

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет
вычислительной техники

Кафедра
МО и ПЭВМ

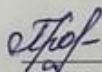
Направление подготовки 09.03.02 «Информационные системы и технологии»

БАКАЛАВРСКАЯ РАБОТА

на тему

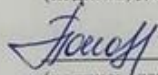
**Программное обеспечение автоматизации учета
гостиничного бизнеса в системе 1С**

Студент


(подпись, дата)

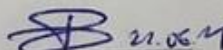
Прокофьева Алена Сергеевна
(ФИО полностью)

Руководитель


(подпись, дата)

Попова Н.А.
(фамилия, инициалы)

Нормоконтролёр


(подпись, дата)

Такташкин Д.В.
(фамилия, инициалы)

Работа допущена к защите (протокол заседания кафедры от _____ № _____)

Заведующий кафедрой


(подпись)

Макарычев П.П.
(фамилия, инициалы)

Работа защищена с отметкой 4 (протокол заседания ГЭК от 29.06.21 № 12)

Секретарь ГЭК


(подпись)

Попова Н.А.
(фамилия, инициалы)

Пенза, 2021

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

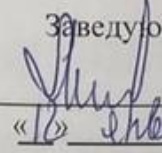
Вычислительной техники

Кафедра

МО и ПЭВМ

«Утверждаю»

Заведующий кафедрой

 П.П. Макарычев
«16» января 2021 г

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
БАКАЛАВРА

1. Студент Прокофьева Алена Сергеевна гр. 17ВИ1 факультета вычислительной техники
<Фамилия, имя, отчество полностью>

2. Тема работы «Программное обеспечение автоматизации учета гостиничного бизнеса
в системе 1С»

Тема утверждена приказом ПГУ № _____ от " ____ " _____ 20 ____

2. Руководитель работы Попова Н.А.

3. Задание на работу (назначение разработки, исходные данные и т.п.)
Назначение: автоматизировать деятельность гостиницы и упростить работу
администратора

Основные функции системы:

1. оформление документов гостиницы;
2. ведение баз клиентов и номеров;
3. формирование отчетов по деятельности отеля;

Технология разработки: RUP.

Инструментальные средства: Платформа «1С Предприятие»

Языковые средства: встроенные языковые средства 1С, UML

Среда разработки: ОС семейства Windows.

4. Перечень подлежащих разработке вопросов

1. Анализ предметной области и постановка задачи

2. Анализ требований на разработки

3. Разработка программных средств

4. Тестирование программных средств

5. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Анализ предметной области и требований	10%	14.04.2021 – 19.04.2021	<i>Гомофф</i>
2	Проектирование	25%	20.04.2021 – 01.05.2021	<i>Гомофф</i>
3	Реализация	35%	02.05.2021 – 17.05.2021	<i>Гомофф</i>
4	Контроль качества	15%	18.05.2021 – 14.06.2021	<i>Гомофф</i>
5	План разработки бюджета	5%	26.05.2021 – 02.06.2021	<i>Гомофф</i>
6	Оформление пояснительной записки	10%	04.06.2021 – 14.06.2021	<i>Гомофф</i>

Дата выдачи задания " 28 " декабря " 2020 "

Руководитель бакалаврской работы
(подпись, дата)

Гомофф Кошова Н.А.

Задание к исполнению принял студент
(подпись, дата)

Гомофф

Работу к защите допустить
Декан факультета
(подпись, дата)

Л.Р. Кошова

29.06.2021

Реферат

Пояснительная записка содержит 78 листов, 21 рисунок, 12 использованных источников, 24 таблицы и 3 приложения.

АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА, 1С:
ПРЕДПРИЯТИЕ, ГОСТИНИЦА, КОНФИГУРАЦИЯ, IDEF0, UML.

Объект исследования: автоматизация учета деятельности гостиницы.

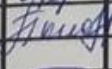
Предмет исследования: программное средство с возможностью учета деятельности гостиницы.

Цель работы: разработка информационной системы по автоматизации и учета гостиничного бизнеса. Создаваемая система должна упростить и ускорить работу администратора, вследствие чего уменьшить время ожидания клиента.

Технология разработки: для создания автоматизированной информационной системы используется Платформа «1С: Предприятие 8.3», для проектирования системы по автоматизации учета гостиничного бизнеса используются CASE-средства CA ERwin Process Modeler.

Результаты работы: спроектирована, разработана, отлажена и протестирована программа по автоматизации учета гостиничного бизнеса.

Область применения: все организации, занимающиеся гостиничным бизнесом.

					ПГУ 09.03.02-08БР151.14 ПЗ				
Изм.	Лист	№ докум.	Подпись	Дата	«Программное обеспечение автоматизации учета гостиничного бизнеса в системе 1С» Пояснительная записка	Лит.	Лист	Листов	
Разраб.		Прокофьева А.С.						4	78
Провер.		Попова Н.А.							
Реценз.									
И. Контр.		Такташкин Д.В.		22.02.14					
Утв.		Макарычев П.П.		28.02.14					
						Группа 17ВИ1			

Содержание

1 Анализ предметной области и постановка задачи.....	8
1.1 Анализ предметной области.....	8
1.2 Постановка задачи.....	8
1.3 Обзор исследуемых программ по автоматизации учета гостиничного бизнеса	9
1.3.1 Программа по автоматизации учета гостиничного бизнеса «HotelCloud».....	9
1.3.2 Программа по автоматизации учета гостиничного бизнеса «Вnovo».....	10
1.3.3 Программа по автоматизации учета гостиничного бизнеса «Контур.ОТЕЛЬ».....	11
1.3.4 Сравнительный анализ программ по автоматизации учета гостиничного бизнеса.....	13
2 Проектирование ИС гостиничного бизнеса.....	15
2.1 Функциональное моделирование разрабатываемого приложения	15
2.2 Технология разработки ПО	19
2.2.1 Технология разработки RUP	19
2.3 Средства разработки	20
2.4 Проектирование программных средств	20
2.4.1 Диаграмма классов	21
2.4.2 Диаграмма деятельности.....	28
2.4.3 Диаграмма состояния	29
2.4.4 Диаграмма коммуникации	30
2.4.5 Диаграмма последовательности	31
2.5 Планирование и расчет бюджета разрабатываемой системы	33
3 Реализация программного обеспечения.....	36
3.1 Диаграмма компонентов	36
3.2 Диаграмма классов	37
4 Контроль качества программного обеспечения	43
4.1 Функциональное тестирование	43
4.1.1 Определение классов эквивалентности.....	43
4.1.2 Определение граничных условий	44
4.1.3 Метод функциональных диаграмм.	46
4.1.4 Метод парного тестирования.	47

4.2 Структурное тестирование	49
4.2.1 Метод покрытия операторов.....	51
4.2.2 Метод покрытия условий	52
4.2.3 Метод покрытия маршрутов	53
Заключение	55
Приложение А. Глоссарий.....	57
Приложение Б. Листинг программы	59
Приложение В. Тестирование	74

Введение

В современном мире информационные технологии развиваются с бешеной скоростью. Особенно сильно изменения коснулись вопросов ведения предпринимательской деятельности. Информационные технологии позволяют автоматизировать работу предприятия. Исключением не стал и гостиничный бизнес.

Гостиничный бизнес связан с оказанием услуг в сфере гостеприимства и ориентирован главным образом на удовлетворение потребностей туристов, в том числе отдыхающих, прибывающих с деловыми целями и др.

Гостиничный бизнес очень разнообразен – он включает в себя как большие транснациональные сети, так и небольшие семейные гостиницы; как роскошные пятизвёздочные отели, похожие на сказочные дворцы, так и небольшие хостелы, позволяющие переночевать в чужом городе за минимальные деньги. Согласно статистике, на крупные отели приходится 60% постояльцев, а оставшиеся отдают предпочтение небольшим гостиницам [1].

На сегодняшний день существует множество требований к ведению работы отеля, что приводит к большим временным затратам на её оформление. Решением данной проблемы может стать автоматизация деятельности гостиницы.

1 Анализ предметной области и постановка задачи

1.1 Анализ предметной области

В настоящее время управление гостиницей невозможно без специальных программ учета и контроля. ПО для отеля включает в себя выполнение множества функций. Таких как ведение учета клиентов, номеров, установка скидок, оформление документов, создание отчетов.

Программное обеспечение для гостиницы должно работать быстро и просто, но при этом эффективно. Поэтому примитивные не специализированные программы для ведения учета являются не самым лучшим выбором. Программы для отелей при своей кажущейся простоте представляют собой универсальный сложный инструмент, незаменимый в управлении бизнесом.

Программное обеспечение для гостиниц предполагает, прежде всего, выполнение учетной функции. Система основана на ведении документов размещения гостей и их выбытия, информация по которым включает в себя дату прибытия и отбытия, размер скидки, стоимость проживания. Таким образом, программа регистрации в гостинице фиксирует все необходимые для дальнейшей работы данные.

Программа управления отелем включает в себя помимо учета документов размещения гостей и их выбытия, учет базы клиентов и номеров. Информационная система позволяет формировать различные отчеты необходимые в данной отрасли.

1.2 Постановка задачи

Целью данной выпускной квалификационной работы является разработка информационной системы по автоматизации учета гостиничного бизнеса. Создаваемая система должна упростить и ускорить работу администратора, вследствие чего уменьшить время ожидания клиента.

Для достижения указанной цели были поставлены следующие задачи:

- анализ предметной области и постановка задачи на разработку информационной системы по автоматизации учета гостиничного бизнеса;
- анализ существующих аналогов информационной системы;
- определение вариантов использования информационной системы;
- выполнение функционального моделирования системы;
- проектирование системы по автоматизации учета гостиничного бизнеса;
- реализация приложения по автоматизации учета гостиничного бизнеса;
- тестирование приложения по автоматизации учета гостиничного бизнеса.

1.3 Обзор исследуемых программ по автоматизации учета гостиничного бизнеса

В рамках данной выпускной квалификационной работы был проведен сравнительный анализ программ по автоматизации учета гостиничного бизнеса. Далее будут рассмотрены самые популярные из них: «HotelCloud», «Bnovo», «Контур.ОТЕЛЬ».

1.3.1 Программа по автоматизации учета гостиничного бизнеса «HotelCloud»

«HotelCloud» – программа для отелей, гостиниц, хостелов, пансионатов и других предприятий сферы обслуживания гостей. Имеет ряд особенностей:

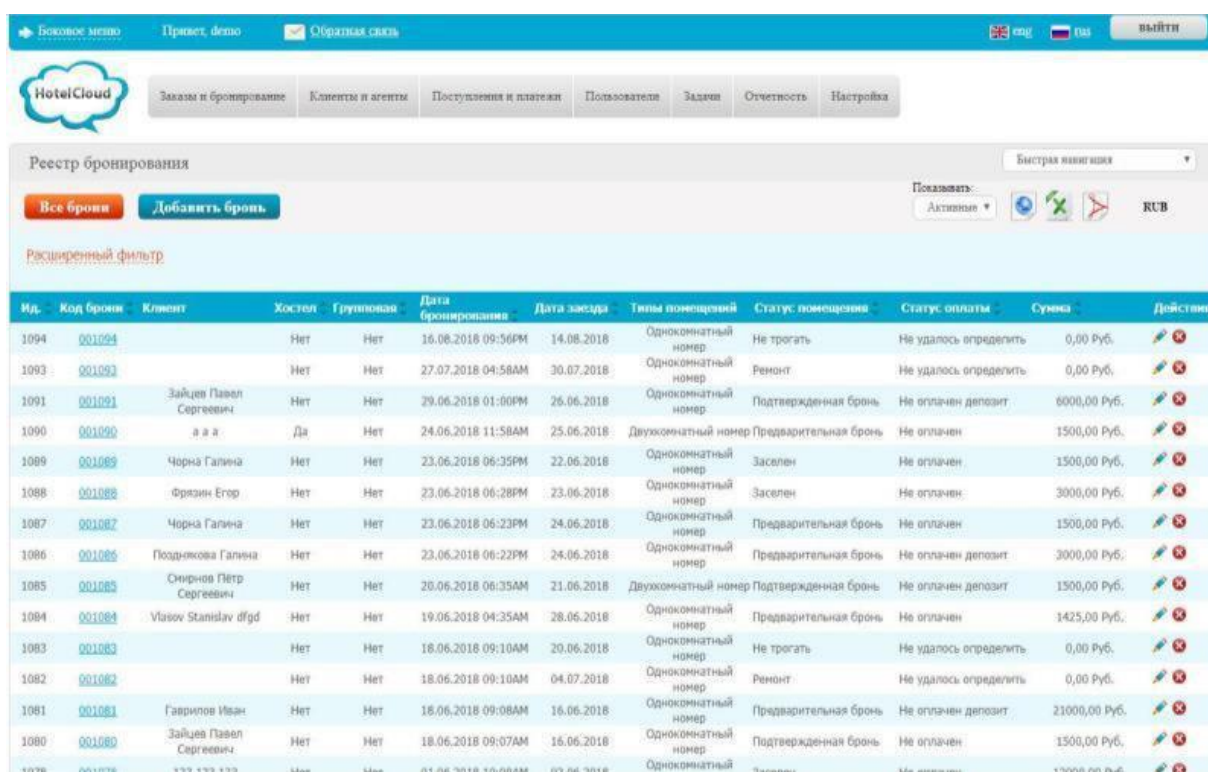
- 1) удобная таблица бронирования позволяет оперативно отслеживать состояние номерного фонда и управлять заселением/выселением гостей;
- 2) база данных заказов и информации о гостях позволяет эффективно управлять взаимоотношениями с клиентами и агентами;
- 3) система «HotelCloud» позволяет создавать помещения (номера, комнаты, апартаменты) различных типов, определять вместимость,

возможность хостельного (коечного размещения), наличие удобств, прикреплять фотографии помещений;

4) интерактивная таблица помещений позволит персоналу в оперативном режиме контролировать загрузку номерного фонда: наличие занятых и свободных номеров, ремонтируемых помещений;

5) в системе HotelCloud предусмотрена гибкая настройка тарифов на размещение гостей, а также прайс-листов на дополнительные услуги [2].

Интерфейс данной программы представлен на рисунке 1.



The screenshot shows the 'Реестр бронирования' (Reservation Register) interface of the HotelCloud application. It features a top navigation bar with a search menu, a demo login, and a settings link. Below the navigation bar is a main menu with options like 'Заказы и бронирования', 'Клиенты и агенты', 'Поступления и платежи', 'Пользователи', 'Задачи', 'Отчетность', and 'Настройка'. The main content area displays a table of reservations with columns for ID, Code, Client, Hostel, Group, Booking Date, Check-in Date, Room Type, Room Status, Payment Status, Sum, and Actions. The table lists various reservations, including those for 'Зайцев Павел Сергеевич', 'Чорна Галина', and 'Власов Станислав дфгд'.

Ид.	Код брони	Клиент	Хостел	Групповая	Дата бронирования	Дата заезда	Типы помещений	Статус помещения	Статус оплаты	Сумма	Действия
1094	001094		Нет	Нет	16.08.2018 09:56PM	14.08.2018	Одноместный номер	Не трогать	Не удалось определить	0,00 Руб.	
1093	001093		Нет	Нет	27.07.2018 04:58AM	30.07.2018	Одноместный номер	Ремонт	Не удалось определить	0,00 Руб.	
1091	001091	Зайцев Павел Сергеевич	Нет	Нет	29.06.2018 01:00PM	26.06.2018	Одноместный номер	Подтвержденная бронь	Не оплачен депозит	6000,00 Руб.	
1090	001090	а а а	Да	Нет	24.06.2018 11:58AM	25.06.2018	Двухместный номер	Предварительная бронь	Не оплачен	1500,00 Руб.	
1089	001089	Чорна Галина	Нет	Нет	23.06.2018 06:35PM	22.06.2018	Одноместный номер	Заселен	Не оплачен	1500,00 Руб.	
1088	001088	Брянин Егор	Нет	Нет	23.06.2018 06:28PM	23.06.2018	Одноместный номер	Заселен	Не оплачен	3000,00 Руб.	
1087	001087	Чорна Галина	Нет	Нет	23.06.2018 06:23PM	24.06.2018	Одноместный номер	Предварительная бронь	Не оплачен	1500,00 Руб.	
1086	001086	Позднякова Галина	Нет	Нет	23.06.2018 06:22PM	24.06.2018	Одноместный номер	Предварительная бронь	Не оплачен депозит	3000,00 Руб.	
1085	001085	Смирнов Петр Сергеевич	Нет	Нет	20.06.2018 06:35AM	21.06.2018	Двухместный номер	Подтвержденная бронь	Не оплачен депозит	1500,00 Руб.	
1084	001084	Vlasov Stanislav dfgd	Нет	Нет	19.06.2018 04:35AM	28.06.2018	Одноместный номер	Предварительная бронь	Не оплачен	1425,00 Руб.	
1083	001083		Нет	Нет	18.06.2018 09:10AM	20.06.2018	Одноместный номер	Не трогать	Не удалось определить	0,00 Руб.	
1082	001082		Нет	Нет	18.06.2018 09:10AM	04.07.2018	Одноместный номер	Ремонт	Не удалось определить	0,00 Руб.	
1081	001081	Гаврилов Иван	Нет	Нет	18.06.2018 09:08AM	16.06.2018	Одноместный номер	Предварительная бронь	Не оплачен депозит	21000,00 Руб.	
1080	001080	Зайцев Павел Сергеевич	Нет	Нет	18.06.2018 09:07AM	16.06.2018	Одноместный номер	Подтвержденная бронь	Не оплачен	1500,00 Руб.	
1078	001078	133 133 133	Мне	Мне	01.06.2018 10:06AM	03.06.2018	Одноместный номер	Заселен	Не оплачен	15000,00 Руб.	

Рисунок 1 – Интерфейс приложения «HotelCloud»

Здесь можно увидеть довольно дружелюбный интерфейс. Информация представлена в удобном, понятном виде.

1.3.2 Программа по автоматизации учета гостиничного бизнеса «Вново»

«Вново» – это модульная система управления и контроля для гостиничного бизнеса, предназначенная для управления продажами, повышения эффективности управленческих и сбытовых процессов в отелях, в хостелах, апартаментах и других объектах размещения.

Обладает следующими особенностями:

- 1) облачная система – при наличии интернета система доступна на любом устройстве: компьютер, смартфон, планшет;
- 2) единый интерфейс (личный кабинет) – легко стыкуется со всеми модулями «Вново» в единую программу, модули доступны в вашем личном кабинете;
- 3) безопасное хранение данных – в соответствии со стандартом безопасности данных индустрии платёжных карт PCI DSS, персональные данные гостей хранятся в соответствии с 152 ФЗ [3].

Интерфейс данной программы представлен на рисунке 2.

Код	Источник	Группа	Дата брони	Заезд	Выезд	Тариф	Категория	Гость	Телефон	Взр./дети	Баланс клиента	Итого
NJFFZ-141216	Прямой		14.12.2016	18.12.2016 14:00	21.12.2016 12:00	Стандартный	Стандарт	Безруков Иван	+7(991)554-77-55	2 / 0	0.00 Р	24 000
STP5T-141216	Прямой		14.12.2016	26.12.2016 14:00	27.12.2016 12:00	Стандартный	Люкс	Безруков Иван	+8_()_-	2 / 0	0.00 Р	5 000.0
9XD4N-141216	Прямой		14.12.2016	28.12.2016 14:00	30.12.2016 12:00	Стандартный	Бунгало	Безруков Иван	+8_()_-	4 / 0	0.00 Р	20 000
3WUR5-141216	Прямой		14.12.2016	04.01.2017 14:00	06.01.2017 12:00	Стандартный	Стандарт	Ворошилов Илья	+7 905 202 13 31	2 / 0	18 000.00 Р	18 000
CZJSV-141216	Прямой	Кира Нова	14.12.2016	06.01.2017 14:00	10.01.2017 12:00	Стандартный	Стандарт	Нова Кира	+81(153)244-664	2 / 0	0.00 Р	36 000
JTR99-141216	Прямой	Кира Нова	14.12.2016	06.01.2017 14:00	10.01.2017 12:00	Стандартный	Стандарт	Нова Кира	+81(153)244-664	2 / 0	0.00 Р	36 000
HCN4T-141216	Прямой	Кира Нова	14.12.2016	06.01.2017 14:00	10.01.2017 12:00	Стандартный	Стандарт	Нова Кира	+81(153)244-664	2 / 0	0.00 Р	36 000
CZ2NY-141216	Прямой	Кира Нова	14.12.2016	06.01.2017 14:00	10.01.2017 12:00	Стандартный	Стандарт	Нова Кира	+81(153)244-664	2 / 0	5 000.00 Р	36 000
FMRV3-141216	Прямой	Кира Нова	14.12.2016	06.01.2017 14:00	10.01.2017 12:00	Стандартный	Стандарт	Нова Кира	+81(153)244-664	2 / 0	0.00 Р	36 000

Рисунок 2 – Интерфейс приложения «Вново»

Данное приложение схоже с предыдущим. По сути является неким аналогом информационной системы «HotelCloud».

1.3.3 Программа по автоматизации учета гостиничного бизнеса «Контур.ОТЕЛЬ»

«Контур.ОТЕЛЬ» – Сервис позволяет управлять номерным фондом, получать брони с сайта отеля и площадок бронирования.

Данный сервис имеет следующие особенности:

- 1) все бронирования в одном окне – подключайте booking.com и другие площадки бронирования – и получайте новые брони прямо в шахматку. Отслеживайте заезды, выезды и оплаты;
- 2) настройка тарифов через календарь – настраивайте тарифы для номеров, добавляйте акции и спец условия на проживание, например, в праздники и выходные;
- 3) управление номерным фондом – добавляйте фото и описания, меняйте список дополнительных услуг, отмечайте готовность номеров к бронированию и проживанию;
- 4) расчет курортного сбора – рассчитывайте сумму курортного сбора и формируйте отчет за период [4].

Интерфейс данной программы представлен на рисунке 3.

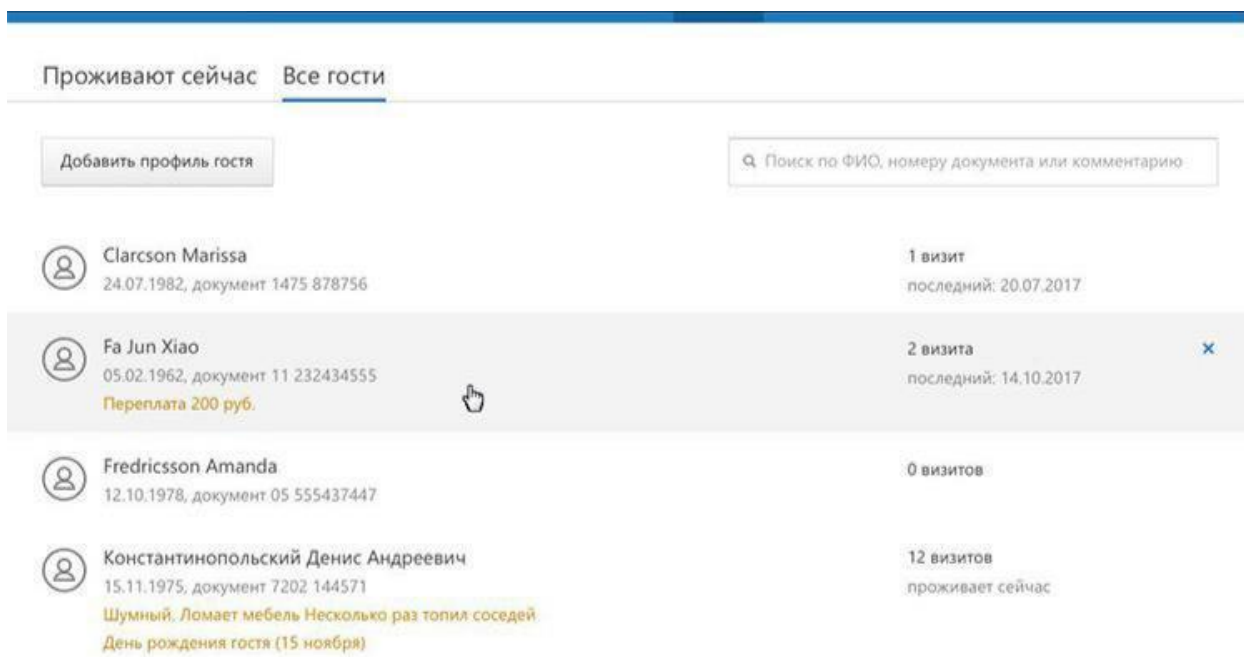


Рисунок 3 – Интерфейс приложения «Контур.ОТЕЛЬ»

В отличие от предшествующих приложений, не обладает интуитивно-понятным интерфейсом, но в свою очередь его стоимость значительно ниже.

1.3.4 Сравнительный анализ программ по автоматизации учета гостиничного бизнеса

Для сравнительного анализа программ по автоматизации учета гостиничного бизнеса были выделены такие критерии сравнения, как:

- 1) удобство интерфейса;
- 2) стоимость использования;
- 3) автоматическое обновление программы;
- 4) возможность изменения программы под потребности потребителя;
- 5) возможность добавления системы лояльности;
- 6) удобство использования для хостелов.

Результаты сравнительного анализа программ по автоматизации учета гостиничного бизнеса представлен в таблице 1.

Таблица 1 – Сравнительный анализ программ по автоматизации учета гостиничного бизнеса

Критерий	«HotelCloud»	«Вново»	«Контур.ОТЕЛЬ»
Удобство интерфейса	Интуитивно понятный интерфейс	Интуитивно понятный интерфейс	Сложный для восприятия интерфейс
Стоимость использования	От 75 рублей в месяц за номер	От 791 рубля в месяц	От 8100 рублей в год
Автоматическое обновление программы	Присутствует	Отсутствует	Присутствует
Возможность добавления системы лояльности	Присутствует	Присутствует	Присутствует
Удобство использования для хостелов	Неудобно оформлять заселение в непустой номер. Высока вероятность ошибок "человеческого фактора"	Среднее. Легко запутаться в форме заселения	Неудобно оформлять заселение в непустой номер. Высока вероятность ошибок "человеческого фактора"

Продолжение таблицы 1

Критерий	«HotelCloud»	«Вново»	«Контур.ОТЕЛЬ»
Возможность изменения программы под потребности предприятия	Отсутствует	Отсутствует	Отсутствует

В результате проведенного анализа можно прийти к выводу, что каждая из рассмотренных программ имеет свои недостатки и преимущества.

Главным минусом данного программного обеспечения является отсутствие возможности изменить программу самим пользователем. В свою очередь это приводит к тому, что многие отчеты приходится изменять вручную, а отсутствующие документы полностью создавать «с нуля».

Еще одним недостатком является не удобный механизм регистрации для хостелов.

В итоге можно сказать, что в ходе данной выпускной квалификационной работы было принято решение о разработке программы по автоматизации учета гостиничного бизнеса, в которой будет возможно изменить программу под потребности предприятия и будет разработан автоматизированный инструмент для регистрации постояльцев в хостеле.

2 Проектирование ИС гостиничного бизнеса

2.1 Функциональное моделирование разрабатываемого приложения

Для проектируемой конфигурации по автоматизации учета гостиничного бизнеса были выделены бизнес-процессы, которые представлены в виде диаграмм при помощи CASE-средства CA ERwin Process Modeler, поддерживающего такие методологии моделирования бизнес - процессов, как IDEF0, IDEF3 и DFD. CA ERwin Process Modeler является средством визуального моделирования.

Диаграммы IDEF0 позволяют понять, какие объекты или информация служат сырьем для процессов, какие результаты производят работы, что является управляющими факторами и какие ресурсы для этого необходимы. Процесс моделирования системы в IDEF0 начинается с определения контекста, т. е. наиболее абстрактного уровня описания системы в целом.

Контекстная диаграмма бизнес-процессов проектируемой системы в методологии IDEF0 представлена на рисунке 4.

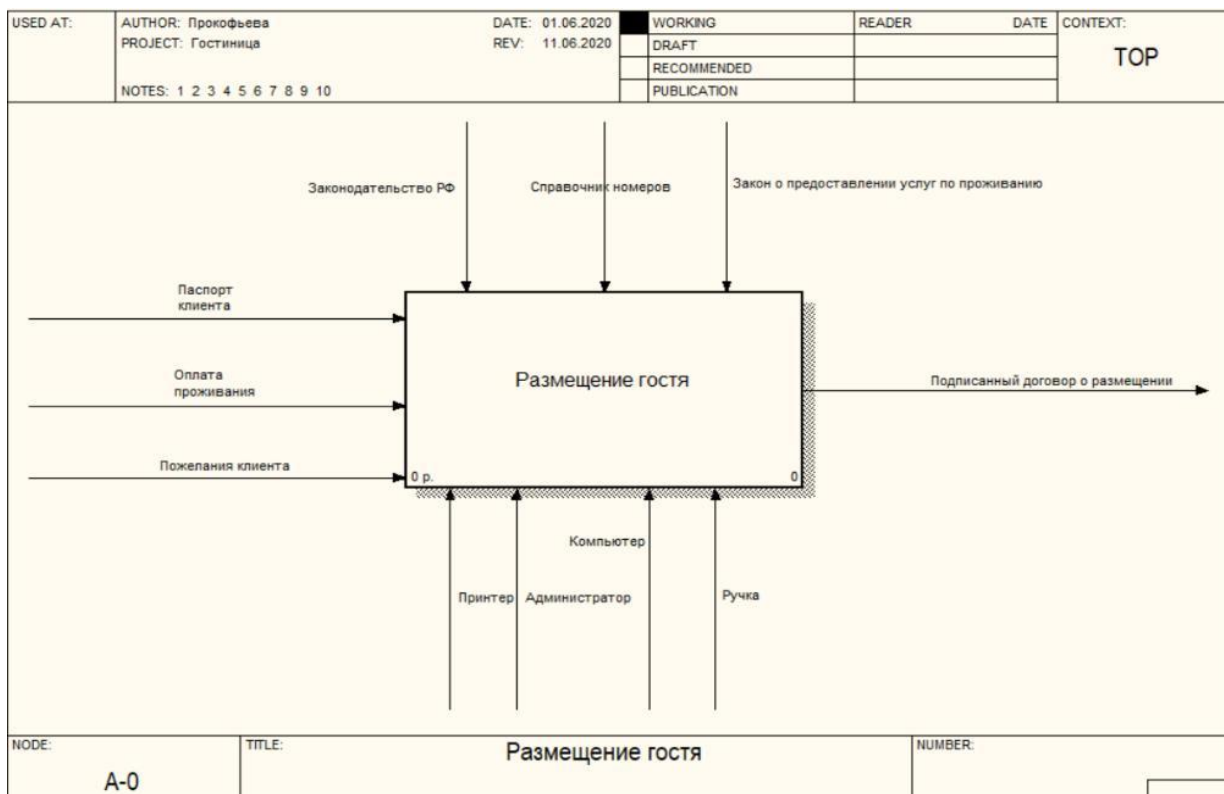


Рисунок 4 – Контекстная диаграмма конфигурации по автоматизации учета гостиничного бизнеса в методологии IDEF0

На контекстной диаграмме отражается общее описание основного бизнес-процесса (блока) системы – размещение клиента в гостинице.

После описания системы проводится ее разбиение на фрагменты. Этот процесс называется функциональной декомпозицией. Основной бизнес-процесс был декомпозирован на следующие составляющие бизнес-процессы: разработка клиентской части приложения, разработка интерфейса приложения, разработка серверной части приложения. Диаграмма декомпозиции представлена на рисунке 5.

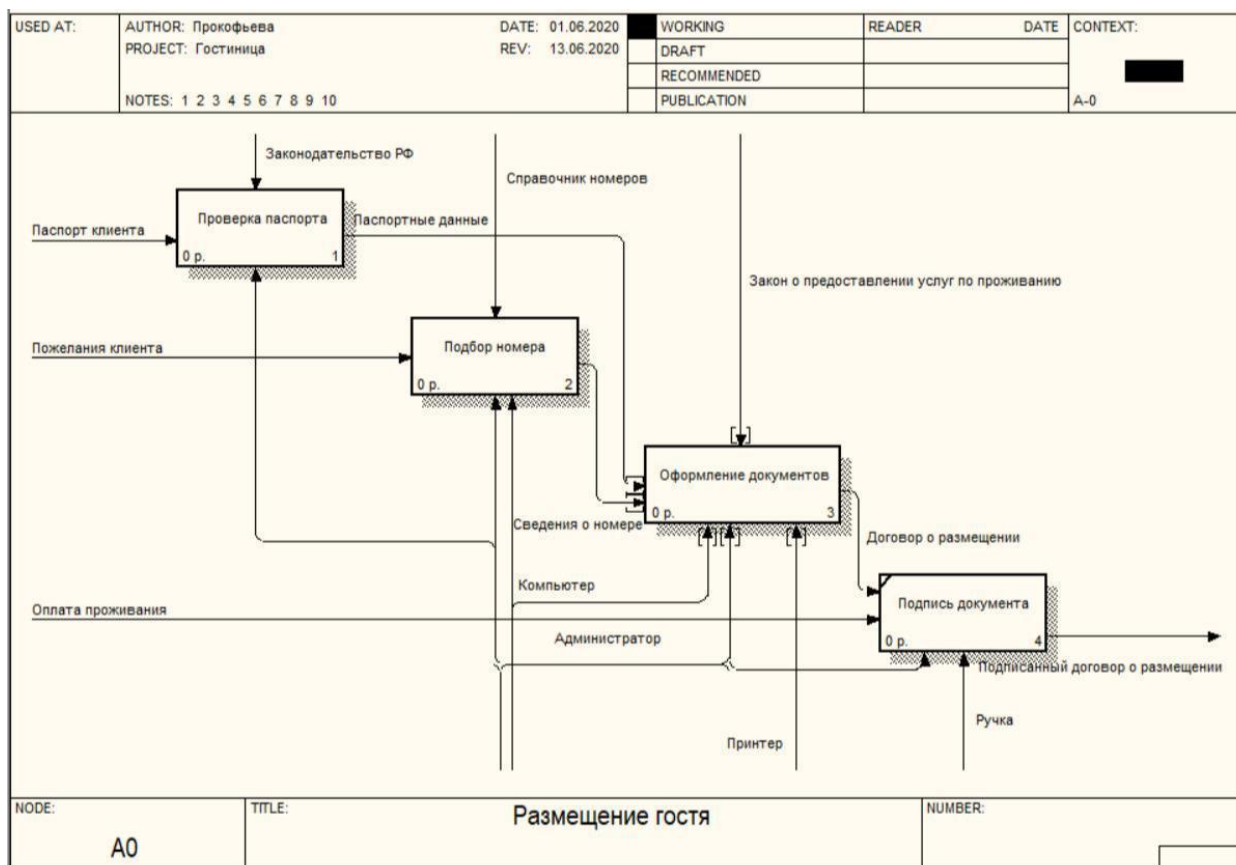


Рисунок 5 –Диаграмма декомпозиции конфигурации по автоматизации учета гостиничного бизнеса в нотации IDEF0

На первом этапе размещения гостя нужно проверить его паспорт. Второй этап заключается в подборе номера в соответствии с пожеланиями гостя. На третьем этапе оформляется договор о размещении. На четвертом – его распечатка.

Бизнес-процесс «Проверка паспорта» декомпозирован в нотации IDEF0 и представлен на рисунке 6.

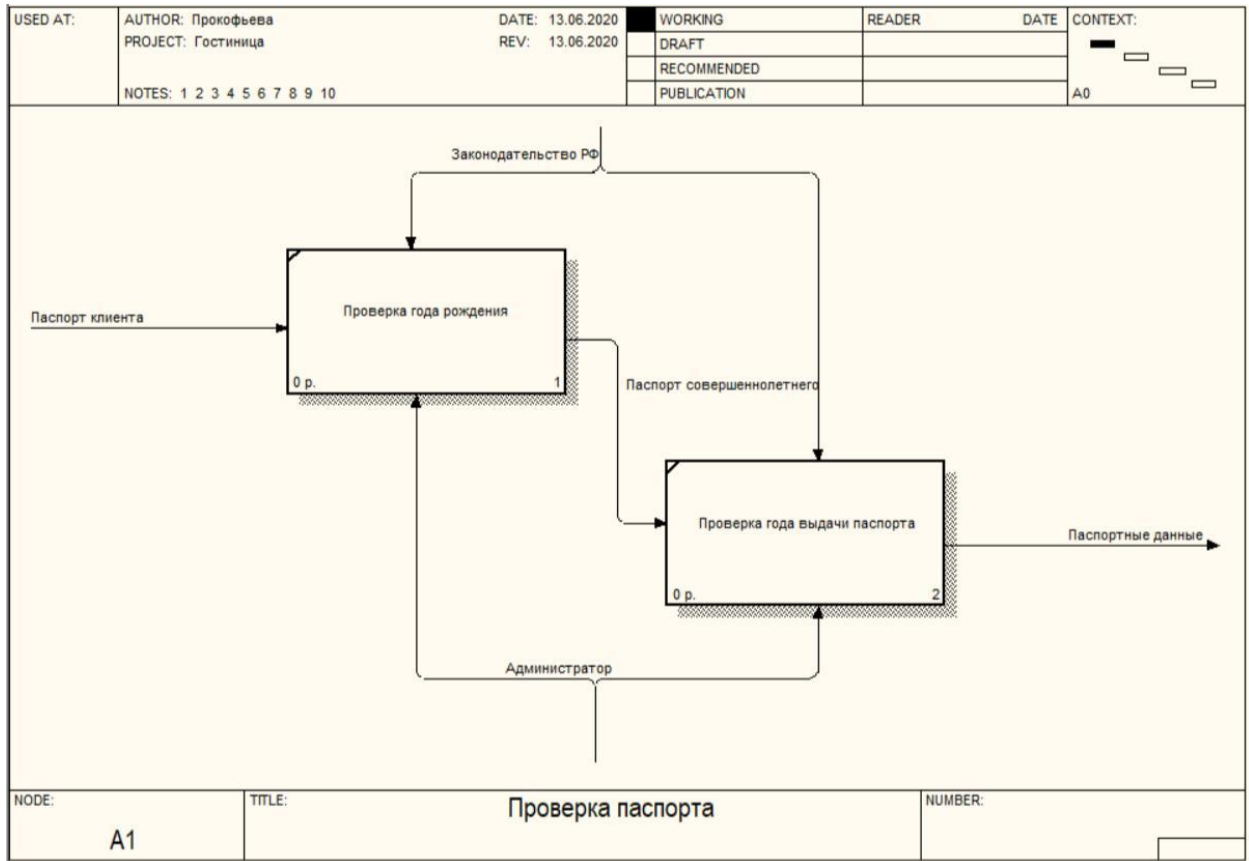


Рисунок 6 – Декомпозиция бизнес-процесса «Проверка паспорта» в нотации IDEF0

На данном этапе администратору необходимо удостовериться, что клиент является совершеннолетним и документ не просрочен.

Методология IDEF3 является одним из стандартов семейства IDEF и довольно широко используется при декомпозиции моделей IDEF0 для моделирования процессов более низкого уровня, поскольку с его помощью можно смоделировать технологические процессы, происходящие на предприятии, т.е. описать возможные сценарии реализации процессов, в рамках которых происходит последовательное изменение свойств объекта.

Бизнес-процесс «Подбор номера» декомпозирован в нотации IDEF3. IDEF3 является нотацией для описания логики взаимодействия информационных потоков. Декомпозиция бизнес-процесса «Подбор номера» представлена на рисунке 7.

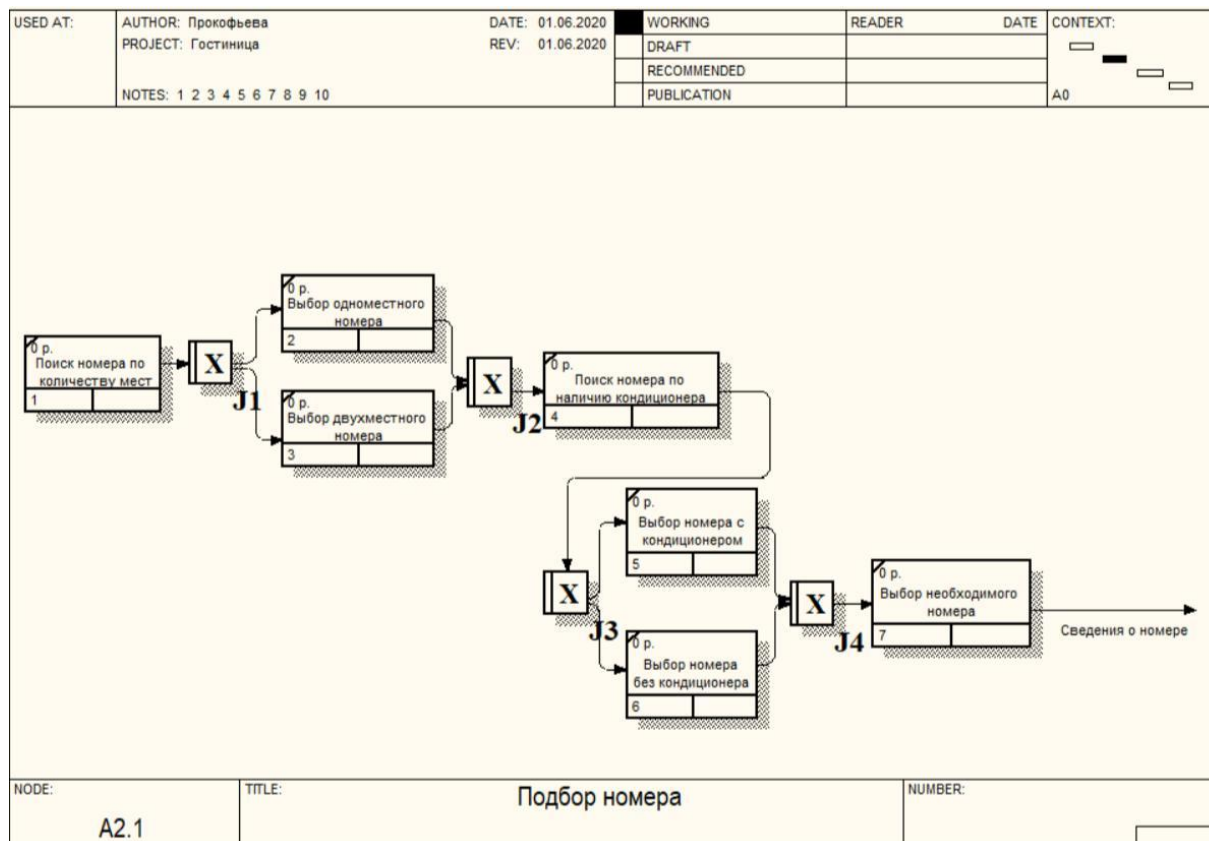


Рисунок 7 – Декомпозиция бизнес-процесса «Подбор номера» в нотации IDEF3

После проверки паспорта администратор начинает подбор номера, который удовлетворит все потребности клиента.

Основу методологии IDEF3 составляет графический язык описания процессов. Модель в нотации IDEF3 может содержать два типа диаграмм:

- диаграмму описания последовательности этапов процесса;
- диаграмму сети трансформаций состояний объекта.

Бизнес-процесс оформления документов декомпозирован в нотации DFD (нотации, предназначенной для моделирования информационных систем с точки зрения хранения, обработки и передачи данных). Диаграмма бизнес-процесса «Оформление документа» представлена на рисунке 8.

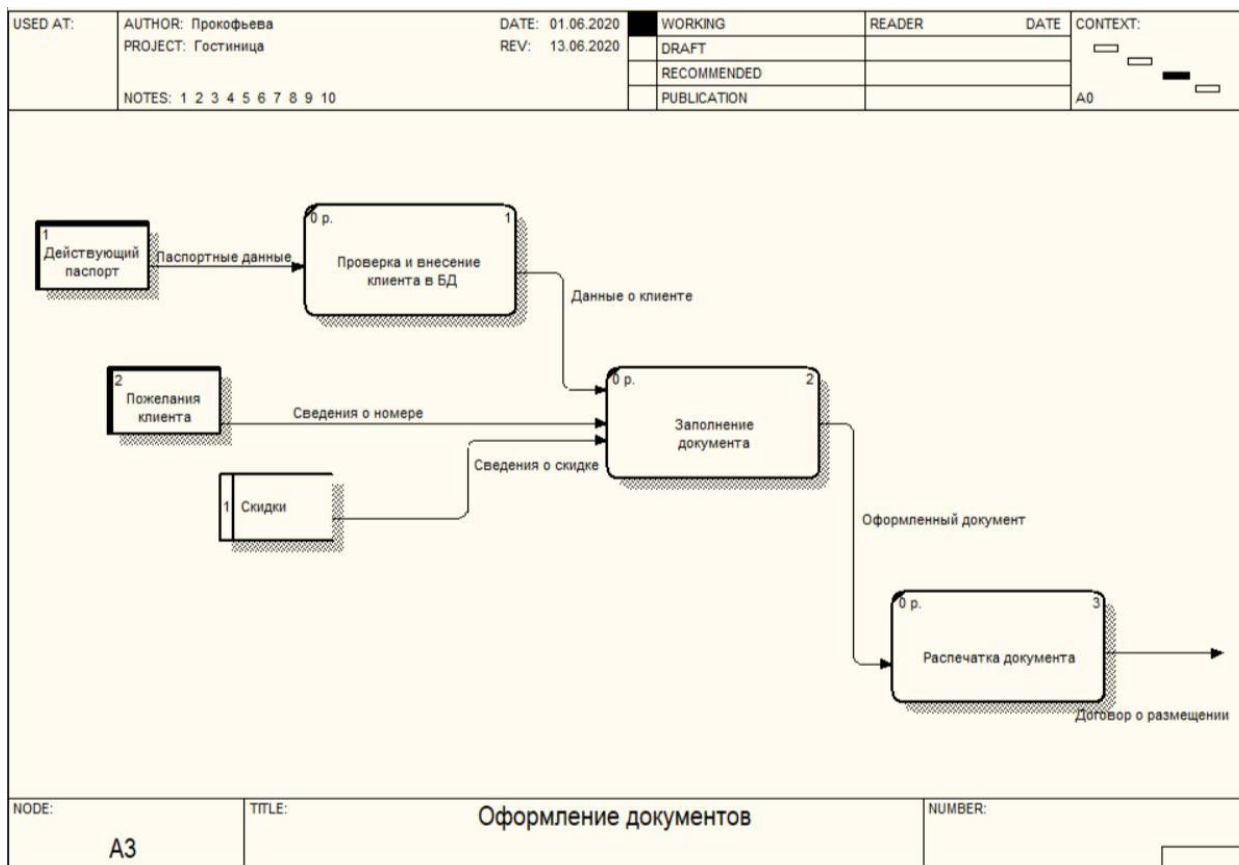


Рисунок 8 – Диаграмма бизнес-процесса «Оформление документов» в нотации DFD

После того как клиент определился с номером, администратор заполняет все поля в договоре о размещении и печатает получившийся документ.

2.2 Технология разработки ПО

Информационная система автоматизации гостиничного бизнеса будет создаваться с использованием методологии разработки RUP (Rational Unified Process).

2.2.1 Технология разработки RUP

Rational Unified Process (RUP) – методология разработки программного обеспечения, созданная компанией Rational Software.

RUP позволяет организовать строгое распределение задач. Данная методология помогает разработать программное обеспечение ровно в отведенное на это время.

RUP – это подход к тому, как эффективно работать с UML. Стандартный язык моделирования делает понятными описания требований, проектирование архитектуры системы [2].

В UML предусмотрено множество различных диаграмм. Они наглядно представляют процессы проектируемой программы. Из-за этого данная технология была выбрана для информационной системы автоматизации гостиничного бизнеса. Были разработаны следующие диаграммы:

- диаграмма деятельности;
- диаграмма классов;
- диаграмма состояния;
- диаграмма последовательности;
- диаграмма коммуникации.

Благодаря диаграммам можно значительно упростить решение сложных задач в программировании.

2.3 Средства разработки

Для разработки информационной системы автоматизации гостиничного бизнеса была выбрана технологическая платформа «1С Предприятие».

Она использует объекты (данных и метаданных) и механизмы управления объектами. Объекты (данные и метаданные) представляются в виде конфигураций. При автоматизации какой-либо деятельности составляется своя конфигурация объектов, которая и представляет собой законченное прикладное решение. Конфигурация разрабатывается в специальном режиме работы программы «Конфигуратор», далее инициализируется режим работы «1С: Предприятие», в нем пользователь получает доступ к созданной информационной системе [4].

2.4 Проектирование программных средств

Проектирование программных средств выполнено в соответствии с методологией RUP приведенной выше. Проектирование было произведено с использованием диаграмм UML.

2.4.1 Диаграмма классов

Диаграмма классов описывает типы классов системы и разнообразные связи, которые имеются между ними. Кроме того, в диаграммах классов представляются атрибуты и методы классов. Атрибуты представляют свойства объектов класса. За именованнием атрибута может следовать его тип. Метод представляет функцию класса. Метод способен обладать параметрами и возвращать значения.

На основе анализа предметной области была построена диаграмма программных классов, представленная на рисунке 9.

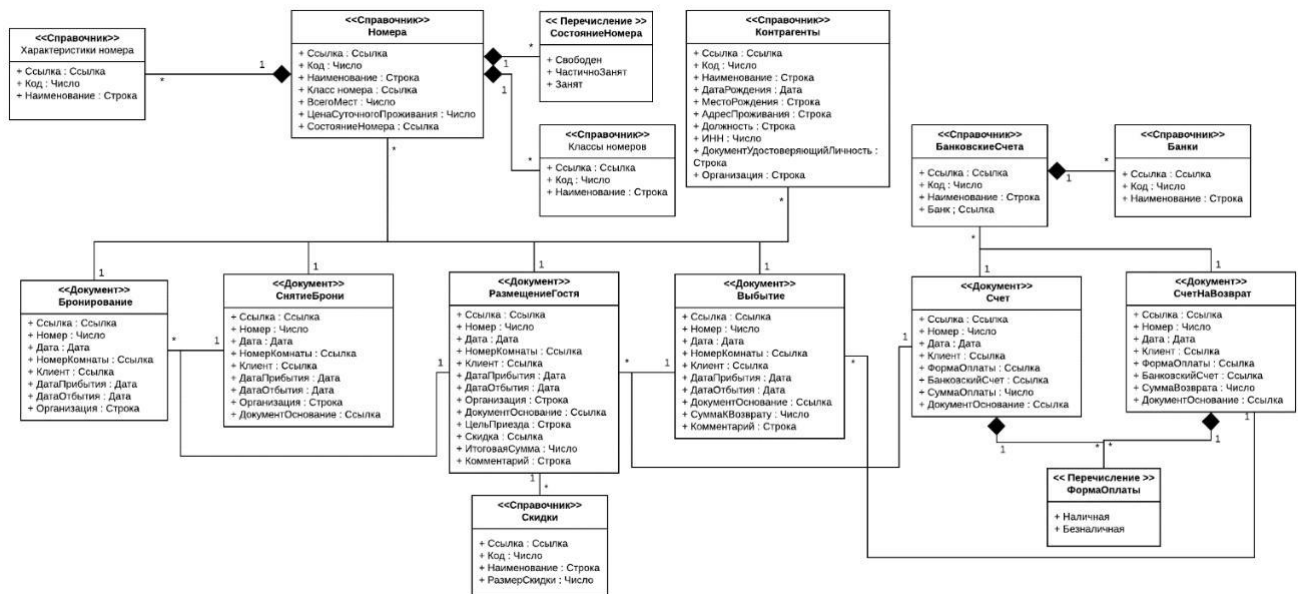


Рисунок 9 – Диаграмма классов конфигурации по автоматизации учета гостиничного бизнеса

Классами данной программы выступают различные объекты конфигурации. Они подразделяются на несколько типов: справочники, документы и перечисления.

Класс «ХарактеристикиНомера», представленный в таблице 2, является справочником, который хранит данные о различных удобствах,

соответствующих категории номера.

Таблица 2 – Класс «ХарактеристикиНомера»

Назначение:	Описывает номера.
Атрибуты:	Код: Число - код элемента справочника Наименование: Строка - наименование элемента справочника
Связь:	Ссылка: Ссылка - ссылка на элемент справочника, связь с классом «Номера»

Класс «КлассыНомеров», представленный в таблице 3, является справочником, который хранит данные о категориях номеров.

Таблица 3 – Класс «КлассыНомеров»

Назначение:	Описывает номера.
Атрибуты:	Код: Число - код элемента справочника Наименование: Строка - наименование элемента справочника
Связь:	Ссылка: Ссылка - ссылка на элемент справочника, связь с классом «Номера»

Класс «Банки», представленный в таблице 4, является справочником, который содержит информацию о банках.

Таблица 4 – Класс «Банки»

Назначение:	Описывает банки, в которых обслуживаются контрагенты.
Атрибуты:	Код: Число - код элемента справочника Наименование: Строка - наименование элемента справочника
Связь:	Ссылка: Ссылка - ссылка на элемент справочника, связь с классом «БанковскиеСчета»

Класс «БанковскиеСчета», представленный в таблице 5, представляет собой справочник, для хранения данных о банковских счетах клиентов.

Таблица 5 – Класс «Банковские счета»

Назначение:	Описывает банковские счета контрагентов.
Атрибуты:	Код: Число - код элемента справочника Наименование: Строка - наименование элемента справочника
Связь:	Ссылка: Ссылка - ссылка на элемент справочника, связь с классом «Счет» Банк: Ссылка - ссылка на элемент справочника «Банки», связь с классом «Банк»

Класс «Скидки», представленный в таблице 6, является справочником, который хранит информацию о доступных скидках.

Таблица 6 – Класс «Скидки»

Назначение:	Описывает скидки, предоставляемые контрагентам.
Атрибуты:	Код: Число - код элемента справочника Наименование: Строка - наименование элемента справочника РазмерСкидки: Число – размер скидки, предоставляемой клиенту
Связь:	Ссылка: Ссылка - ссылка на элемент справочника, связь с классом РазмещениеГостя»

Класс «Номера», представленный в таблице 7, является справочником, который хранит информацию о всех номерах гостиницы.

Таблица 7 – Класс «Номера»

Назначение:	Описывает номера гостиницы.
Атрибуты:	Код: Число - код элемента справочника Наименование: Строка - наименование элемента справочника РазмерСкидки: Число – размер скидки, предоставляемой клиенту ЦенаСуточногоПроживания: Число - цена суточного проживания в номере ВсегоМест: Число - количество свободных мест в номере

Продолжение таблицы 7

Назначение:	Описывает номера гостиницы.
Связь:	<p>Ссылка: Ссылка - ссылка на элемент справочника, связь с классом «РазмещениеГостя»</p> <p>КлассНомера: Ссылка - ссылка на элемент справочника «КлассыНомеров», связь с классом «КлассыНомеров»</p> <p>СостояниеНомера: Ссылка - ссылка на элемент перечисления «СостояниеНомера», связь с классом «СостояниеНомера»</p>

Класс «Контрагенты», представленный в таблице 8, является справочником, который хранит информацию о клиентах.

Таблица 8 – Класс «Контрагенты»

Назначение:	Описывает контрагентов.
Атрибуты:	<p>Код: Число - код элемента справочника</p> <p>Наименование: Строка - наименование элемента</p> <p>Справочника РазмерСкидки: Число – размер скидки, предоставляемой клиенту</p> <p>ДатаРождения: Дата - дата рождения контрагента</p> <p>МестоРождения: Строка - место рождения контрагента</p> <p>АдресПроживания: Строка – адрес проживания контрагента</p> <p>Должность: Строка - должность контрагента</p> <p>ИНН: Число - ИНН контрагента</p> <p>ДокументУдостоверяющийЛичность: Строка - документ, удостоверяющий личность, по которому регистрируется контрагент</p> <p>Организация: Строка - организация, к которой относится контрагент</p>
Связь:	<p>Ссылка: Ссылка - ссылка на элемент справочника, связь с классами «РазмещениеГостя», «Бронирование», «СнятиеБрони», «Выбытие»</p>

Класс «Бронирование», представленный в таблице 9, предназначен для обеспечения возможности бронирования номеров.

Таблица 9 – Класс «Бронирование»

Назначение:	Описывает документ, регистрирующий бронь клиента.
Атрибуты:	Номер: Число - номер документа. Дата: Дата - дата документа. ДатаПрибытия: Дата - дата прибытия клиента. ДатаОтбытия: Дата - дата отбытия клиента. Организация: Строка - организация, к которой относится контрагент.
Связь:	Ссылка: Ссылка - ссылка на элемент справочника, связь с классом «РазмещениеГостя». НомерКомнаты: Ссылка - ссылка на элемент справочника «НомерКомнаты», связь с классом «Номер комнаты». Клиент: Ссылка - ссылка на элемент справочника «Контрагенты», связь с классом «Контрагенты»

Класс «СнятиеБрони», представленный в таблице 10, является документом, который обеспечивает механизм снятия номера с резерва.

Таблица 10 – Класс «СнятиеБрони»

Назначение:	Описывает документ, регистрирующий снятие бронирования клиента.
Атрибуты:	Номер: Число - номер документа. Дата: Дата - дата документа. ДатаПрибытия: Дата - дата прибытия клиента. ДатаОтбытия: Дата - дата отбытия клиента. Организация: Строка - организация, к которой относится контрагент.
Связь:	Ссылка: Ссылка - ссылка на элемент справочника. НомерКомнаты: Ссылка - ссылка на элемент справочника «НомерКомнаты», связь с классом Клиент: Ссылка - ссылка на элемент справочника «Контрагенты», связь с классом «Контрагенты». ДокументОснование: Ссылка - ссылка на документ «Бронирование», связь с классом «Бронирование».

Класс «РазмещениеГостя», представленный в таблице 11, является документом, служащим для отражения факта заселения клиента.

Таблица 11 – Класс «РазмещениеГостя»

Назначение:	Описывает документ, регистрирующий заселение клиента.
Атрибуты:	<p>Номер: Число - номер документа.</p> <p>Дата: Дата - дата документа.</p> <p>ДатаПрибытия: Дата - дата прибытия клиента.</p> <p>ДатаОтбытия: Дата - дата отбытия клиента.</p> <p>Организация: Строка - организация, к которой относится контрагент.</p>
Связь:	<p>Ссылка: Ссылка - ссылка на элемент справочника, связь с классом «Счет».</p> <p>НомерКомнаты: Ссылка - ссылка на элемент справочника «НомерКомнаты», связь с классом «Номер комнаты».</p> <p>Клиент: Ссылка - ссылка на элемент справочника «Контрагенты», связь с классом «Контрагенты».</p> <p>ДокументОснование: Ссылка - ссылка на документ «Бронирование», связь с классом «Бронирование».</p>

Класс «Выбытие», представленный в таблице 12, является документом, который отражает факт выселения клиента.

Таблица 12 – Класс «Выбытие»

Назначение:	Описывает документ, регистрирующий выселение клиента.
Атрибуты:	<p>Номер: Число - номер документа.</p> <p>Дата: Дата - дата документа.</p> <p>ДатаПрибытия: Дата - дата прибытия клиента.</p> <p>ДатаОтбытия: Дата - дата отбытия клиента.</p>
Связь:	<p>Ссылка: Ссылка - ссылка на элемент справочника, связь с классом «СчетНаВозврат».</p> <p>НомерКомнаты: Ссылка - ссылка на элемент справочника «НомерКомнаты», связь с классом «Номер комнаты».</p> <p>Клиент: Ссылка - ссылка на элемент справочника «Контрагенты», связь с классом «Контрагенты».</p> <p>ДокументОснование: Ссылка - ссылка на документ «РазмещениеГостя», связь с классом «РазмещениеГостя».</p>

Класс «Счет», представленный в таблице 13, является документом, отражающим факт оплаты клиента.

Таблица 13 – Класс «Счет»

Назначение:	Описывает документ, регистрирующий счет клиента.
Атрибуты:	Номер: Число - номер документа. Дата: Дата - дата документа.
Связь:	Ссылка: Ссылка - ссылка на элемент справочника. Клиент: Ссылка - ссылка на элемент справочника «Контрагенты», связь с классом «Контрагенты». ДокументОснование: Ссылка - ссылка на документ «РазмещениеГостя», связь с классом «РазмещениеГостя». ФормаОплаты: Ссылка - ссылка на перечисление «ФормаОплаты», связь с классом «ФормаОплаты». БанковскийСчет: Ссылка - ссылка на элемент справочника «БанковскиеСчета», связь с классом «БанковскиеСчета».

Класс «СчетНаВозврат», представленный в таблице 14, является документом, который отражает факт выдачи клиенту денежных средств.

Таблица 14 – Класс «СчетНаВозврат»

Назначение:	Описывает документ, регистрирующий возврат денежных средств клиенту.
Атрибуты:	Номер: Число - номер документа. Дата: Дата - дата документа. СуммаНаВозврат: Число - сумма, возвращаемая клиенту.
Связь:	Ссылка: Ссылка - ссылка на элемент справочника. Клиент: Ссылка - ссылка на элемент справочника «Контрагенты», связь с классом «Контрагенты». ДокументОснование: Ссылка - ссылка на документ «Выбытие», связь с классом «Выбытие». ФормаОплаты: Ссылка - ссылка на перечисление «ФормаОплаты», связь с классом «ФормаОплаты». БанковскийСчет: Ссылка - ссылка на элемент справочника «БанковскиеСчета», связь с классом «БанковскиеСчета».

Данные таблицы наглядно показывают все атрибуты и связи между объектами конфигурации.

2.4.2 Диаграмма деятельности

С целью визуализации прецедента «Размещение гостя» была разработана диаграмма деятельности, представленная на рисунке 10.

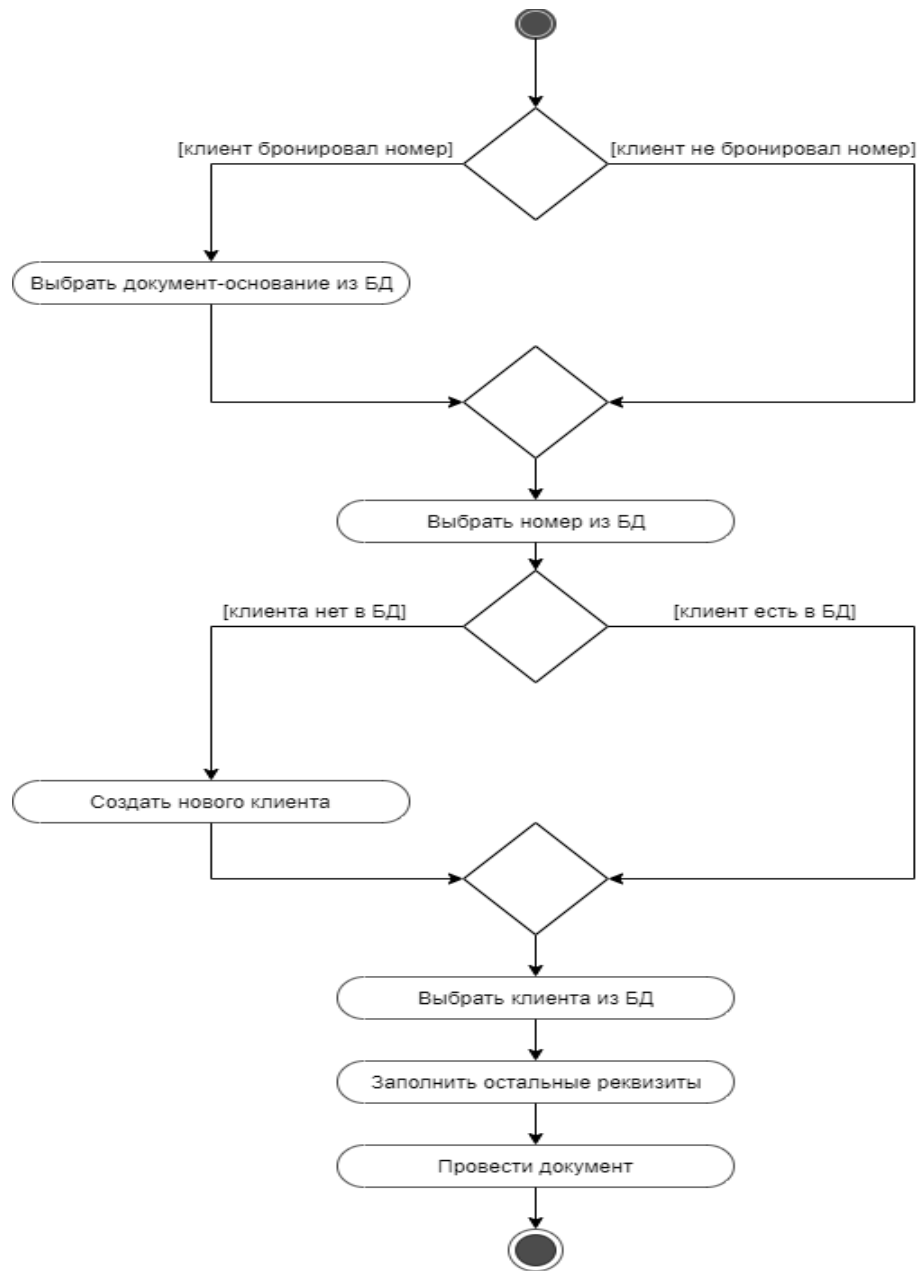


Рисунок 10 – Диаграмма деятельности «Размещение гостя»

Рассмотрим диаграмму деятельности прецедента «Размещение гостя». В начале создания нового размещения может произойти два сценария:

- 1) клиент бронировал номер, так что нужно выбрать документ-основание из БД;
- 2) клиент не бронировал номер, по этому поле «Документ-

основание» останется пустым.

После прохода любого из сценариев пользователь выбирает номер комнаты из БД. После этого происходит один из следующих сценариев:

- 1) если клиент никогда не пользовался услугами данной гостиницы, то его данные нужно занести в БД;
- 2) если клиент уже не первый раз обращается в данную гостиницу, то его нужно выбрать из БД.

После прохода любого из сценариев пользователь выбирает клиента из БД. Затем заполняет оставшиеся реквизиты и проводит документ.

2.4.3 Диаграмма состояния

С целью визуализации прецедента «Добавление нового счета» была разработана диаграмма состояния, представленная на рисунке 11.



Рисунок 11 – Диаграмма состояния прецедента «Добавление нового счета»

Рассмотрим диаграмму состояния прецедента «Добавление нового счета». Она показывает 3 возможных состояния:

- 1) документ «Счет» заполнен – переход к следующему состоянию возможен, только если добавлены все элементы документа, также из этого состояния можно вернуться в него же, если заполнены не все элементы и к закрытию формы, если пользователь решил отменить операцию;

2) документ «Счет» распечатан – переход к следующему состоянию производится после распечатки документа, которая в свою очередь инициируется нажатием на кнопку «Печать»;

3) документ «Счет» записан в системе – переход к закрытию форму реализуется посредством нажатия на кнопку «Провести и закрыть».

2.4.4 Диаграмма коммуникации

С целью визуализации прецедента «Счет на возврат» разработана диаграмма коммуникации, представленная на рисунке 12.



Рисунок 12 – Диаграмма коммуникации прецедента «Счет на возврат»

Рассмотрим диаграмму коммуникации прецедента «Формирование счета на возврат». На ней отражена последовательность 12 действий для создания отчета:

выбрать документ-основание из БД – пользователь выбирает документ «Выбытие» из БД;

1) обратиться к БД – система запрашивает данные, выбранные пользователем;

2) передать результаты поиска – посылаются необходимые данные;

3) добавить документ-основание на форму – система отображает выбранные данные на форме;

4) выбрать клиента из БД – пользователь выбирает нужного клиента из списка элементов справочника;

- 5) обратиться к БД – система запрашивает данные, выбранные пользователем;
- 6) передать результаты поиска – посылаются необходимые данные;
- 7) добавить клиента на форму – система отображает выбранные данные на форме;
- 8) нажать кнопку «Провести» – пользователь запускает процесс проводки документа;
- 9) добавить документ в БД – данные о документе добавляются в БД;
- 10) отобразить новый документ в системе – в списке документов «Счета на возврат» появляется новый документ;
- 11) вывести сообщение о добавлении нового документа – в правом нижнем углу формы всплывает сообщение об успешном добавлении нового документа.

2.4.5 Диаграмма последовательности

Диаграмма последовательности – UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актеров (действующих лиц) информационной системы в рамках прецедента.

При построении диаграммы последовательности обязанности распределяются между объектами и классами. Данные диаграммы удобно использовать не только при проектировании архитектуры приложений, но и для описания взаимодействия процессов или потоков.

С целью визуализации прецедента «Создание нового бронирования» разработана диаграмма последовательности прецедента, представленная на рисунке 13.

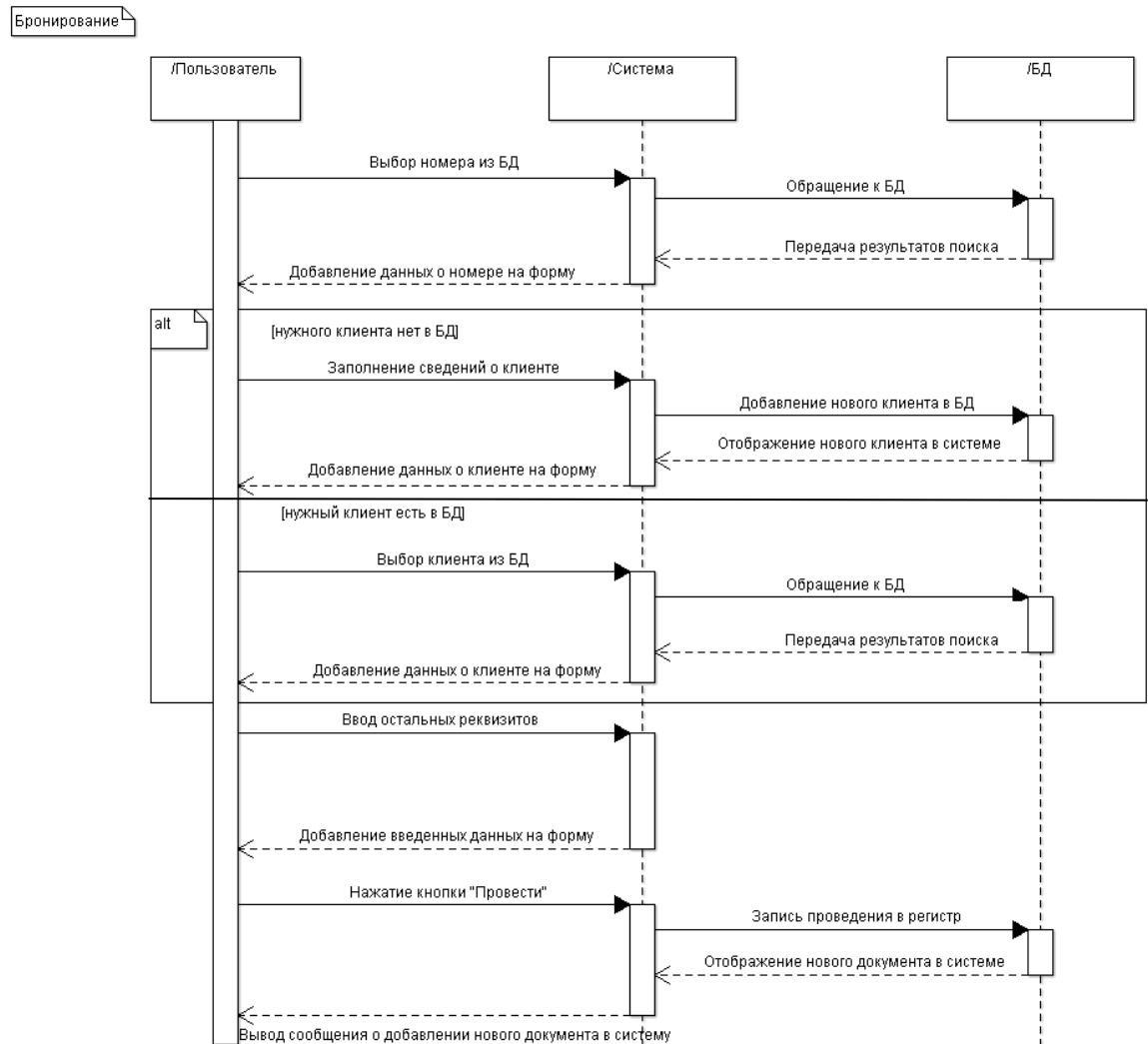


Рисунок 13 – Диаграмма последовательности прецедента «Создание нового бронирования»

Рассмотрим диаграмму последовательности прецедента «Создание нового бронирования». На ней показаны взаимодействия пользователя, системы и БД. В начале могут произойти 2 сценария:

- 1) клиента нет в БД – при этом сценарии пользователь должен заполнить все данные о клиенте и внести его в систему;
- 2) клиент есть в БД – в этом сценарии пользователь выбирает клиента из БД и добавляет его на форму.

Вне зависимости от выбранного сценария далее заполняются все элементы документа, и он добавляется в систему.

2.5 Планирование и расчет бюджета разрабатываемой системы

Для осуществления анализа была рассмотрена статистика заработной платы программиста 1С в промежутке от августа 2019 года по май 2020 года.

Диаграмма средней зарплаты разработчика представлена на рисунке 14.

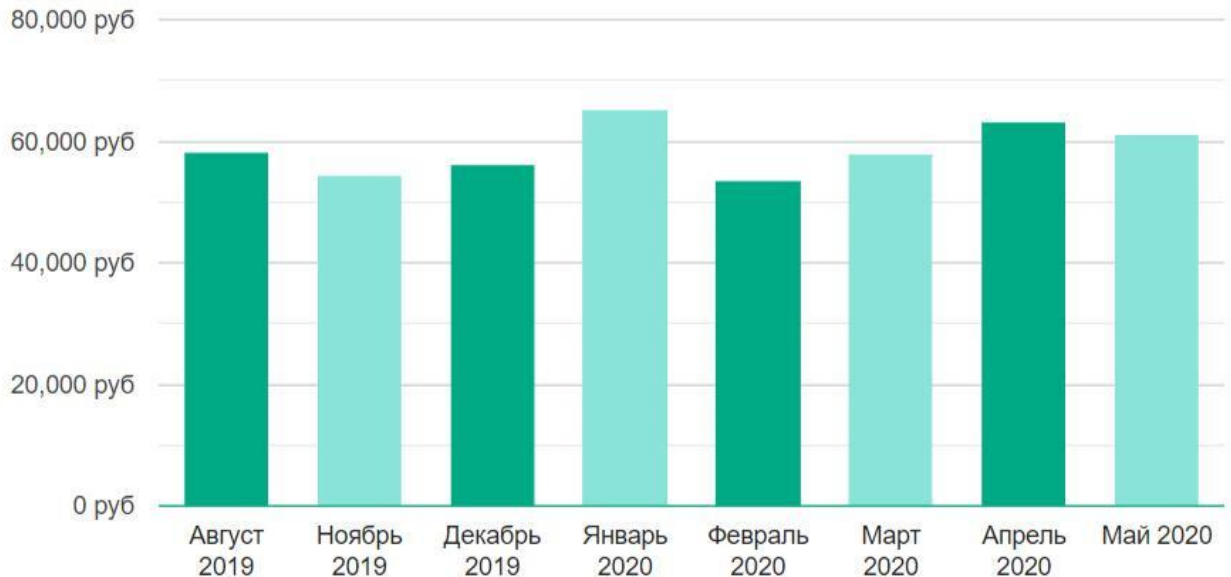


Рисунок 14 – Статистика заработной платы программиста 1С в России

В среднем в месяц разработчик 1С получает около 58663 рублей.

Согласно Трудовому кодексу Российской Федерации продолжительность рабочего времени не может превышать 40 часов в неделю. Количество часов в месяц примерно равняется 170 часам. Отсюда получим, что в среднем за час работы разработчик получает $58663/170=345$ руб/час. Затраты электроэнергии, потребляемой компьютером в среднем составляют 210 Вт/ч. Примерные затраты электричества монитором – до 40 Вт/ч. Одноставочный тариф на электроэнергию в среднем по России составляет 3,4 руб. за 1 кВт.

Выполнение выпускной работы состоит из пяти бизнес-процессов: «Анализ предметной области и постановка задачи», «Анализ требований и

проектирование», «Реализация», «Тестирование» и «Оформление пояснительной записки».

Был проведен анализ затрат времени и количества дней выполнения на каждый этап для одного разработчика.

Данные для формирования диаграммы Ганта представлены в таблице 15.

Таблица 15 – Информация по бизнес-процессам

Название задачи	Длительность	Начало	Окончание	Предшественники	Исполнитель
Выполнение ВКР	35 дней	20.04.2021	05.06.2021		
Анализ предметной области и постановка задачи	5 дней	20.04.2021	24.04.2021		Разработчик
Анализ требований и проектирование	6 дней	27.04.2021	04.05.2021	2	Разработчик; Электричество
Реализация	13 дней	05.05.2021	21.05.2021	3	Разработчик; Электричество
Тестирование	6 дней	22.05.2021	29.05.2021	4	Разработчик; Электричество
Оформление пояснительной записки	5 дней	01.06.2021	14.06.2021	5	Разработчик; Электричество

Далее по представленной выше таблице была построена диаграмма Ганта плана разработки, представленная на рисунке 15.

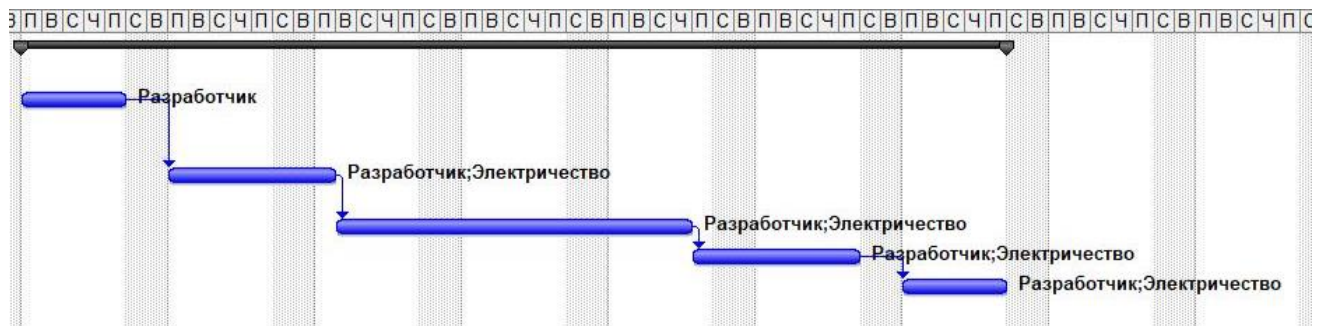


Рисунок 15 – Диаграмма Ганта

Результаты вычислений представлены в таблице 16.

Таблица 16 – Затраты на разработку

Этапы	Время на выполнение, ч	Зарплата разработчика, руб/ч	Затраты на электроэнергию, руб/ч	Общая сумма затрат, руб
Анализ предметной области и постановка задачи	40	345	0,085	13803,4
Анализ требований и проектирование	48	345	0,085	16564,08
Реализация	104	345	0,085	35888,84
Тестирование	48	345	0,085	16564,08
Оформление пояснительной записки	40	345	0,085	13803,4
Итого	280			96623,8

Предположительная стоимость разработки автоматизированной информационной системы для гостиничного бизнеса составляет 96624 рубля.

3 Реализация программного обеспечения

Реализация информационной системы по автоматизации учета гостиницы представляет собой конфигурацию на технологической платформе «1С: Предприятие». На выходе получится папка с файлами, необходимыми для ее запуска. Также потребуется установка программного продукта «1С: Предприятие».

Для этой системы была спроектирована диаграмма компонентов.

3.1 Диаграмма компонентов

Диаграмма компонентов отображает структуру представления информационной системы по автоматизации учета гостиницы. Она показывает взаимосвязи между компонентами программы. Диаграмма представлена на рисунке 16.

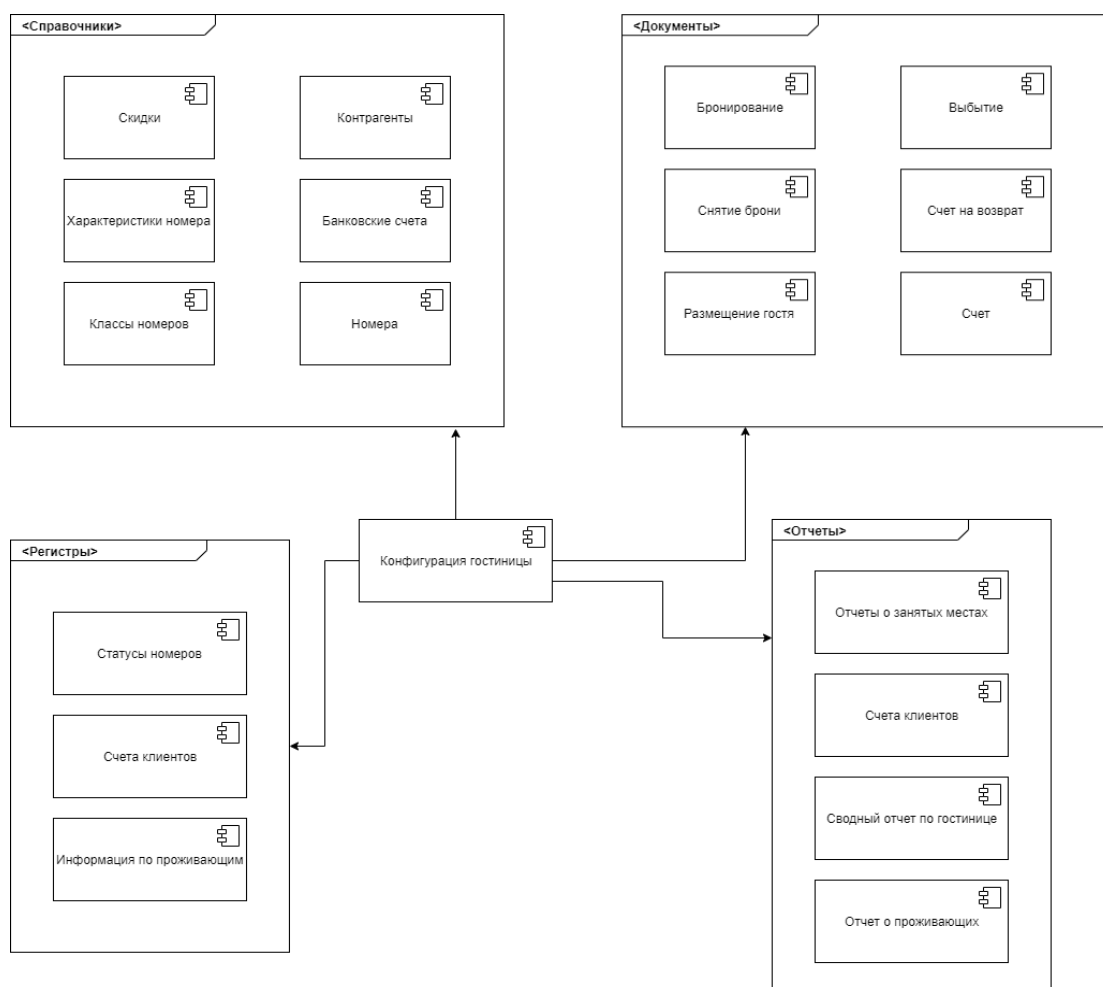


Рисунок 16 – Диаграмма компонентов ИС по автоматизации учета гостиницы

Конфигурация по автоматизации учета гостиницы состоит из различных компонентов:

- 1) справочник – позволяет хранить данные, имеющие одинаковую структуру и списочный характер;
- 2) документ – позволяет хранить информацию о совершенных хозяйственных операциях и событиях;
- 3) регистр – составляет основу механизма учета движения средств;
- 4) отчет – позволяет формировать данные в удобном для пользователя виде [8].

Разрабатываемая программа позволяет создавать отчеты для упрощения работы администратора. Рассмотрим их подробнее:

- 1) счета клиентов – для отображения данных о всех счетах клиентов за выбранный период.
- 2) отчет о проживающих – для отображения информации о клиентах, проживающих в гостинице за выбранный период.
- 3) отчет о занятых местах – для отображения информации о количестве занятых мест за определенный период.
- 4) сводный отчет по гостинице – для отображения информации о номерах.

Схема компоновки данных отчета «Счета клиентов» указана в листинге Б.1, «Отчет о проживающих» в листинге Б.2, «Отчет о занятых местах» в листинге Б.3 и «Сводный отчет по гостинице» в листинге Б.4.

3.2 Диаграмма классов

Диаграмма классов – диаграмма, предназначенная для представления модели статической структуры программной системы в терминологии классов объектно-ориентированного программирования.

Разобрать классы разрабатываемой информационной системы более подробно позволяет диаграмма классов, представленная на рисунке 17.

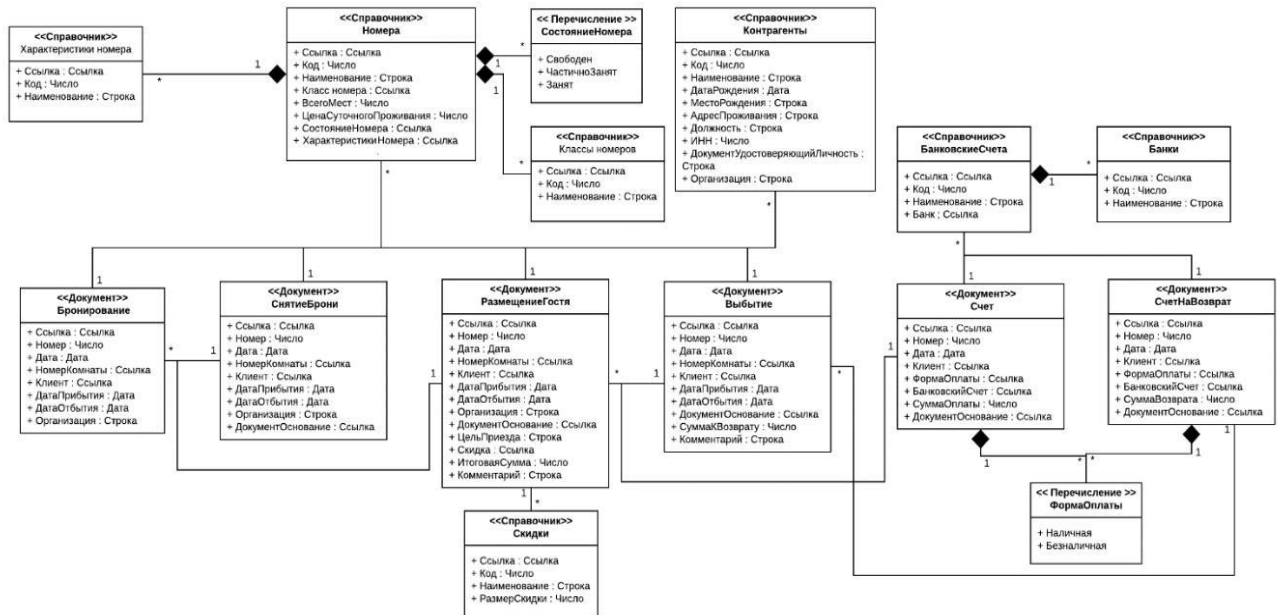


Рисунок 17 – Диаграмма классов

Рассмотрим классы информационной системы отеля.

В каждом справочнике имеются стандартные атрибуты такие как:

- 1) ссылка – она содержит в себе ссылку на элемент справочника;
- 2) код – номер элемента справочника;
- 3) наименование – название элемента справочника.

Справочник «ХарактеристикиНомера». Содержит в себе данные об удобствах, соответствующих тому или иному номеру отеля. Состоит исключительно из стандартных атрибутов.

Справочник «КлассыНомеров». Содержит в себе данные о количестве мест и типе номера. Состоит исключительно из стандартных атрибутов.

Справочник «Банки». Содержит в себе данные о банках, в которых обслуживаются контрагенты гостиницы. Состоит исключительно из стандартных атрибутов.

Справочник «Банковские счета». Содержит в себе данные о банковских счетах клиентов гостиницы. Состоит из стандартных атрибутов и ссылки на банк, в котором обслуживается клиент.

Справочник «Скидки». Содержит в себе данные о предоставляемых клиентам скидках. Состоит из стандартных атрибутов и размера предоставляемой скидки.

Справочник «Контрагенты». Содержит в себе данные о клиентах гостиницы. Состоит из стандартных атрибутов, даты и места рождения клиента, его адреса проживания, должности, организации, в которой он работает, ИНН, если клиент в командировке и информации удостоверяющей личность. Причем документ, удостоверяющий личность подставляется автоматически, как паспорт, так как это самый частый документ для регистрации нового гостя.

Справочник «Номера». Содержит в себе данные о гостиничных номерах. Состоит из стандартных атрибутов, количества мест для заселения, цены суточного проживания, класса, характеристик и состояния, соответствующих данному номеру. Причем состояние номера и количество мест определяются автоматически, что ускоряет работу администратора и исключает ошибки «человеческого фактора».

В каждом документе имеются стандартные атрибуты такие как:

- 1) ссылка – она содержит в себе ссылку на создаваем документ;
- 2) номер – код, присваиваемый документу;
- 3) дата – дата создания документа.

Документ «Бронирование». Отмечает факт бронирования клиентом номера и резервирует его в системе. Включает в себя стандартные атрибуты и некоторые другие:

- 1) номер комнаты – ссылается на элемент справочника «Номера», при этом на форме реализован механизм, в соответствие с которым администратор не сможет выбрать забронированную или уже занятую комнату;
- 2) клиент - ссылается на элемент справочника «Контрагенты»;
- 3) дата прибытия – дата, в которую клиент заселился в гостиницу;
- 4) дата отбытия – дата, в которую клиент выселится из отеля;
- 5) организация – предприятие, в котором работает клиент, заполняется автоматически из справочника «Контрагенты», при выборе клиента из БД, при этом администратору задается вопрос о том, стоит ли заполнять это поле.

Код модуля объекта данного документа представлен в листинге Б.5, модуля менеджера в листинге Б.6.

Документ «СнятиеБрони». Отмечает факт снятие брони клиентом и освобождает номер с резерва в системе. Включает в себя те же реквизиты, что и документ «Бронирование», за исключением атрибута «Документ-основание». При выборе данного атрибута все остальные поля заполняются автоматически. Так же для этого документа реализован механизм ввода на основании, который позволяет заполнить все поля нажатием на одну кнопку. Это уменьшает затраты времени на оформление документа и снижает нагрузку на администратора.

Код модуля объекта данного документа представлен в листинге Б.7, модуля менеджера в листинге Б.8.

Документ «РазмещениеГостя». Отмечает факт заселения клиента в номер и исключает данную комнату из списка свободных. Включает в себя стандартные атрибуты, реквизиты «Номер комнаты», «Клиент», «Дата прибытия», «Дата отбытия» и «Организация», которые описаны выше для документа «Бронирование», при этом, если создавать документ «РазмещениеГостя» вводом на основании документа «Бронирование» эти реквизиты подставляются автоматически, также это произойдет, если в поле «Документ-основание» выбрать нужное бронирование. Так же имеются и другие реквизиты, заполняемые вручную:

- 1) цель приезда – позволяет анализировать для каких целей чаще посещают гостиницу;
- 2) скидка – ссылается на справочник «Скидки» и уменьшает итоговую сумму на определенный процент, расчет происходит автоматически;
- 3) итоговая сумма – рассчитывается автоматически исходя из цены суточного проживания в номере и действующих для данного клиента скидок;
- 4) комментарий – несет в себе информацию справочного характера.

Код модуля объекта данного документа представлен в листинге Б.9, модуля менеджера в листинге Б.10 и модуля формы в листинге Б.11.

Документ «Выбытие». Отмечает факт выселения клиента из номера и освобождает резерв комнаты в системе. Включает в себя стандартные атрибуты, реквизиты «Номер комнаты», «Клиент», «Дата прибытия» и «Дата

отбытия», которые описаны выше для документа «Бронирование», при этом, если создавать документ «Выбытие» вводом на основании документа «РазмещениеГостя» эти реквизиты подставятся автоматически, также это произойдет, если в поле «Документ-основание» выбрать нужное размещение. Так же имеется поле «Комментарий», описанное выше и реквизит «Сумма к возврату», который заполняется программой автоматически, если дата выселения раньше, чем запланирована.

Код модуля объекта данного документа представлен в листинге Б.12, модуля менеджера в листинге Б.13 и модуля формы в листинге Б.14.

Документ «Счет». Отмечает факт оплаты клиентом стоимости проживания. Включает в себя стандартные атрибуты, реквизит «Клиент» уже описанный выше, при этом, если создавать документ «Счет» вводом на основании документа «РазмещениеГостя» этот реквизит и реквизит «Сумма оплаты» подставятся автоматически, также это произойдет, если в поле «Документ-основание» выбрать нужное размещение. Так же данный документ включает и другие поля:

- 1) форма оплаты – выбирается из перечисления «ФормаОплаты», что позволяет экономить время администратора;
- 2) банковский счет – отражает информацию по банковскому счету клиента, данное поле появляется только, если выбрать форму оплаты «Безналичная».

Код модуля объекта данного документа представлен в листинге Б.15, модуля менеджера в листинге Б.16 и модуля формы в листинге Б.17.

Документ «СчетНаВозврат». Отмечает факт возврата клиенту денежных средств. Включает в себя те же атрибуты, что и в документе «Счет», за исключением реквизита «Сумма оплаты». Вместо него имеется поле «Сумма возврата», которое рассчитывается автоматически, исходя из того, на сколько раньше оплаченного срока выселился клиент. Также данный документ можно ввести на основание документа «Выбытие», тогда большинство полей заполнятся автоматически, это же произойдет, если выбрать нужное выбытие в поле «Документ-основание».

Для всех документов предусмотрены печатные формы, которые создаются в соответствии с модулем менеджера.

Код модуля объекта данного документа представлен в листинге Б.18, модуля менеджера в листинге Б.19 и модуля формы в листинге Б.20.

4 Контроль качества программного обеспечения

4.1 Функциональное тестирование

Функциональное тестирование, так называемый метод «черного ящика», это тестирование программного обеспечения, с целью определить, может ли оно выполнить поставленные пользователем задачи. При этом выясняется, на решение каких задач способен тот или иной программный продукт.

Рассмотрим функциональное тестирование, используя прецедент «Справочник скидок». В этом случае, чтобы добавить очередную позицию нужно использовать корректную информацию.

4.1.1 Определение классов эквивалентности

Применяя метод эквивалентного разбиения сначала выделяют классы эквивалентности, затем переходят к составлению самих тестов.

Для тестирования поля ввода «Наименование» были выделены следующие классы эквивалентности:

- количество символов в строке.

Количество символов в строке должно находиться в промежутке от 1 до 25 включительно.

Для тестирования поля ввода «Размер скидки» были выделены следующие классы эквивалентности:

- допустимые значения;
- недопустимые значения.

К допустимым значениям относятся все целые неотрицательные числа, при условии, что их значение не будет превышать 100.

Не допустимые значения – все символы любого алфавита, специальные символы или число, превышающее 100.

На основании классов эквивалентности были сформированы возможные ситуации, которые представлены в таблицах 17, 18.

Таблица 17 – Функциональное тестирование поля ввода «Наименование»

Входное значение	Предполагаемый результат	Результат
Очень-очень большая скидка	В поле будут введены только первые 25 символов входного значения	Тест произведен, в поле ввелись только первые 25 символов входного значения (рисунок В.1)
Большая скидка	В поле будут введены все символы входного значения	Тест произведен, в поле ввелись все символы входного значения (рисунок В.2)

Таблица 18 – Функциональное тестирование поля ввода «Размер скидки»

Входное значение	Предполагаемый результат	Результат
80	В поле будут введены все символы входного значения	Тест произведен, в поле ввелись все символы входного значения (рисунок В.3)
Скидка = «80»	В поле будут введены только цифры из входного значения	Тест произведен, в поле ввелись только цифры из входного значения (рисунок В.3)

Функциональное тестирование показало, что система функционирует корректно и не имеет явных ошибок и сбоев в работе, что является положительным результатом проведения тестирования.

4.1.2 Определение граничных условий

Граничное тестирование состоит из тестов, которые контролируют, как ведет себя система на уровне входных данных, которые не соответствуют допустимому диапазону указанных значений. Граничное тестирование проведено на примере позиции ввода «Размер скидки».

Определены границы:

- логическая граница;

- произвольная граница;
- технологическая граница.

Логическая граница – все возможные значения не могут быть отрицательными. Произвольная граница – возможно указать значение только в диапазоне от 0 до 100 включительно. Технологическая граница – все возможные значения только целые числа. Используемые границы изображены на рисунке 19.



Рисунок 18 – Произвольная граница ввода поля «Размер скидки»

Тестовые наборы приведены в таблице 19.

Таблица 19 – Граничные условия

№	Входное значение	Ожидаемый результат	Результат
1	-20	В поле будут введены только цифры из входного значения	Тест произведен, в поле ввелись только цифры из входного значения (рисунок В.4)
2	0	В поле будут введены все символы входного значения	Тест произведен, в поле ввелись все символы сходного значения (рисунок В.5)
3	80	В поле будут введены все символы входного значения	Тест произведен, в поле ввелись все символы сходного значения (рисунок В.6)
4	100	В поле будут введены все символы входного значения	Тест произведен, в поле ввелись все символы сходного значения (рисунок В.7)
5	120	После введения данного числа, будет выдана ошибка о некорректности данных	Тест произведен, выдана ошибка (рисунок В.8)
6	60	В поле будут введены только цифры из входного значения	Тест произведен, в поле ввелись только цифры из входного значения (рисунок В.9)

Граничное тестирование проведено успешно в диапазоне от 0 до 100.

4.1.3 Метод функциональных диаграмм

Метод функциональных диаграмм актуален для определения тестов с наиболее плодотворными результатами.

Для тестирования было выбрано поле ввода «Цена суточного проживания» из справочника «Номера».

Причины:

- количество символов в строке находится в промежутке от 1 до 8 включительно;
- наличие допустимых символов;
- отсутствие недопустимых символов.

Следствие:

- успешный ввод символов;
- некорректный ввод символов.

Функциональная диаграмма для выбранных классов эквивалентности представлена на рисунке 19.

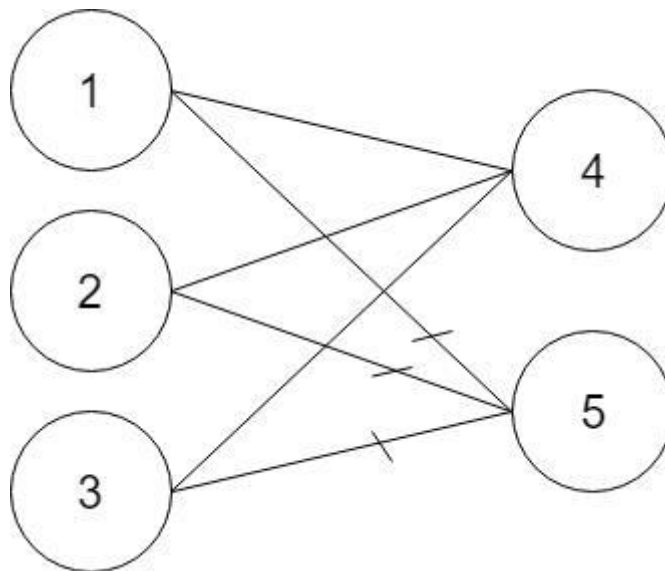


Рисунок 19 – Функциональная диаграмма

Причины и результаты тестов представлены в таблице 20.

Таблица 20 – Причины и тесты

Причины/тесты	1	2	3	4
1	0	-	-	1
2	-	0	-	1
3	-	-	0	1
4	0	0	0	1
5	1	1	1	0

Таким образом, тестирование для поля ввода «Цена суточного проживания» из справочника «Номера» было выполнено успешно.

4.1.4 Метод парного тестирования

Используя парное, другими словами комбинаторное, тестирование – составляют группу тестовых данных из общей группы данных, вводимых в систему, тем самым значительно уменьшается объем тест-кейсов. Парное тестирование проведено на примере позиции ввода «Цена суточного проживания» с применением программного обеспечения «allpairs». Запуск производится из консоли. Входные классы эквивалентности изображены в таблице 21.

Таблица 21 – Описание классов эквивалентности

length	acceptablesymbols	unacceptablesymbols	cost
[-;0]	no	no	[-;0]
[1;8]	yes	yes	[1;99999999]
[9;+]			[100000000;+]

Результаты создания тестовых случаев:

TEST CASES

case	length	acceptablesymbols	unacceptablesymbols	cost	pairings
1	[-;0]	no	no	[-;0]	6

2	[-;0]	yes	yes	[1;99999999]	6
3	[1;8]	yes	no	[-;0]	5
4	[1;8]	no	yes	[1;99999999]	5
5	[9;+]	no	no	[100000000;+]	5
6	[9;+]	yes	yes	[-;0]	4
7	[-;0]	yes	yes	[100000000;+]	3
8	[1;8]	~no	no	[1;99999999]	1
9	[1;8]	~yes	~no	[100000000;+]	1
10	[9;+]	~no	~yes	[1;99999999]	1

PAIRING DETAILS

var1	var2	value1	value2	appearances cases	
length cost	[-;0]	[-;0]	1	1	
length cost	[-;0]	[1;99999999]	1	2	
length cost	[-;0]	[100000000;+]	1	7	
length cost	[1;8]	[-;0]	1	3	
length cost	[1;8]	[1;99999999]	2	4, 8	
length cost	[1;8]	[100000000;+]	1	9	
length cost	[9;+]	[-;0]	1	6	
length cost	[9;+]	[1;99999999]	1	10	
length cost	[9;+]	[100000000;+]	1	5	
length acceptablesymbols	[-;0]	no	1	1	
length acceptablesymbols	[-;0]	yes	2	2, 7	
length acceptablesymbols	[1;8]	no	2	4, 8	
length acceptablesymbols	[1;8]	yes	2	3, 9	
length acceptablesymbols	[9;+]	no	2	5, 10	
length acceptablesymbols	[9;+]	yes	1	6	
length unacceptablesymbols	[-;0]	no	1	1	
length unacceptablesymbols	[-;0]	yes	2	2, 7	
length unacceptablesymbols	[1;8]	no	3	3, 8, 9	
length unacceptablesymbols	[1;8]	yes	1	4	
length unacceptablesymbols	[9;+]	no	1	5	

length	unacceptablesymbols	[9;+]	yes	2	6, 10
cost	acceptablesymbols	[-;0]	no	1	1
cost	acceptablesymbols	[-;0]	yes	2	3, 6
cost	acceptablesymbols	[1;99999999]	no	3	4, 8, 10
cost	acceptablesymbols	[1;99999999]	yes	1	2
cost	acceptablesymbols	[100000000;+]	no	1	5
cost	acceptablesymbols	[100000000;+]	yes	2	7, 9
cost	unacceptablesymbols	[-;0]	no	2	1, 3
cost	unacceptablesymbols	[-;0]	yes	1	6
cost	unacceptablesymbols	[1;99999999]	no	1	8
cost	unacceptablesymbols	[1;99999999]	yes	3	2, 4, 10
cost	unacceptablesymbols	[100000000;+]	no	2	5, 9
cost	unacceptablesymbols	[100000000;+]	yes	1	7
acceptablesymbols	unacceptablesymbols		no	no	3, 1,5,8
acceptablesymbols	unacceptablesymbols		no	yes	2, 4, 10
acceptablesymbols	unacceptablesymbols		yes	no	2, 3, 9

4.2 Структурное тестирование

Структурное тестирование – тестирование логики готовой программы или ее частей. Логика программы определяется управляющими конструкциями, а качество структурного тестирования – степенью, в которой тесты обеспечивают проверку (т.е. прохождение) всевозможных путей программы.

Структурное тестирование, в котором используется принцип «белого ящика», определяется уровнем покрытия тестами первоначального текста программного обеспечения. Эта методология тестирования подразумевает, что все пути программы выполнимы. В этой выпускной квалификационной работе тестирование по принципу белого ящика включает в себя метод покрытия операторов, метод покрытия условий и метод покрытия маршрутов. Для тестирования использован пример прецедента «Размещение гостя». Его блок-схема представлена на рисунке 20.

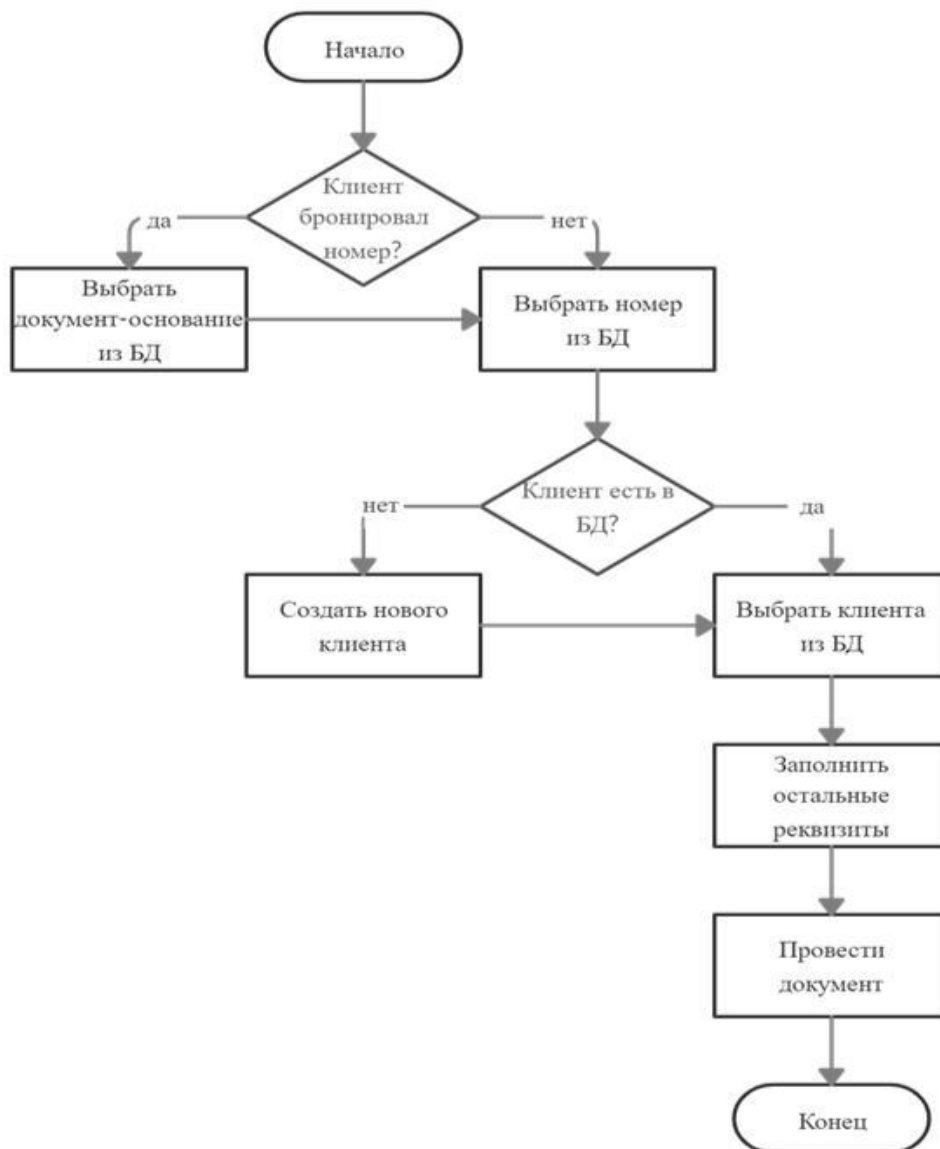


Рисунок 20 – Блок-схема прецедента «Размещение гостя»

Был построен функциональный граф, представленный на рисунке 21.

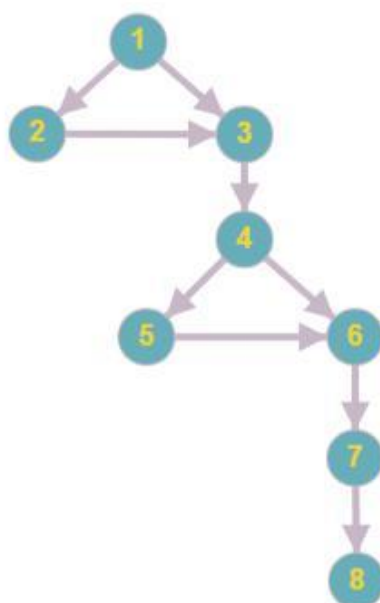


Рисунок 21 – Функциональный граф прецедента «Размещение гостя»

Тестирование методом «белого ящика» было произведено успешно.

4.2.1 Метод покрытия операторов

Тестирование программного обеспечения охватывает целый ряд видов деятельности, аналогичных последовательности процессов разработки программного обеспечения. В него входят: постановка задачи для теста, проектирование теста, написание тестов, тестирование тестов, выполнение тестов, изучение результатов тестирования.

Один из подходов основан на анализе логики программы – стратегия «белого ящика». Сущность подхода – в проверке каждого пути, каждой ветви алгоритма. При этом внешняя спецификация во внимание не принимается. Методы этой группы называются методами структурного тестирования.

Критерии покрытия операторов предполагают, что каждый оператор программного обеспечения будет выполнен хотя бы однажды. Это необходимое, но недостаточное условие для приемлемого тестирования по принципу «белого ящика» Представлен один тест с учетом выполнения всех операторов. Результаты представлены в таблице 22.

Таблица 22 – Результат тестирования методом покрытия операторов

Операторы	Тест	Фактический результат	Результат тестирования
1, 2, 3, 4, 5, 6, 7, 8	Пользователь выбрал клиента в БД. Все требуемые поля в документе заполнены.	Документ успешно проведен.	Успешно

Тестирование методом покрытия операторов выполнено успешно.

4.2.2 Метод покрытия условий

Идея метода покрытия условий в составлении определенного числа тестов, которого будет достаточно для того, чтобы все имеющиеся результаты каждого условия в решении осуществлялись хотя бы один раз. Необходимо соблюдение условия выполнения каждого оператора хотя бы один раз. Для демонстрации метода составлено четыре теста. Результаты представлены в таблице 23.

Таблица 23 – Результат тестирования методом покрытия условий

Условия	Тест	Фактический результат	Результат тестирования
1 – нет, нет	Документ-основание и клиент в системе не обнаружены.	Данные о документе-основании и клиенте отсутствуют.	Успешно
1 – да, 2 – нет	Документ-основание в системе обнаружен, клиент – нет.	Данные о клиенте отсутствуют.	Успешно
1 – нет, 2 – да	Клиент в системе обнаружен, документ-основание – нет.	Данные о документе-основании отсутствуют.	Успешно
1 – да, 2 – да,	Документ-основание и клиент обнаружены в системе.	Успешное проведение документа в системе.	Успешно

Тестирование методом покрытия условий выполнено успешно. По результатам проведенных тестов, можно сказать, что информационная система работает корректно, выполняет все необходимые задачи.

4.2.3 Метод покрытия маршрутов

Метод покрытия маршрутов основан на прохождении по всем путям программного обеспечения, которые только возможны. При этом это самый действенный метод.

С целью выполнения всех маршрутов составлено 4 теста. Результаты представлены в таблице 24.

Таблица 24 – Результат тестирования методом покрытия маршрутов

Маршрут	Тест	Фактический результат	Результат тестирования
1, 2, 3, 4, 5, 6, 7, 8	Пользователь выбрал документ-основание в БД, создал нового клиента. Все требуемые поля в документе заполнены.	Документ успешно проведен.	Успешно
1,2,3,4,6,7,8	Пользователь выбрал документ-основание и клиента в БД. Все требуемые поля в документе заполнены.	Документ успешно проведен.	Успешно
1, 3, 4, 5, 6, 7, 8	Пользователь добавил клиента в БД. Все требуемые поля в документе заполнены.	Документ успешно проведен.	Успешно
1,3,4,6,7,8	Пользователь выбрал клиента в БД. Все требуемые поля в документе заполнены.	Документ успешно проведен.	Успешно

Тестирование методом покрытия маршрутов выполнено успешно. По результатам проведенных тестов, можно сказать, что информационная система работает корректно, выполняет все необходимые задачи.

Заключение

В рамках данной выпускной квалификационной работы с помощью встроенного языка 1С и многофункционального языка UML была разработана информационная система по автоматизации учета гостиницы, упрощающая работу администратора и уменьшающая время ожидания клиента. Удобный, интуитивно понятный интерфейс, автоматически вычисляемые поля и предопределенные данные позволяют сократить время, затрачиваемое на оформление документов и добавление объектов в справочники. Механизм формирования отчетов позволяет создавать все отчеты, требуемые для данного вида деятельности.

На этапе тестирования использовались различные методы, такие как: определение классов эквивалентности, определение граничных условий, метод функциональных диаграмм, метод парного тестирования, метод покрытия операторов, метод покрытия условий и метод покрытия маршрутов. В процессе тестирования ошибки не выявлены. Информационная система работает исправно.

Таким образом, в результате выполнения выпускной квалификационной работы была разработана конфигурация по автоматизации учета гостиницы, отвечающая всем требованиям технического задания.

Список использованных источников

1. Что такое гостиничный бизнес? [Электронный ресурс]. URL: https://vsetreningi.ru/schools/gostinichnyj_biznes/ (дата обращения 10.05.2021).
2. Возможности системы. [Электронный ресурс] URL: <https://hotelcloud.ru/index.php/about-system/facilities/#orders/> (дата обращения 10.05.2021).
3. Интеграция с CRM. [Электронный ресурс]. URL: <https://bnovo.ru/crm> (дата обращения 11.05.2021).
4. Контур.ОТЕЛЬ – PMS система управления гостиницей. [Электронный ресурс] URL: <https://kontur.ru/hotel?p=w03698> (дата обращения 11.05.2021).
5. HotelCloud – система управления гостиницей [Электронный ресурс] URL: <https://www.hotelcloud.ru/> (дата обращения 11.05.2021).
6. Bnovo (Биново) – программа для управления отелями, хостелами и апартаментами [Электронный ресурс] URL : <https://bnovo.ru/> (дата обращения 11.05.2021).
7. Rational Unified Process. [Электронный ресурс]. URL: http://www.interface.ru/rational/ru01_t.htm (дата обращения 01.06.2021).
8. Общая характеристика языка UML. [Электронный ресурс]. URL: <http://www.informicus.ru/default.aspx?SECTION=6&id=73&subdivisionid=2> (дата обращения 02.06.2021).
9. 1С:Предприятие8.3 [Электронный ресурс]. URL:http://v8.1c.ru/overview/Term_000000169.htm (дата обращения 31.05.2021).
10. Информация о гостиничном бизнесе [Электронный ресурс]. URL: <http://inn35.ru/gostinichnyj-biznes-prostymi-slovami%20/> (дата обращения 03.06.2021).
11. Построение диаграмм [Электронный ресурс]. URL: <https://brasmlibras.ru/postroenie-uml-diagram%20/> (дата обращения 03.06.2021).
12. Статистика заработной платы программиста 1С [Электронный ресурс]. URL:<https://russia.trud.com/salary/692/3356.html>. (дата обращения 03.06.2021).

Приложение А
ГЛОССАРИЙ
(справочное)

Информационная система (ИС) – взаимосвязанный комплекс средств, методов и трудовых ресурсов, который направлен на хранение, модификацию и выдачу информации для дальнейшего использования.

Выпускная квалификационная работа (ВКР) – это итоговая аттестационная работа студента, выполненная им на последнем курсе обучения. ВКР оформляется в письменном виде в соответствии с методическими указаниями и с соблюдением всех правил написания таких работ, на завершающем этапе обучения подлежит защите перед государственной аттестационной комиссией.

Программное обеспечение (ПО) – это совокупность программ, позволяющих взаимодействовать с компьютером, решая с его помощью все необходимые задачи.

Приложение Б
ЛИСТИНГ ПРИЛОЖЕНИЯ
(рекомендуемое)

Листинг Б.1 – Код схемы компоновки данных отчета «Счета клиентов»

```

ВЫБРАТЬ
    СчетаКлиентов.Регистратор.Ссылка КАК Счет,
    СчетаКлиентов.Клиент,
    СчетаКлиентов.Сумма
ИЗ
    РегистрНакопления.СчетаКлиентов КАК СчетаКлиентов

```

Листинг Б.2 – Код схемы компоновки данных отчета «Отчет о проживающих»

```

ВЫБРАТЬ
    ИнформацияПоПроживающим.НомерКомнаты КАК Номер,
    ИнформацияПоПроживающим.ДатаЗаезда КАК ДатаЗаезда,
    ИнформацияПоПроживающим.ДатаВыселения КАК ДатаВыселения,
    ИнформацияПоПроживающим.Клиент КАК Клиент,
    ИнформацияПоПроживающим.Сумма КАК Сумма
ИЗ
    Документ.Выбытие КАК Выбытие,
    РегистрНакопления.ИнформацияПоПроживающим КАК
ИнформацияПоПроживающим
ГДЕ
    Выбытие.ДокументОснование.Ссылка <>
ИнформацияПоПроживающим.Регистратор.Ссылка

```

Листинг Б.3 – Код схемы компоновки данных отчета «Отчет о занятых местах»

```

ВЫБРАТЬ
    КоличествоМестВНомерахОстаткиИОбороты.НомерКомнаты КАК
Номер,
    КоличествоМестВНомерахОстаткиИОбороты.КоличествоЗанятыхМестК
онечныйОстаток КАК КоличествоЗанятыхМест,
    КоличествоМестВНомерах.Период
ИЗ
    РегистрНакопления.КоличествоМестВНомерах.ОстаткиИОбороты КАК
КоличествоМестВНомерахОстаткиИОбороты
    ВНУТРЕННЕЕ СОЕДИНЕНИЕ
РегистрНакопления.КоличествоМестВНомерах КАК
КоличествоМестВНомерах
    ПО
КоличествоМестВНомерахОстаткиИОбороты.НомерКомнаты.Ссылка =
КоличествоМестВНомерах.НомерКомнаты.Ссылка
ГДЕ
    КоличествоМестВНомерахОстаткиИОбороты.КоличествоЗанятыхМестК
онечныйОстаток <> 0

```

Листинг Б.4 – Код схемы компоновки данных отчета «Сводный отчет по гостинице»

```

ВЫБРАТЬ
    Счет.ДокументОснование.НомерКомнаты КАК Номер,
    КОЛИЧЕСТВО(РАЗЛИЧНЫЕ Счет.Ссылка) КАК
КоличествоВыставленныхСчетов,
    СУММА(СчетаКлиентовОбороты.СуммаОборот) КАК Сумма
ПОМЕСТИТЬ Вт
ИЗ
    РегистрНакопления.СчетаКлиентов.Обороты КАК
СчетаКлиентовОбороты
    ВНУТРЕННЕЕ СОЕДИНЕНИЕ Документ.Счет КАК Счет
    ПО СчетаКлиентовОбороты.Клиент.Ссылка =
Счет.Клиент.Ссылка

СГРУППИРОВАТЬ ПО
    Счет.ДокументОснование.НомерКомнаты
;

////////////////////////////////////
////////////////////////////////////
ВЫБРАТЬ
    СправочникНомеров.КлассНомера,
    Вт.Номер,
    СправочникНомеров.ВсегоМест,
    СправочникНомеров.ВсегоМест -
СправочникНомеров.СвободныеМеста КАК Занято,
    Вт.КоличествоВыставленныхСчетов,
    Вт.Сумма
ИЗ
    Вт КАК Вт
    ЛЕВОЕ СОЕДИНЕНИЕ Справочник.СправочникНомеров КАК
СправочникНомеров
    ПО Вт.Номер = СправочникНомеров.Ссылка

```

Листинг Б.5 – Код модуля объекта документа «Бронирование»

Процедура ОбработкаПроведения(Отказ, РежимПроведения)

```

Спр = НомерКомнаты.ПолучитьОбъект();
Спр.СвободныеМеста = Спр.СвободныеМеста-1;
Если Спр.СвободныеМеста<>0 Тогда

    Спр.СостояниеНомера =
ПредопределенноеЗначение("Перечисление.СостояниеНомера.ЧастичноЗ
анят");
Иначе
    Спр.СостояниеНомера =
ПредопределенноеЗначение("Перечисление.СостояниеНомера.Занят");
КонецЕсли;

```

```

Спр.Записать ( ) ;
Движения.СтатусыНомеров.Записывать = Истина;
Движение = Движения.СтатусыНомеров.Добавить ( ) ;
Движение.Период = Дата;
Движение.НомерКомнаты = НомерКомнаты;
Движение.СостояниеНомера = Спр.СостояниеНомера;

```

```

Движение.КоличествоЗанятыхМест = Спр.ВсегоМест -
Спр.СвободныеМеста;

```

```

КонецПроцедуры

```

Листинг Б.6 – Код модуля менеджера документа «Бронирование»

Процедура Печать (ТабДок, Ссылка) Экспорт

```

Макет = Документы.Бронирование.ПолучитьМакет ("Печать") ;
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
|     Бронирование.Дата,
|     Бронирование.ДатаОтбытия,
|     Бронирование.ДатаПрибытия,
|     Бронирование.Клиент,
|     Бронирование.Номер,
|     Бронирование.НомерКомнаты,
|     Бронирование.Организация
| ИЗ
|     Документ.Бронирование КАК Бронирование
| ГДЕ
|     Бронирование.Ссылка В (&Ссылка) ";
Запрос.Параметры.Вставить ("Ссылка", Ссылка);
Выборка = Запрос.Выполнить ().Выбрать ();

```

```

ОбластьЗаголовков = Макет.ПолучитьОбласть ("Заголовков") ;
Шапка = Макет.ПолучитьОбласть ("Шапка") ;
ТабДок.Очистить ( ) ;

```

```

ВставляяРазделительСтраниц = Ложь;
Пока Выборка.Следующий ( ) Цикл
    Если ВставляяРазделительСтраниц Тогда
        ТабДок.ВывестиГоризонтальныйРазделительСтраниц ( ) ;
    КонецЕсли;

```

```

ТабДок.Вывести (ОбластьЗаголовков) ;

```

```

Шапка.Параметры.Заполнить (Выборка) ;
ТабДок.Вывести (Шапка, Выборка.Уровень ( ) ) ;

```

```

ВставляяРазделительСтраниц = Истина;

```

```

КонецЦикла;

```

```

КонецПроцедуры

```

Листинг Б.7 – Код модуля объекта документа «Снятие брони»

Процедура ОбработкаЗаполнения (ДанныеЗаполнения,
СтандартнаяОбработка)

```

    Если ТипЗнч (ДанныеЗаполнения) =
    Тип ("ДокументСсылка.Бронирование") Тогда
        ДатаОтбытия = ДанныеЗаполнения.ДатаОтбытия;

        ДатаПрибытия = ДанныеЗаполнения.ДатаПрибытия;
        Клиент = ДанныеЗаполнения.Клиент;
        НомерКомнаты = ДанныеЗаполнения.НомерКомнаты;
        Организация = ДанныеЗаполнения.Организация;
        ДокументОснование = ДанныеЗаполнения.Ссылка;

    КонецЕсли;

```

КонецПроцедуры

Процедура ОбработкаПроведения (Отказ, РежимПроведения)

```

    Спр = НомерКомнаты.ПолучитьОбъект ();
    Спр.СвободныеМеста = Спр.СвободныеМеста+1;
    Если Спр.СвободныеМеста<Спр.ВсегоМест Тогда

        Спр.СостояниеНомера =
        ПредопределенноеЗначение ("Перечисление.СостояниеНомера.ЧастичноЗ
        анят");
        Иначе
            Спр.СостояниеНомера =
            ПредопределенноеЗначение ("Перечисление.СостояниеНомера.Свободен"
            );
        КонецЕсли;
        Спр.Записать ();

        Движения.СтатусыНомеров.Записывать = Истина;
        Движение = Движения.СтатусыНомеров.Добавить ();
        Движение.Период = Дата;
        Движение.НомерКомнаты = НомерКомнаты;
        Движение.СостояниеНомера = Спр.СостояниеНомера;
        Движение.КоличествоЗанятыхМест = Спр.ВсегоМест-
        Спр.СвободныеМеста;

    КонецПроцедуры

```

Листинг Б.8 – Код модуля менеджера документа «СнятиеБрони»

```

Процедура Печать(ТабДок, Ссылка) Экспорт
    Макет = Документы.СнятиеБрони.ПолучитьМакет("Печать");
    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        |     СнятиеБрони.Дата,
        |     СнятиеБрони.ДатаОтбытия,
        |     СнятиеБрони.ДатаПрибытия,
        |     СнятиеБрони.ДокументОснование,
        |     СнятиеБрони.Клиент,
        |     СнятиеБрони.Номер,
        |     СнятиеБрони.НомерКомнаты,
        |     СнятиеБрони.Организация
        | ИЗ
        |     Документ.СнятиеБрони КАК СнятиеБрони

        | ГДЕ
        |     СнятиеБрони.Ссылка В (&Ссылка)";
    Запрос.Параметры.Вставить("Ссылка", Ссылка);
    Выборка = Запрос.Выполнить().Выбрать();

    ОбластьЗаголовков = Макет.ПолучитьОбласть("Заголовков");
    Шапка = Макет.ПолучитьОбласть("Шапка");
    ТабДок.Очистить();

    ВставлятьРазделительСтраниц = Ложь;
    Пока Выборка.Следующий() Цикл
        Если ВставлятьРазделительСтраниц Тогда
            ТабДок.ВывестиГоризонтальныйРазделительСтраниц();
        КонецЕсли;

        ТабДок.Вывести(ОбластьЗаголовков);

        Шапка.Параметры.Заполнить(Выборка);
        ТабДок.Вывести(Шапка, Выборка.Уровень());

        ВставлятьРазделительСтраниц = Истина;
    КонецЦикла;
КонецПроцедуры

```

Листинг Б.9 – Код модуля объекта документа «Размещение гостя»

Процедура ОбработкаЗаполнения (ДанныеЗаполнения,
СтандартнаяОбработка)

Если ТипЗнч (ДанныеЗаполнения) =
Тип ("ДокументСсылка.Бронирование") Тогда

ДатаОтбытия = ДанныеЗаполнения.ДатаОтбытия;
ДатаПрибытия = ДанныеЗаполнения.ДатаПрибытия;
Клиент = ДанныеЗаполнения.Клиент;
НомерКомнаты = ДанныеЗаполнения.НомерКомнаты;
Организация = ДанныеЗаполнения.Организация;
ДокументОснование = ДанныеЗаполнения.Ссылка;

КонецЕсли;

КонецПроцедуры

Процедура ОбработкаПроведения (Отказ, РежимПроведения)

Если ДокументОснование.Пустая () Тогда

Спр = НомерКомнаты.ПолучитьОбъект ();
Спр.СвободныеМеста = Спр.СвободныеМеста-1;
Если Спр.СвободныеМеста<>0 Тогда

Спр.СостояниеНомера =
ПредопределенноеЗначение ("Перечисление.СостояниеНомера.ЧастичноЗ
анят");

Иначе

Спр.СостояниеНомера =
ПредопределенноеЗначение ("Перечисление.СостояниеНомера.Занят");
КонецЕсли;
Спр.Записать ();
Иначе Спр = НомерКомнаты.ПолучитьОбъект ();

КонецЕсли;

Движения.ИнформацияПоПроживающим.Записывать = Истина;
Движение = Движения.ИнформацияПоПроживающим.Добавить ();
Движение.Период = Дата;
Движение.НомерКомнаты = НомерКомнаты;
Движение.ДатаЗаезда = ДатаПрибытия;
Движение.ДатаВыселения = ДатаОтбытия;
Движение.Клиент = Клиент;
Движение.Сумма = ИтоговаяСумма;

Движения.СтатусыНомеров.Записывать = Истина;
Движение = Движения.СтатусыНомеров.Добавить ();
Движение.Период = Дата;

```

Движение.НомерКомнаты = НомерКомнаты;
Движение.СостояниеНомера = Спр.СостояниеНомера;
Движение.КоличествоЗанятыхМест = Спр.ВсегоМест-
Спр.СвободныеМеста;

```

КонецПроцедуры

Листинг Б.10 – Код модуля менеджера документа «Размещение гостя»

```

Процедура Печать (ТабДок, Ссылка) Экспорт
    Макет = Документы.РазмещениеГостя.ПолучитьМакет ("Печать");
    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        |     РазмещениеГостя.Дата,
        |     РазмещениеГостя.ДатаКомандировочногоУдостоверения,
        |     РазмещениеГостя.ДатаОтбытия,
        |     РазмещениеГостя.ДатаПрибытия,
        |     РазмещениеГостя.ДокументОснование,
        |     РазмещениеГостя.ИтоговаяСумма,
        |     РазмещениеГостя.Клиент,
        |     РазмещениеГостя.Комментарий,
        |     РазмещениеГостя.Номер,
        |     РазмещениеГостя.НомерКомандировочногоУдостоверения,
        |     РазмещениеГостя.НомерКомнаты,
        |     РазмещениеГостя.Организация,
        |     РазмещениеГостя.Скидка,
        |     РазмещениеГостя.СтоимостьПроживанияБезСкидки,
        |     РазмещениеГостя.СтоимостьПроживанияСоСкидкой,
        |     РазмещениеГостя.ЦельПриезда
        | ИЗ
        |     Документ.РазмещениеГостя КАК РазмещениеГостя
        | ГДЕ

        |     РазмещениеГостя.Ссылка В (&Ссылка)";
    Запрос.Параметры.Вставить ("Ссылка", Ссылка);
    Выборка = Запрос.Выполнить ().Выбрать ();

    ОбластьЗаголовков = Макет.ПолучитьОбласть ("Заголовков");
    Шапка = Макет.ПолучитьОбласть ("Шапка");
    ТабДок.Очистить ();

    ВставлятьРазделительСтраниц = Ложь;
    Пока Выборка.Следующий () Цикл
        Если ВставлятьРазделительСтраниц Тогда
            ТабДок.ВывестиГоризонтальныйРазделительСтраниц ();
        КонецЕсли;

        ТабДок.Вывести (ОбластьЗаголовков);

        Шапка.Параметры.Заполнить (Выборка);
        ТабДок.Вывести (Шапка, Выборка.Уровень ());

```

```

        ВставитьРазделительСтраниц = Истина;
    КонецЦикла;
КонецПроцедуры

```

Листинг Б.11 – Код модуля формы документа «Размещение гостя»

```

&НаСервереБезКонтекста
Функция ПолучитьЦенуНомера (НомерКомнаты)
    Спр = НомерКомнаты.ПолучитьОбъект();
    Возврат Спр.ЦенаСуточногоПроживания;
КонецФункции
&НаСервереБезКонтекста
Функция ПолучитьРазмерСкидки (Скидка)
    Спр = Скидка.ПолучитьОбъект();
    Возврат Спр.РазмерСкидки;
КонецФункции
&НаКлиенте
Функция ПолучитьКоличествоДней ()
    Возврат (НачалоДня (Объект.ДатаОтбытия) –
НачалоДня (Объект.ДатаПрибытия)) / 60 / 60 / 24 + 1;
КонецФункции
&НаКлиенте
Функция ПолучитьСтоимостьБезСкидки ()
    Возврат
ПолучитьЦенуНомера (Объект.НомерКомнаты) * ПолучитьКоличествоДней ()

КонецФункции
&НаКлиенте
Функция ПолучитьСтоимостьСоСкидкой ()
    Если НЕ Объект.Скидка.Пустая () Тогда
        Возврат (1 –
ПолучитьРазмерСкидки (Объект.Скидка) / 100) * ПолучитьЦенуНомера (Объе
кт.НомерКомнаты) * ПолучитьКоличествоДней ();
    КонецЕсли;
КонецФункции

&НаКлиенте
Процедура ЗаполнитьЦены ()
    Если Объект.Скидка.Пустая () Тогда
        Объект.СтоимостьПроживанияСоСкидкой = 0;
        Объект.СтоимостьПроживанияБезСкидки =
ПолучитьСтоимостьБезСкидки ();
        Объект.ИтоговаяСумма =
Объект.СтоимостьПроживанияБезСкидки;
    Иначе
        Объект.СтоимостьПроживанияСоСкидкой =
ПолучитьСтоимостьСоСкидкой ();
        Объект.ИтоговаяСумма =
Объект.СтоимостьПроживанияСоСкидкой;
    КонецЕсли;
КонецПроцедуры
&НаКлиенте
Процедура ПриОткрытии (Отказ)

```

```

        Если ЗначениеЗаполнено (Объект.НомерКомнаты) Тогда
            ЗаполнитьЦены();
        КонецЕсли;
    КонецПроцедуры
    &НаКлиенте
    Процедура СкидкаПриИзменении (Элемент)
        ЗаполнитьЦены();
    КонецПроцедуры
    &НаКлиенте
    Процедура ДатаПрибытияПриИзменении (Элемент)
        ЗаполнитьЦены();
    КонецПроцедуры
    &НаКлиенте
    Процедура ДатаОтбытияПриИзменении (Элемент)
        ЗаполнитьЦены();
    КонецПроцедуры

```

Листинг Б.12 – Код модуля объекта документа «Выбытие»

Процедура ОбработкаЗаполнения (ДанныеЗаполнения,
СтандартнаяОбработка)

```

        Если ТипЗнч (ДанныеЗаполнения) =
        Тип ("ДокументСсылка.РазмещениеГостя") Тогда
            ДатаОтбытия = ДанныеЗаполнения.ДатаОтбытия;
            ДатаПрибытия = ДанныеЗаполнения.ДатаПрибытия;
            Клиент = ДанныеЗаполнения.Клиент;
            НомерКомнаты = ДанныеЗаполнения.НомерКомнаты;
            ДокументОснование = ДанныеЗаполнения.Ссылка;
        КонецЕсли;
    КонецПроцедуры

```

Процедура ОбработкаПроведения (Отказ, РежимПроведения)

```

        Спр = НомерКомнаты.ПолучитьОбъект();
        Спр.СвободныеМеста = Спр.СвободныеМеста+1;
        Если Спр.СвободныеМеста<Спр.ВсегоМест Тогда

            Спр.СостояниеНомера =
            ПредопределенноеЗначение ("Перечисление.СостояниеНомера.ЧастичноЗ
            анят");
            Иначе
                Спр.СостояниеНомера =
            ПредопределенноеЗначение ("Перечисление.СостояниеНомера.Свободен"
            );
            КонецЕсли;
        Спр.Записать();
    КонецПроцедуры

```

```

Движения.СтатусыНомеров.Записывать = Истина;
Движение = Движения.СтатусыНомеров.Добавить ();
Движение.Период = Дата;
Движение.НомерКомнаты = НомерКомнаты;
Движение.СостояниеНомера = Спр.СостояниеНомера;
Движение.КоличествоЗанятыхМест = Спр.ВсегоМест-
Спр.СвободныеМеста;

```

КонецПроцедуры

Листинг Б.13 – Код модуля менеджера документа «Выбытие»

```

Процедура Печать (ТабДок, Ссылка) Экспорт
Макет = Документы.Выбытие.ПолучитьМакет ("Печать");
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
|     Выбытие.Дата,
|     Выбытие.ДатаОтбытия,
|     Выбытие.ДатаПрибытия,
|     Выбытие.ДокументОснование,
|     Выбытие.Клиент,
|     Выбытие.Комментарий,
|     Выбытие.Номер,
|     Выбытие.НомерКомнаты,
|     Выбытие.СуммаКВозврату
| ИЗ
|     Документ.Выбытие КАК Выбытие
| ГДЕ
|     Выбытие.Ссылка В (&Ссылка)";
Запрос.Параметры.Вставить ("Ссылка", Ссылка);
Выборка = Запрос.Выполнить ().Выбрать ();

ОбластьЗаголовков = Макет.ПолучитьОбласть ("Заголовков");
Шапка = Макет.ПолучитьОбласть ("Шапка");
ТабДок.Очистить ();

ВставлятьРазделительСтраниц = Ложь;
Пока Выборка.Следующий () Цикл
    Если ВставлятьРазделительСтраниц Тогда

        ТабДок.ВывестиГоризонтальныйРазделительСтраниц ();
        КонецЕсли;

        ТабДок.Вывести (ОбластьЗаголовков);

        Шапка.Параметры.Заполнить (Выборка);
        ТабДок.Вывести (Шапка, Выборка.Уровень ());

        ВставлятьРазделительСтраниц = Истина;
    КонецЦикла;
КонецПроцедуры

```

Листинг Б.14 – Код модуля формы документа «Выбытие»

```

&НаКлиенте
Процедура ПриОткрытии(Отказ)
    ЛишниеДни = (НачалоДня(Объект.ДатаОтбытия) -
НачалоДня(Объект.Дата)) / 60 / 60 / 24;

    Если ЛишниеДни >= 0 Тогда
        ОбщееКоличествоДней = (НачалоДня(Объект.ДатаОтбытия) -
НачалоДня(Объект.ДатаПрибытия)) / 60 / 60 / 24 + 1;
        Объект.СуммаКВозврату =
ПриОткрытииНаСервере(ЛишниеДни, ОбщееКоличествоДней, Объект.Докуме
нтОснование);

        КонецЕсли;
КонецПроцедуры

&НаСервереБезКонтекста
Функция ПриОткрытииНаСервере(ЛишниеДни, ОбщееКоличествоДней,
Документ)

    Док = Документ.ПолучитьОбъект();
    Если Док.Скидка.Пустая() Тогда

        Возврат
Док.СтоимостьПроживанияБезСкидки / ОбщееКоличествоДней * ЛишниеДни;

    Иначе Возврат
Док.СтоимостьПроживанияСоСкидкой / ОбщееКоличествоДней * ЛишниеДни;
    КонецЕсли;

КонецФункции

```

Листинг Б.15 – Код модуля объекта документа «Счет»

```

Процедура ОбработкаЗаполнения(ДанныеЗаполнения,
СтандартнаяОбработка)
    Если ТипЗнч(ДанныеЗаполнения) =
Тип("ДокументСсылка.РазмещениеГостя") Тогда
        Клиент = ДанныеЗаполнения.Клиент;
        ДокументОснование = ДанныеЗаполнения.Ссылка;
        СуммаОплаты = ДанныеЗаполнения.ИтоговаяСумма;

        КонецЕсли;
КонецПроцедуры

Процедура ОбработкаПроведения(Отказ, Режим)

    Движения.СчетаКлиентов.Записывать = Истина;
    Движение = Движения.СчетаКлиентов.Добавить();
    Движение.Период = Дата;
    Движение.Клиент = Клиент;
    Движение.Сумма = СуммаОплаты;

```

КонецПроцедуры

Листинг Б.16 – Код модуля менеджера документа «Счет»

```

Процедура Печать (ТабДок, Ссылка) Экспорт
    Макет = Документы.Счет.ПолучитьМакет ("Печать");
    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        |     Счет.БанковскийСчет,
        |     Счет.Дата,
        |     Счет.ДокументОснование,
        |     Счет.Клиент,
        |     Счет.Номер,
        |     Счет.СуммаОплаты,
        |     Счет.ФормаОплаты
        | ИЗ
        |     Документ.Счет КАК Счет
        | ГДЕ
        |     Счет.Ссылка В (&Ссылка)";
    Запрос.Параметры.Вставить ("Ссылка", Ссылка);
    Выборка = Запрос.Выполнить ().Выбрать ();

    ОбластьЗаголовок = Макет.ПолучитьОбласть ("Заголовок");
    Шапка = Макет.ПолучитьОбласть ("Шапка");
    ТабДок.Очистить ();

    ВставлятьРазделительСтраниц = Ложь;
    Пока Выборка.Следующий () Цикл
        Если ВставлятьРазделительСтраниц Тогда
            ТабДок.ВывестиГоризонтальныйРазделительСтраниц ();
        КонецЕсли;

        ТабДок.Вывести (ОбластьЗаголовок);

        Шапка.Параметры.Заполнить (Выборка);
        ТабДок.Вывести (Шапка, Выборка.Уровень ());

        ВставлятьРазделительСтраниц = Истина;
    КонецЦикла;
КонецПроцедуры

```

Листинг Б.17 – Код модуля формы документа «Счет»

```

&НаКлиенте
Процедура ФормаОплатыПриИзменении (Элемент)
    Если Объект.ФормаОплаты =
        ПредопределенноеЗначение ("Перечисление.ФормыОплаты.Безналичная")
        Тогда

            Элементы.БанковскийСчет.Видимость = Истина;

        КонецЕсли;

```

КонецПроцедуры

Листинг Б.18 – Код модуля объекта документа «СчетНаВозврат»

Процедура ОбработкаЗаполнения (ДанныеЗаполнения,
СтандартнаяОбработка)

```

    Если ТипЗнч (ДанныеЗаполнения) =
Тип ("ДокументСсылка.Выбытие") Тогда
        Клиент = ДанныеЗаполнения.Клиент;
        ДокументОснование = ДанныеЗаполнения.Ссылка;
        СуммаВозврата = ДанныеЗаполнения.СуммаКВозврату;
    КонецЕсли;
КонецПроцедуры

```

Листинг Б.19 – Код модуля менеджера документа «СчетНаВозврат»

Процедура Печать (ТабДок, Ссылка) Экспорт

```

    Макет = Документы.СчетНаВозврат.ПолучитьМакет ("Печать");
    Запрос = Новый Запрос;
    Запрос.Текст =
"ВЫБРАТЬ
|     СчетНаВозврат.БанковскийСчет,
|     СчетНаВозврат.Дата,
|     СчетНаВозврат.ДокументОснование,
|     СчетНаВозврат.Клиент,
|     СчетНаВозврат.Номер,
|     СчетНаВозврат.СуммаВозврата,
|     СчетНаВозврат.ФормаОплаты
| ИЗ
|     Документ.СчетНаВозврат КАК СчетНаВозврат
| ГДЕ
|     СчетНаВозврат.Ссылка В (&Ссылка)";
    Запрос.Параметры.Вставить ("Ссылка", Ссылка);
    Выборка = Запрос.Выполнить ().Выбрать ();

```

```

    ОбластьЗаголовков = Макет.ПолучитьОбласть ("Заголовков");
    Шапка = Макет.ПолучитьОбласть ("Шапка");
    ТабДок.Очистить ();

```

```

    ВставлятьРазделительСтраниц = Ложь;
    Пока Выборка.Следующий () Цикл

```

```

Если ВставлятьРазделительСтраниц Тогда
    ТабДок.ВывестиГоризонтальныйРазделительСтраниц ( ) ;
КонецЕсли;

ТабДок.Вывести (ОбластьЗаголовков) ;

Шапка.Параметры.Заполнить (Выборка) ;
ТабДок.Вывести (Шапка, Выборка.Уровень ( ) ) ;

ВставлятьРазделительСтраниц = Истина; КонецЦикла;
КонецПроцедуры

```

Листинг Б.20 – Код модуля формы документа «СчетНаВозврат»

```

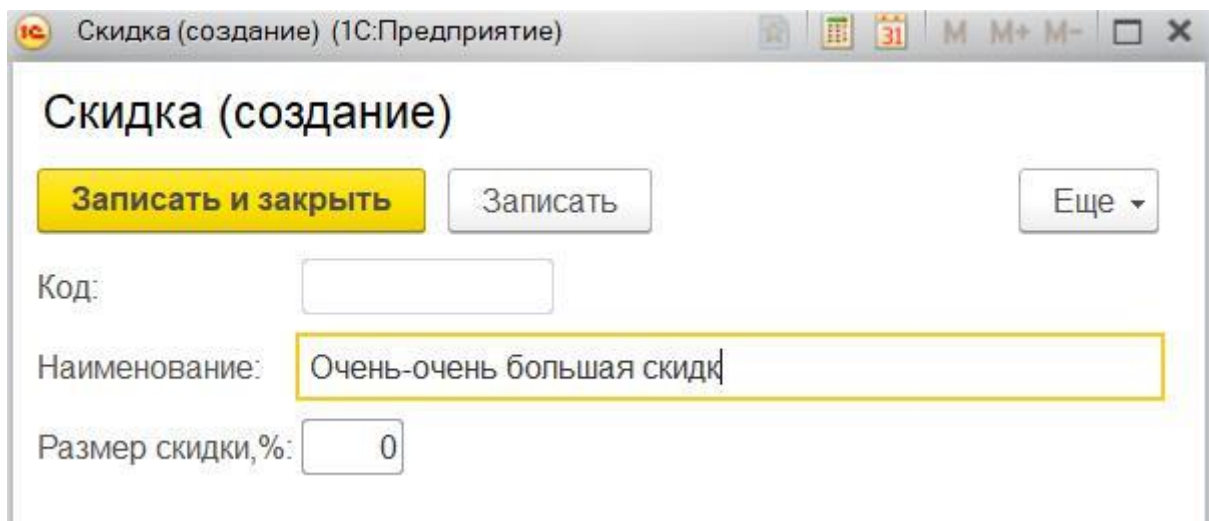
&НаКлиенте
Процедура
    ФормаОплатыПриИзменении (Элемент) Если
        Объект.ФормаОплаты =
        ПредопределенноеЗначение ("Перечисление.ФормыОплаты.Безналичная") Тогда

            Элементы.БанковскийСчет.Видимость =

                Истина; КонецЕсли;
КонецПроцедуры

```

Приложение В
РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ
(рекомендуемое)



Скидка (создание) (1С:Предприятие)

Скидка (создание)

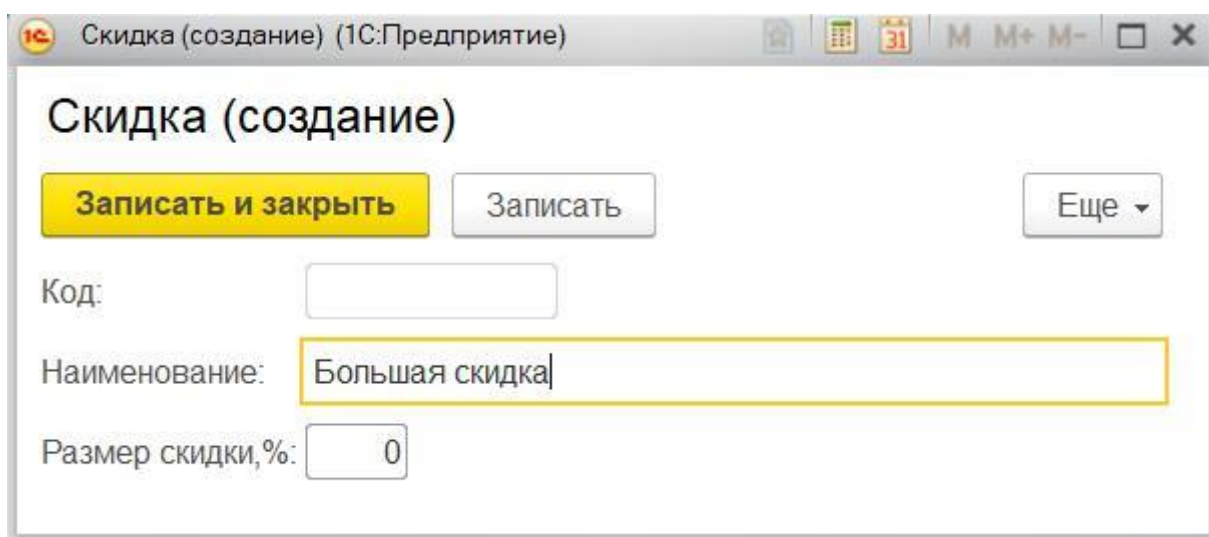
Записать и закрыть Записать Еще ▾

Код:

Наименование:

Размер скидки, %:

Рисунок В.1 – Результат функционального тестирования поля ввода
«Наименование» (тест 1)



Скидка (создание) (1С:Предприятие)

Скидка (создание)

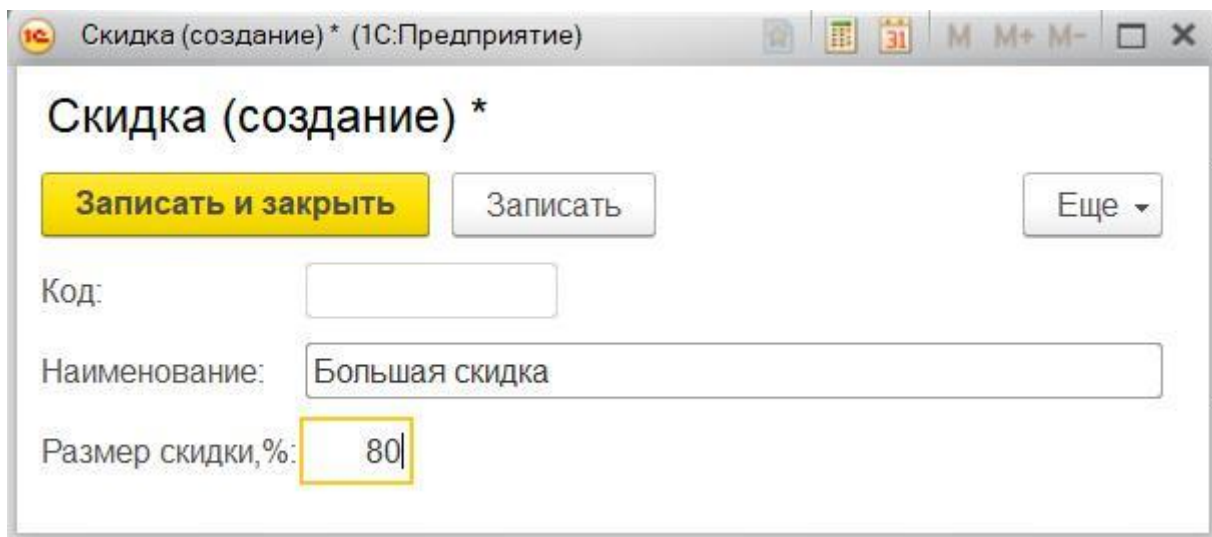
Записать и закрыть Записать Еще ▾

Код:

Наименование:

Размер скидки, %:

Рисунок В.2 – Результат функционального тестирования поля ввода
«Наименование» (тест 2)



Скидка (создание) * (1С:Предприятие)

Скидка (создание) *

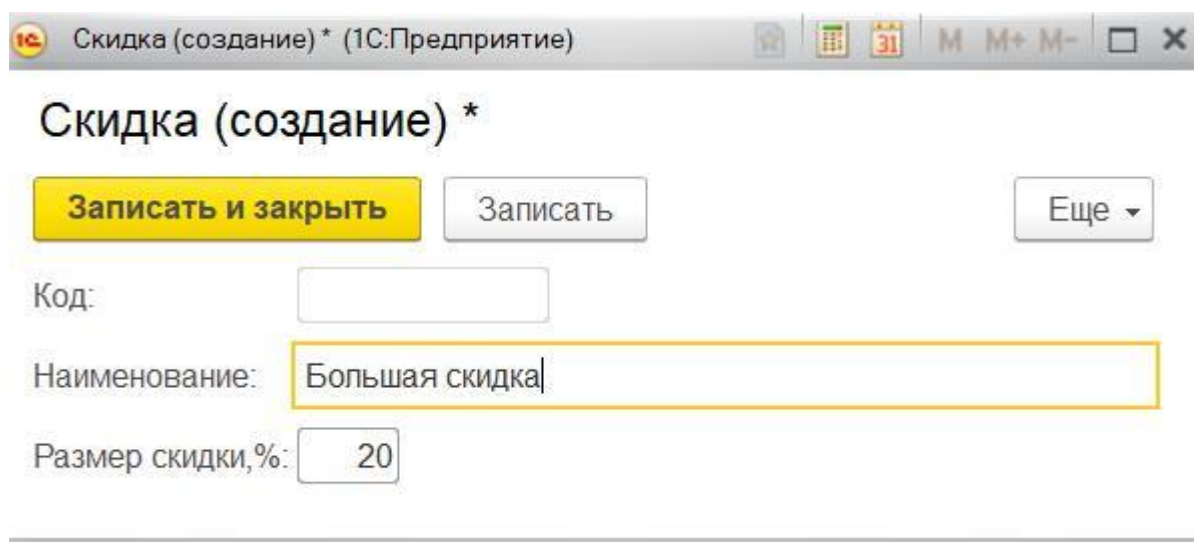
Записать и закрыть Записать Еще ▾

Код:

Наименование:

Размер скидки, %:

Рисунок В.3 – Результат функционального тестирования поля ввода
«Размер скидки» (тест 1)



Скидка (создание) * (1С:Предприятие)

Скидка (создание) *

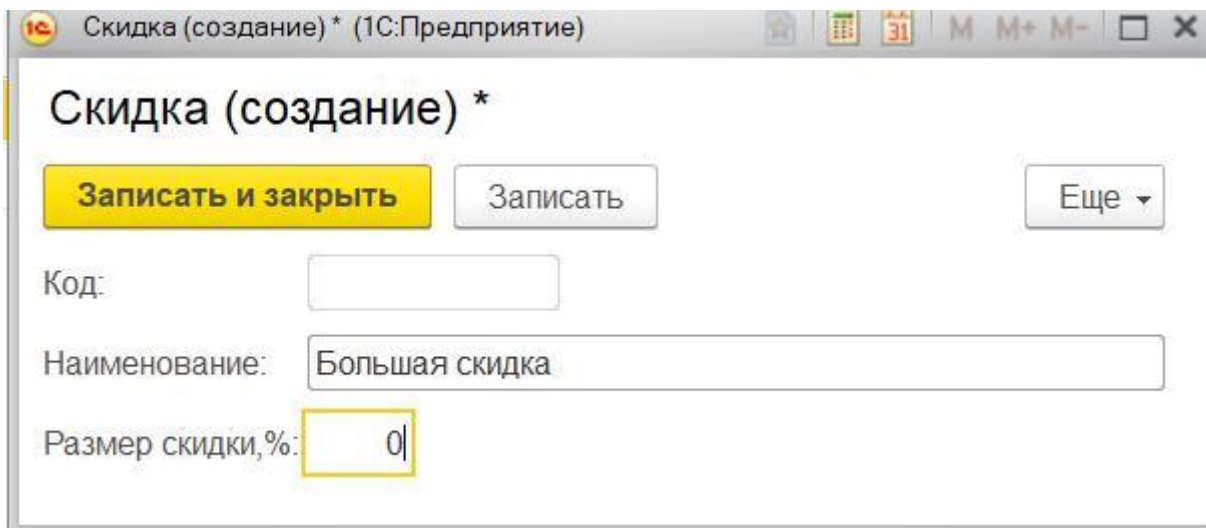
Записать и закрыть Записать Еще ▾

Код:

Наименование:

Размер скидки, %:

Рисунок В.4 – Результат граничного тестирования поля ввода
«Размер скидки» (тест 2)



Скидка (создание) * (1С:Предприятие)

Скидка (создание) *

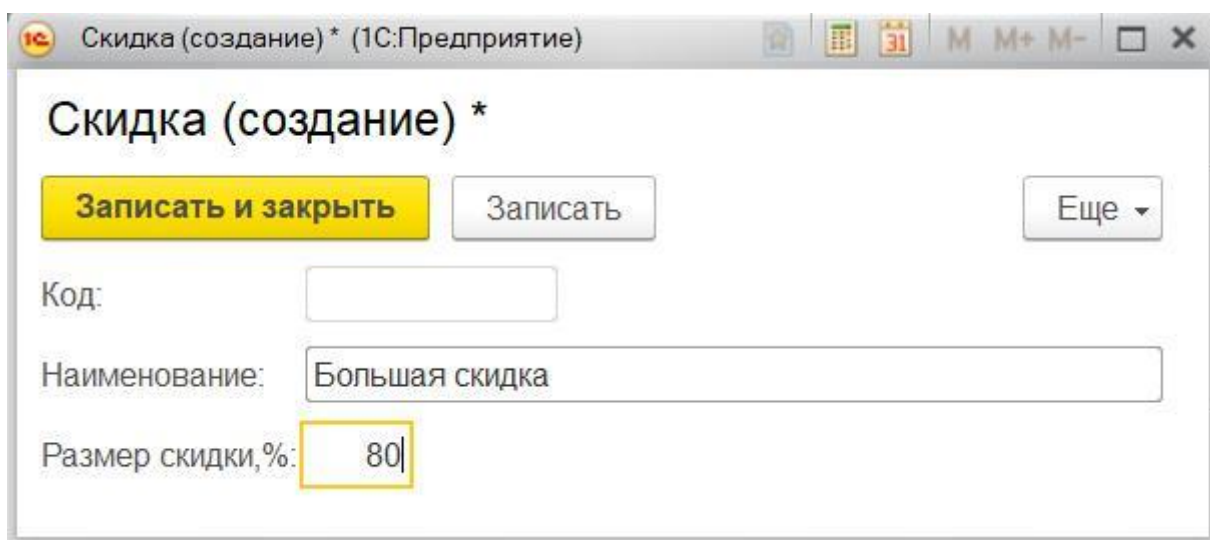
Записать и закрыть Записать Еще ▾

Код:

Наименование:

Размер скидки, %:

Рисунок В.5 – Результат граничного тестирования поля ввода
«Размер скидки» (тест 3)



Скидка (создание) * (1С:Предприятие)

Скидка (создание) *

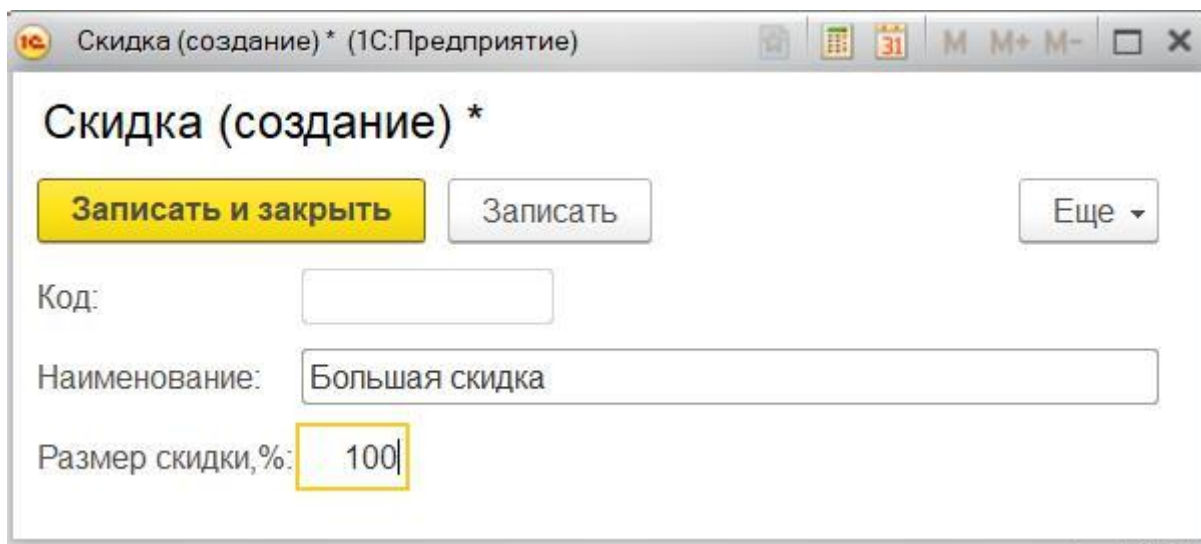
Записать и закрыть Записать Еще ▾

Код:

Наименование:

Размер скидки, %:

Рисунок В.6 – Результат граничного тестирования поля ввода
«Размер скидки» (тест 4)



Скидка (создание) * (1С:Предприятие)

Скидка (создание) *

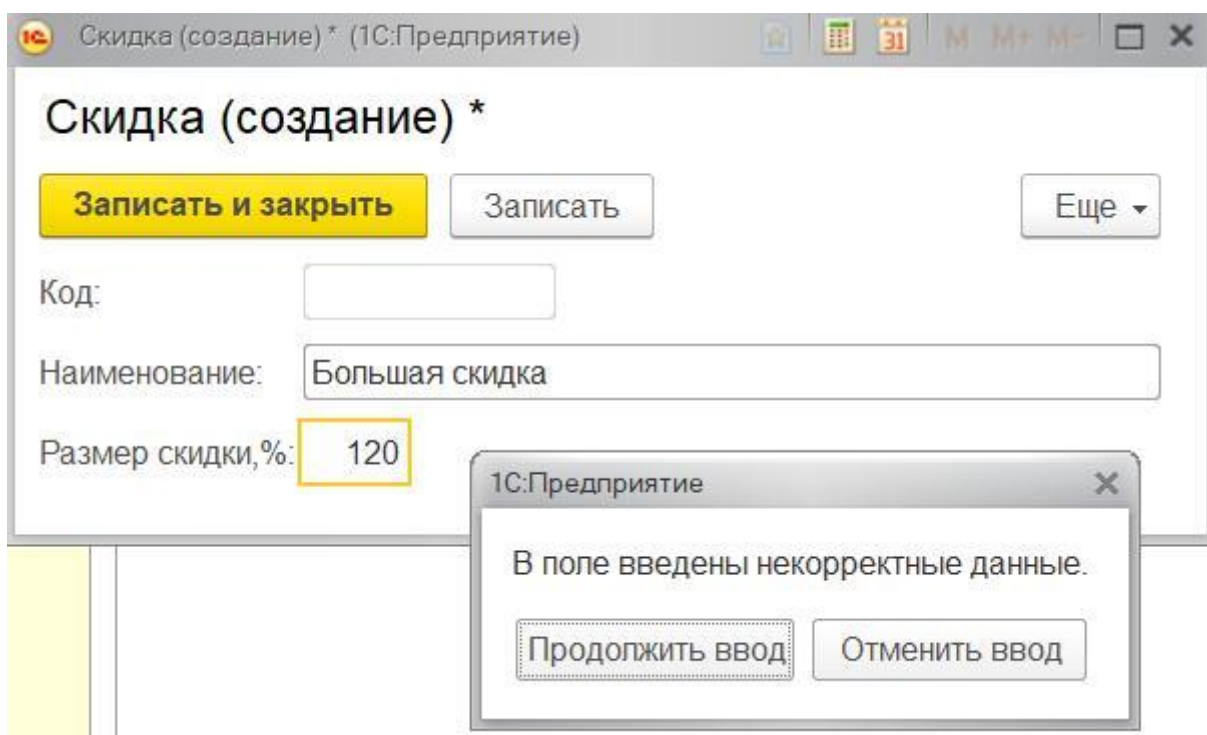
Записать и закрыть Записать Еще ▾

Код:

Наименование: Большая скидка

Размер скидки, %: 100

Рисунок В.7 – Результат граничного тестирования поля ввода
«Размер скидки» (тест 5)



Скидка (создание) * (1С:Предприятие)

Скидка (создание) *

Записать и закрыть Записать Еще ▾

Код:

Наименование: Большая скидка

Размер скидки, %: 120

1С:Предприятие

В поле введены некорректные данные.

Продолжить ввод Отменить ввод

Рисунок В.8 – Результат граничного тестирования поля ввода
«Размер скидки» (тест 6)

Скидка (создание) * (1С:Предприятие)

Скидка (создание) *

Записать и закрыть Записать Еще ▾

Код:

Наименование:

Размер скидки, %:

Рисунок В.9 – Результат граничного тестирования поля ввода
«Размер скидки» (тест 7)