"; //"КПП";
        exapp.Cells[17, 6] = "40702810600000030409"; //"р/с";
        exapp.Cells[18, 3] = "30101810100000000716"; //"к/с";
        exapp.Cells[19, 3] = "044525716"; //"БИК";    }
        else
        {
            MessageBox.Show("Недостаточно данных для отчета");
        }
    }
}
}
}

```