

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ АРХИТЕКТУРНО-СТРОИТЕЛЬНЫЙ УНИВЕРСИТЕТ

Кафедра информационных систем  
и технологий в строительстве

## **Расчетно – графическая работа**

### **Автоматизированная информационная система «Салон красоты»**

Выполнил: студент 3 курса 1ИС02  
Егорова Д.А.

Проверил: доцент, кандидат технических наук  
Галимов Р.А.

Казань, 2023

## Содержание

Введение.....	3
1. Предварительный анализ .....	6
2. Описание программного продукта.....	7
2.1 Входная информация. ....	7
2.2 Выходная информация. ....	8
3. Техническое задание и выбор средств реализации .....	10
4. Описание модулей серверной части программы и их взаимодействие. ....	15
5. Состав и содержание работ по созданию системы.....	18
6. Порядок контроля и приёмки системы .....	19
7. Требования к документированию .....	20
8. Рекомендации по освоению .....	23
9. Код и Базы данных .....	24
Заключение .....	34
Список используемой литературы .....	35

## Введение

Данная работа включает в себя разработку модели бизнес-процесса и разработку клиентского приложения по учету услуг предоставляемых салоном красоты. Итог работы заключается в том что на основе первоначальной модели, будет создана модель «как нужно», а на ее основе спроектирована информационная система и клиентское приложение. В городах сотни салонов красоты и многие из них продолжают работать по старинке, в данной работе будет предложено простое решение по оптимизации существующих бизнес-процессов, и облегчения работы персонала. При желании в клиентское приложение может быть внедрен дополнительный функционал.

Объектом исследования является салон красоты. Предметом исследования является профессиональная деятельность коммерческой организации.

Цель работы: оптимизация существующих бизнес-процессов, показать процесс проектирования и разработки ИС.

Описание предметной области. Постановка задачи.

Салон красоты осуществляет свою деятельность сравнительно недавно, учет услуг предоставляемых салоном в данный момент ведется на бумажных носителях, есть список телефонов клиентов, хранящихся в файле формата xls.

Администратор ведет прием звонков, осуществляет запись клиентов на определенное время к определенному специалисту. Также в обязанности администратора входит расчет итоговой цены с учетом скидок и акций для каждого конкретного клиента, а также прием денежных средств, после предоставления услуг. Еще администратор осуществляет обзвон клиентов, так называемые «холодные звонки». В

конце отчетного периода руководитель салона подсчитывает сумму и количество услуг оказанных специалистами, и передает эти сведения бухгалтеру для расчета заработной платы каждого из специалистов.

Также необходимо фиксировать объем оказанных услуг в разрезе клиентов, для применения системы лояльности постоянным клиентам.

Исходные данные вносятся администратором салона, перед запуском системы мы можем использовать уже имеющуюся оцифрованную информацию о клиентах и их номерах телефонов, загрузив ее в качестве начального наполнения ИБ.

Перечень исходных данных:

1. Заполнение карточки клиента(ФИО, дата рождения, телефон)
2. Заполнение перечня оказываемых услуг(наименование услуги, стоимость)
3. Заполнение документа «Оказание услуг»(Специалист, список оказанных услуг)
4. Заполнение списка дисконтных карт(номер карты, размер скидки)
5. Заполнение списка специалистов
6. Заполнение расписания специалистов

Перечень выходных данных:

1. Отчет по суммам и количеству оказанных услуг каждым специалистом за период
  2. Отчет по сумма и количеству оказанных услуг каждому клиенту за период
  3. Отчет по наиболее востребованным специалистам за период
  4. Отчет по наиболее востребованным услугам за период
  5. Отчет о предоставленных скидках за период
- Выбор программного обеспечения

Для реализации данного проекта будет выбрана среда разработки «Python».

## **1. ПРЕДВАРИТЕЛЬНЫЙ АНАЛИЗ**

В моем курсовом проекте в качестве исследуемой организации выступает Салон красоты.

В нем рассматривается, какие услуги предоставляются в салоне, какие мастера обслуживают в той или иной услуге, сведения о клиентах и сотрудниках салона.

Задачей является разработать приложение салона красоты, позволяющую вести учет работы салона, а именно учет клиентов посещающих салон, сведения об сотрудниках, стоимость услуг салона, какие заказы поступают в салон и должностные инструкции сотрудников.

## **2. Описание программного продукта**

### **2.1 Входная информация.**

Подраздел «Входная информация» содержит перечень и описание входных сообщений. Различается и описывается отдельно входная информация, используемая для заполнения таблиц базы данных, и входная информация, вводимая пользователем для поиска требуемых данных.

Запросы в Python позволяют управлять данными: добавлять, удалять, изменять.

Для построения запросов Python предоставляет очень удобный инструмент - сетку запросов, которая позволяет не писать запрос, а "сконструировать" его.

Сетка запросов позволяет:

- выбрать поля, которые надо отобразить;
- добавить критерий отбора (по каким значениям полей получить записи);

Запросы в Python – объекты базы данных, извлекающие из таблиц или других запросов информацию согласно заданным условиям. Это виртуальные таблицы (существующие в оперативной памяти компьютера).

Для этого в Python предусмотрена возможность ввода параметров запросов непосредственно перед их исполнением. Чтобы установить ввод параметра вместо конкретного значения, в строке "Условие отбора" вводятся имя или фраза, заключенные в квадратные скобки. Все, введенное в квадратных скобках Python рассматривает как имя параметра, и выводится в диалоговом окне запроса как комментарий в вводимому параметру. В запросе можно использовать несколько параметров, но с уникальными именами.

Запросы на выборку также полезны, если нужно извлечь данные из нескольких связанных таблиц, сгруппировать данные или вычислить итоги.

С помощью запроса на выборку можно получить только нужные данные и отобразить их в режиме таблицы.

Запросы на изменение - запрос, который за одну операцию изменяет или перемещает несколько записей. Существует четыре типа запросов на изменение:

- Запрос на удаление;
- Запрос на обновление записей;
- Запрос на добавление;
- Запрос на создание таблицы;

## **2.2 Выходная информация.**

Подраздел «Выходная информация» содержит перечень и описание выходных сообщений (экранных форм, документов и пр.)

В подразделе «Выбор и обоснование выбора используемого программного обеспечения» выбирается программное обеспечение, которое будет использоваться при работе над курсовым проектом.

В программе Python можно создавать различные типы отчетов, отвечающие разнообразным требованиям. Существует несколько способов создания отчета Python:

Автоотчет - Функция «Автоотчет» позволяет быстро создать отчет, включающий все сведения заданной таблицы или данные, извлеченные в результате запроса. В качестве основы для отчета можно использовать таблицу или результат запроса.

Мастер отчетов - Мастер отчетов позволяет быстро создать отчет на основе данных нескольких таблиц или запросов. В качестве основы для отчета можно использовать сочетание таблиц и запросов.

Режим конструктора . С одной стороны, создание отчетов в режиме конструктора занимает больше времени, чем создание отчета с помощью мастера отчетов или функции «Автоотчет», но с другой стороны, отчеты, созданные в режиме конструктора, можно настроить таким образом, чтобы



они максимально отвечали требованиям пользователя. Например, можно сортировать и фильтровать данные более подробно, нежели это позволяет мастер отчетов. Режим конструктора также открывает больше возможностей для настройки внешнего вида отчета.

### **3. Техническое задание и выбор средств реализации**

Для реализации функционала приложения салона красоты, необходимо определить какие функции будет выполнять приложение.

Главные функции приложения:

- Информационная функция – главная цель этой функции — это предоставление информации про салон пользователю, а именно отображение описания приложения, отображение информации про салон красоты, отображение работ мастеров, которые работают в салоне красоты, отображение отзывов клиентов салона и отображение каталога услуг, которые предоставляет салон красоты. Эта функция является частью рекламной функции и помогает пользователям узнавать необходимую информацию про салон.

- Административная функция – главная цель этой функции – это снижение рабочей нагрузки на менеджеров салона, посредством принятия заказов на услуги, формирования графика мастеров и упрощение работы салона, реализация этой функции даст возможность салону принимать заказы на услуги целые сутки и по выходным, поскольку приложение работает круглосуточно, и пользователи смогут отправлять запросы онлайн.

- Рекламная функция - главная цель этой функции заключается в распространении информации и увеличению количества клиентов салона красоты. Рекламная функция способствует развитию бизнеса, что способствует развитию сферы косметологических услуг в стране. Рекламная функция приложения тесно связана с информационной функцией. Информационная функция является частью рекламной функции.

Технические требования к разделам приложения:

Для того чтобы полностью реализовать все три функции приложения салона красоты необходимо реализовать такие разделы на приложение:

- Главный раздел - описывает детально салон красоты и содержит ссылки на остальные разделы.

- Раздел «О нас» - описывает детально салон красоты.
- Раздел «Цены» - содержит цены на услуги, которые предоставляет салон красоты.
- Раздел «Отзывы» - содержит отзывы клиентов, которые воспользовались услугами салона красоты.
- Раздел «Наши работы» - содержит информацию про мастеров, а также фотографии работ, которые были выполнены мастерами салона.
- Раздел «Как связаться» - содержит информацию про то как связаться салоном красоты, контакты, адрес, ссылки на социальные сети.

Для создания приложения использовались такие технологии как:

- Язык Python

С целью определения списка страниц и разделов для реализации приложения необходимо произвести декомпозицию бизнес-процессов деятельности приложения салона красоты и определить, какие именно активности будут реализованы в работе и их особенности.

Для того, чтобы произвести декомпозицию бизнес-процесса необходимо его детально проанализировать. Для этого сделаем рисунок дерева функций, где рассмотрим бизнес-процесс системы.

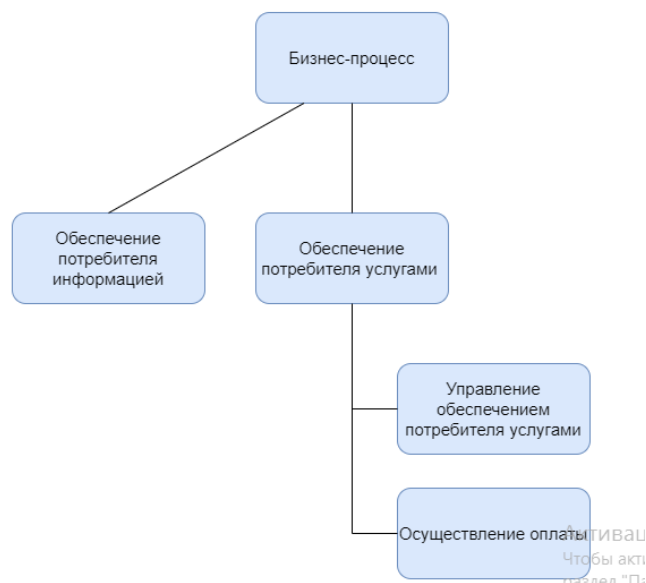


Рис. 1 - Дерево функций

Приложение должно содержать реализацию таких страниц:

- Главная страница приложения
- Страница «Оформление заказа»
- Страница раздела «Отзывы»
- Страница раздела «Цены»
- Страница «О нас»
- Страница каталога
- Страница «Контакты»

Детально структура страниц описана на рисунке структуры клиентской части приложения, изображенном ниже.

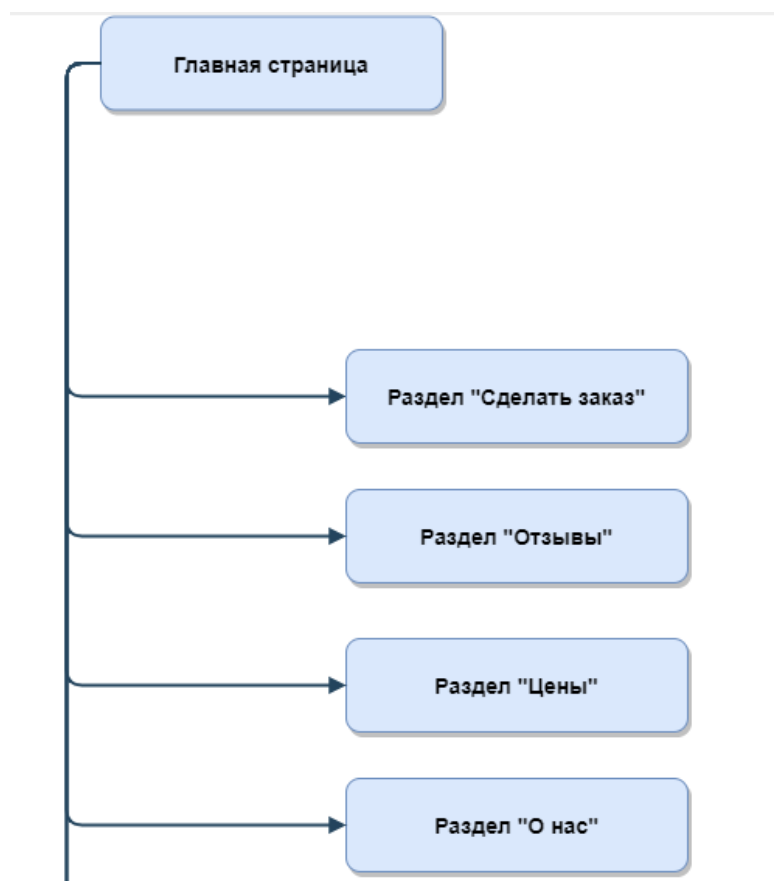


Рис. 2 - Структура клиентской части приложения

Контекстная диаграмма - процесс Салона красоты, взаимодействие его с внешней средой. Что управляет процессом, какая информация поступает, какая выходит, механизмы, связанные с выполнением операции, мы можем видеть на рисунке 3.

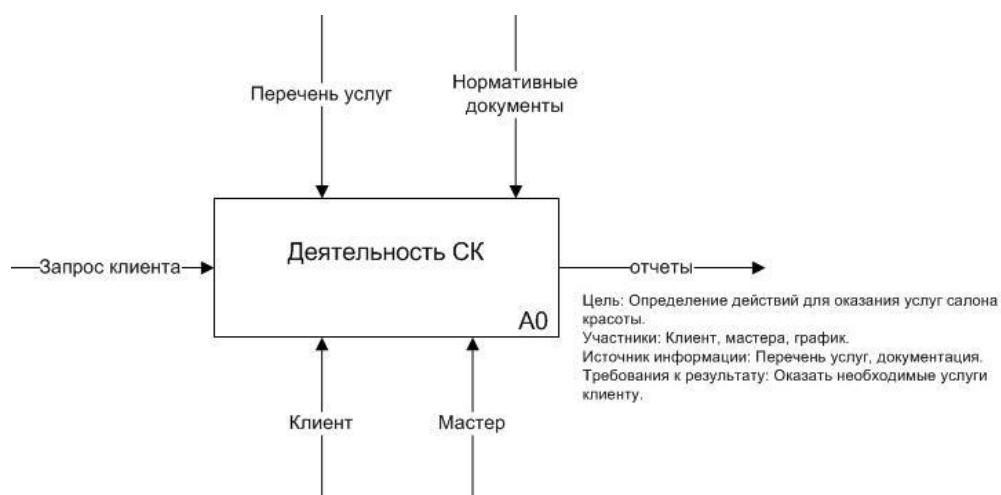


Рисунок 3 - Контекстная диаграмма – «Деятельность организации»

Первая декомпозиция - разбиение контекстной диаграммы на несколько составляющих (Рисунок 4), демонстрируется, какие процессы протекают внутри организации. Показано как внешняя среда действует на организацию изнутри, что с чем связано, а так же что из чего получается.

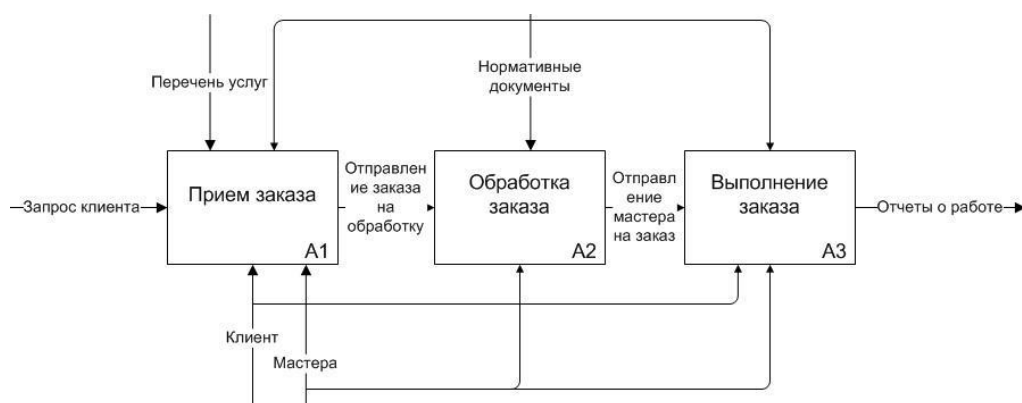


Рисунок 4 – Первая декомпозиция

Вторая декомпозиция - более детальная разбивка каждого блока первой декомпозиции. То есть более глубокое рассмотрение конкретного процесса. На Рисунке 5 мы видим, как внешняя среда взаимодействует процессами, которые протекают на уровне «Прием заказа».

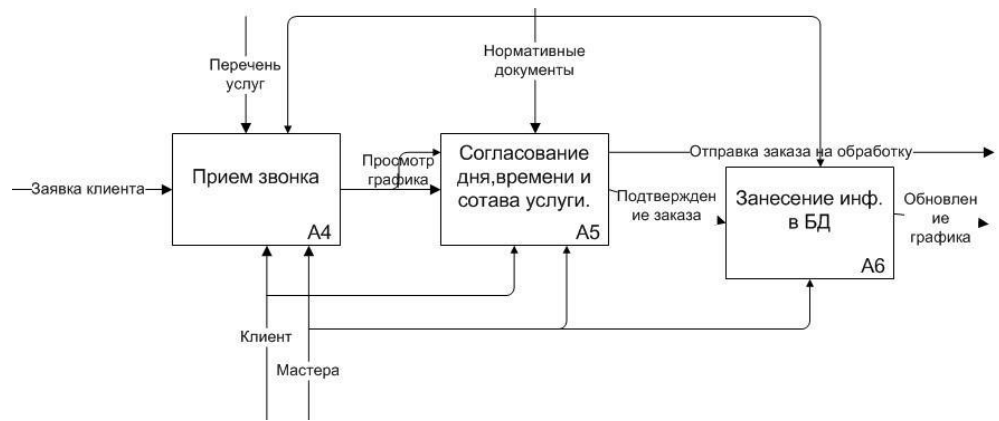


Рисунок 5 – Вторая декомпозиция – «Приём заказа»

#### **4. Описание модулей серверной части программы и их взаимодействие.**

Для серверной части приложения салона красоты необходимо понять и описать структуру взаимодействия серверной и клиентской частей приложения. Эта структура взаимодействия представляет из себя структуру взаимодействия веб-браузеров (клиентская часть, с которой взаимодействуют пользователи) с веб-серверами (серверная часть приложения), серверная часть приложения уже взаимодействует с базой данных.

Веб-браузеры взаимодействуют с веб-серверами при помощи гипертекстового транспортного протокола HTTP. Когда пользователь нажимает на ссылку на странице, то это приводит к заполнению формы и отправки запроса с данными из браузера к серверу. Запрос включает в себя URL, определяющий затронутый ресурс, метод, определяющий требуемое действие и может включать дополнительную информацию, закодированную в параметрах URL, POST запрос.

Серверы ожидают сообщений, отправленных с запросами со стороны клиентской части, обрабатывают их по прибытию и отвечают веб-браузеру при помощи ответного HTTP сообщения. Ответ содержит строку состояния, показывающую, был ли запрос успешным, или нет.

Тело успешного ответа на запрос может содержать запрашиваемые данные (например, новую HTML страницу, или изображение), который может отображаться через веб-браузер.

На схеме серверной части приложения салона красоты видно, как происходит взаимодействие серверной части и клиентской части: веб-сервер извлекает запрошенный документ из своей файловой системы и возвращает HTTP-ответ, содержащий документ и успешный статус (обычно 200 OK). Если файл не может быть извлечен по каким-либо причинам, возвращается статус ошибки (смотри ошибки клиента и ошибки сервера).

Программирование серверной части – это важная часть разработки системы поскольку серверная часть веб-приложения позволяет эффективно доставлять информацию, составленную индивидуально для пользователей и таким образом позитивно повлиять на развитие бизнеса.

Рассмотрим далее детально программные модули, реализованные для исполнения функций приложения салона красоты.

Основные модули приложения это

- Основная процедура работы веб-приложения
- Форма «Оформление заказа»
- Форма «Цены»
- Форма «Каталог»
- Форма «Контакты»
- Форма «Отзывы»
- Форма «О нас»
- Форма «Цены»

Детально взаимодействие модулей описано с помощью дерева вызова программных модулей на рисунке ниже.



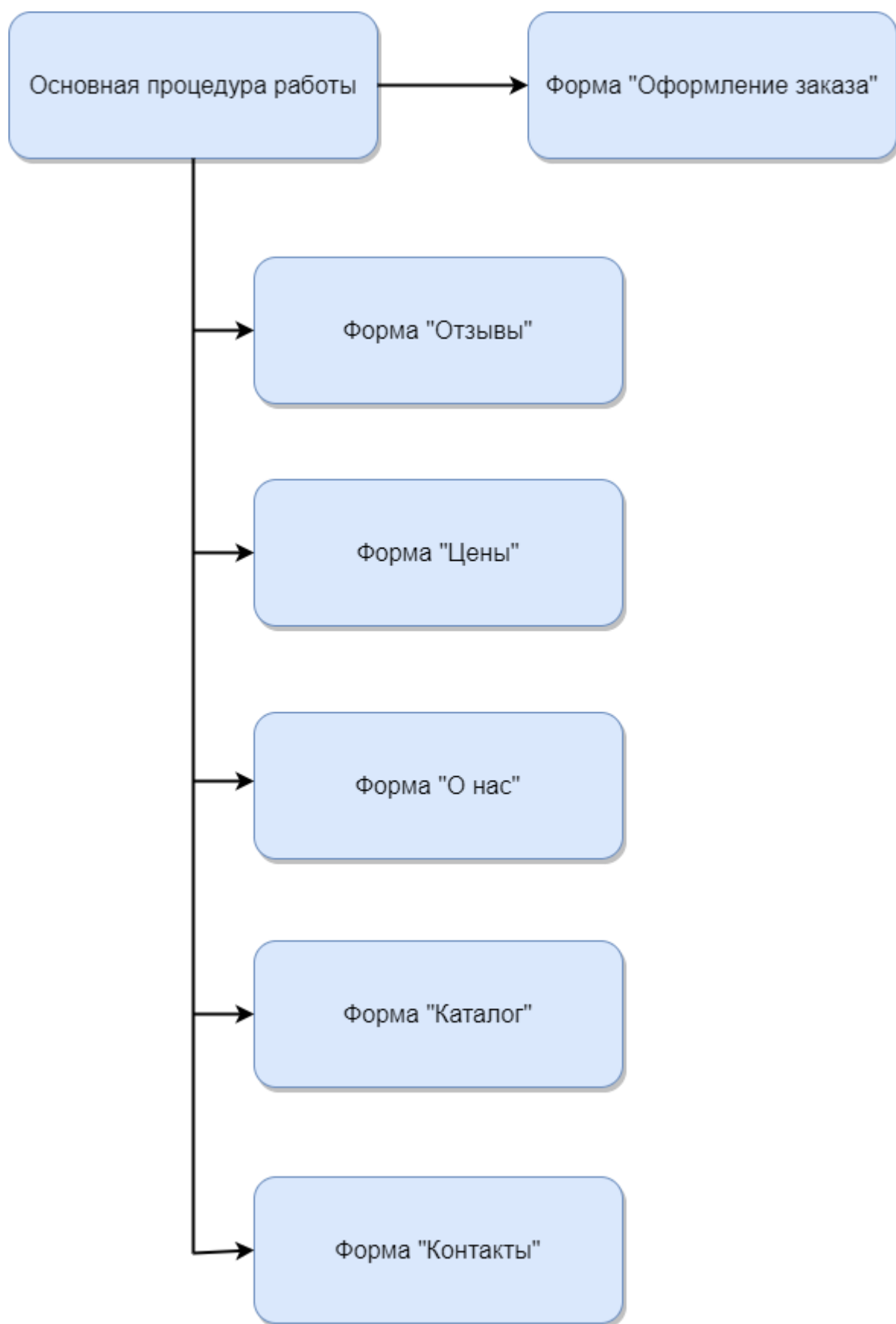


Рис. 4 - Дерево вызовов программных модулей

## **5. Состав и содержание работ по созданию системы**

Процесс создания АС представляет собой совокупность упорядоченных во времени, взаимосвязанных, объединенных в стадии этапы работ, выполнение которых необходимо и достаточно для создания АС, соответствующей заданным требованиям [из п. 1.1 ГОСТ 34.601-90]

Стадии и этапы создания АС выделяются как части процесса создания по соображениям рационального планирования и организации работ, заканчивающихся заданным результатом [из п. 1.2 ГОСТ 34.601-90]

Состав и правила выполнения работ на установленных настоящим стандартом стадиях и этапах определяют в соответствующей документации организаций, участвующих в создании конкретных видов АС.

## **6. Порядок контроля и приёмки системы**

Для АС устанавливают следующие основные виды испытаний:

1. предварительные;
2. опытная эксплуатация;
3. приемочные.

Примечания:

1. Допускается дополнительно проведение других видов испытаний АС и их частей;
2. Допускается классификация приемочных испытаний в зависимости от статуса приемочной комиссии (состав членов комиссии и уровень его утверждения);
3. Виды испытаний и статус приемочной комиссии устанавливают в договоре и (или) ТЗ.

## **7. Требования к документированию**

В ходе выполнения проекта на объекте автоматизации требуется выполнить работы по подготовке к вводу системы в действие. При подготовке к вводу в эксплуатацию АС Кадры Заказчик должен обеспечить выполнение следующих работ:

1. Определить подразделение и ответственных должностных лиц, ответственных за внедрение и проведение опытной эксплуатации;
2. Обеспечить присутствие пользователей на обучении работе с системой, проводимом Исполнителем;
3. Обеспечить соответствие помещений и рабочих мест пользователей системы в соответствии с требованиями, изложенными в настоящем ЧТЗ;
4. Обеспечить выполнение требований, предъявляемых к программно-техническим средствам, на которых должно быть развернуто программное обеспечение;
5. Совместно с Исполнителем подготовить план развертывания системы на технических средствах Заказчика;
6. Провести опытную эксплуатацию

Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие, включая перечень основных мероприятий и их исполнителей должны быть уточнены на стадии подготовки рабочей документации и по результатам опытной эксплуатации.

- Состав и содержание работ по созданию системы

Процесс создания АС представляет собой совокупность упорядоченных во времени, взаимосвязанных, объединенных в стадии этапы работ, выполнение которых необходимо и достаточно для создания АС, соответствующей заданным требованиям [из п. 1.1 ГОСТ 34.601-90]

Стадии и этапы создания АС выделяются как части процесса создания по соображениям рационального планирования и организации работ, заканчивающихся заданным результатом [из п. 1.2 ГОСТ 34.601-90]

Состав и правила выполнения работ на установленных настоящим стандартом стадиях и этапах определяют в соответствующей документации организаций, участвующих в создании конкретных видов АС. Перечень организаций, участвующих в работах по созданию АС, приведен в приложении 2 [из п. 1.4 ГОСТ 34.601-90]

- Порядок контроля и приёмки системы

Для АС устанавливают следующие основные виды испытаний:

4. предварительные;
5. опытная эксплуатация;
6. приемочные.

Примечания:

4. Допускается дополнительно проведение других видов испытаний АС и их частей;

5. Допускается классификация приемочных испытаний в зависимости от статуса приемочной комиссии (состав членов комиссии и уровень его утверждения);

6. Виды испытаний и статус приемочной комиссии устанавливают в договоре и (или) ТЗ.

- Требования к документированию

В ходе выполнения проекта на объекте автоматизации требуется выполнить работы по подготовке к вводу системы в действие. При

подготовке к вводу в эксплуатацию АС Кадры Заказчик должен обеспечить выполнение следующих работ:

7. Определить подразделение и ответственных должностных лиц, ответственных за внедрение и проведение опытной эксплуатации;
8. Обеспечить присутствие пользователей на обучении работе с системой, проводимом Исполнителем;
9. Обеспечить соответствие помещений и рабочих мест пользователей системы в соответствии с требованиями, изложенными в настоящем ЧТЗ;
10. Обеспечить выполнение требований, предъявляемых к программно-техническим средствам, на которых должно быть развернуто программное обеспечение;
11. Совместно с Исполнителем подготовить план развертывания системы на технических средствах Заказчика;
12. Провести опытную эксплуатацию

Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие, включая перечень основных мероприятий и их исполнителей должны быть уточнены на стадии подготовки рабочей документации и по результатам опытной эксплуатации.

## **8. Рекомендации по освоению**

Для освоения необходимо уметь пользоваться компьютером, на базовом уровне и прочитать всё руководство пользователя и только после этого приступать непосредственно за работу с системой.

Необходимо провести тестирование персонала на знание руководства.

При возникновении вопросов в процессе работы рекомендуется найти по оглавлению нужный раздел Руководства пользователя и изучить его.

Необходимо быть аккуратным в системе, и прислушиваться к руководству пользователя.

## 9. Код и Базы данных

- **Пояснение к коду**

Абстрактные классы и методы:

Реализован абстрактный класс `Person` с абстрактным методом `display_info`.

Класс `Customer` является подклассом `Person` и реализует метод `display_info`, реализация абстрактного метода.

Статические свойства и методы:

В классе `Customer` статическое свойство `discount_rate` скидку для всех клиентов.

Метод `set_discount_rate` класса `Customer` является статическим и позволяет установить скидку для всех клиентов.

Композиция или агрегация:

Класс `BeautySalon` содержит список объектов класса `Customer`. Это можно рассматривать как пример агрегации, где `BeautySalon` является владельцем объектов `Customer`.

Перегрузка методов:

В классе `Customer` перегружен метод `apply_discount` для учета скидки при расчете стоимости услуги.

Динамические свойства:

Хранение данных в таблицах Excel и их выгрузка:

`to_excel()`;

Класс `BeautySalon` имеет методы `load_customers_data` и `save_customers_data`, которые загружают данные о клиентах из файла Excel и сохраняют их в файл Excel соответственно.



- **Код**

```
from abc import ABC, abstractmethod
```

```
from openpyxl import Workbook, load_workbook
```

```
class Person(ABC):
```

```
    def __init__(self, name, phone):
```

```
        self.name = name
```

```
        self.phone = phone
```

```
    @abstractmethod
```

```
    def display_info(self):
```

```
        pass
```

```
class Customer(Person):
```

```
    discount_rate = 0.1 # Статическое свойство для всех клиентов
```

```
    def __init__(self, name, phone, loyalty_points=0):
```

```
        super().__init__(name, phone)
```

```
        self.loyalty_points = loyalty_points
```

```
    def display_info(self):
```

```
        print(f"Клиент: {self.name} - Телефон: {self.phone} - Баллы  
        лояльности: {self.loyalty_points}")
```

```
    @staticmethod
```

```
    def set_discount_rate(rate):
```

```
        Customer.discount_rate = rate
```

```
    def earn_loyalty_points(self, points):
```

```
        self.loyalty_points += points
```

```
    def apply_discount(self, amount):
```

```
return amount * (1 - Customer.discount_rate)
```

```
class BeautyService:
```

```
    def __init__(self, name, price):
```

```
        self.name = name
```

```
        self.price = price
```

```
    def display_info(self):
```

```
        print(f"Услуга: {self.name} - Цена: ${self.price}")
```

```
class BeautySalon:
```

```
    def __init__(self, name, customers_file="customers.xlsx"):
```

```
        self.name = name
```

```
        self.customers = []
```

```
        self.customers_file = customers_file
```

```
        # Загрузка данных о клиентах из файла Excel
```

```
        self.load_customers_data()
```

```
    def load_customers_data(self):
```

```
        try:
```

```
            workbook = load_workbook(self.customers_file)
```

```
            sheet = workbook.active
```

```
            for row in sheet.iter_rows(min_row=2, values_only=True):
```

```
                name, phone, loyalty_points = row
```

```
                customer = Customer(name, phone, loyalty_points)
```

```
                self.customers.append(customer)
```

```

        workbook.close()

    except FileNotFoundError:

        pass

def save_customers_data(self):

    workbook = load_workbook(self.customers_file)

    sheet = workbook.active


    for customer in self.customers:

        sheet.append([customer.name, customer.phone,
customer.loyalty_points])

        workbook.save(self.customers_file)

        workbook.close()


def add_customer(self, customer):

    self.customers.append(customer)

def display_customer_services(self):

    workbook = load_workbook("beauty_salon_data.xlsx")

    sheet = workbook.active

    for row in sheet.iter_rows(min_row=1, values_only=True):

        print(f"Имя: {row[0]} Услуга: {row[1]} цена: {row[2]}")

    workbook.close()


def display_customers(self):

    print(f"Клиенты салона '{self.name}':")

    for customer in self.customers:

        customer.display_info()

```

```

def record_appointment(self, customer, service):

    workbook = load_workbook("beauty_salon_data.xlsx")

    sheet = workbook.active

    print(f"Клиент {customer.name} записан на услугу: {service}")

    customer.earn_loyalty_points(5) # Например, начисление баллов за
запись на услугу

    print(f"{customer.name} получил 5 баллов лояльности.")

    discounted_price = customer.apply_discount(service.price)

    print(f"С учетом скидки, клиент заплатит: ${discounted_price}")

    sheet.append([customer.name, service.name, discounted_price])

    workbook.save("beauty_salon_data.xlsx")

```

```

def set_discount_rate(self, rate):

    Customer.set_discount_rate(rate)

    print(f"Скидка для всех клиентов салона '{self.name}' установлена на
{rate * 100}%.")

```

```

def main():

```

```

    salon_name = input("Введите название салона красоты: ")

```

```

    beauty_salon = BeautySalon(salon_name)

```

```

while True:

```

```

    print("\n1. Добавить клиента\n2. Показать клиентов\n3. Записать на
услугу\n")

```

```

    "4. Установить скидку\n5. Показать клиентов и услуги\n6.
Сохранить и выйти")

```

```

    choice = input("Введите номер действия: ")

```

```

if choice == '1':
    name = input("Введите имя клиента: ")
    phone = input("Введите номер телефона клиента: ")
    customer = Customer(name, phone)
    beauty_salon.add_customer(customer)
    print(f"{name} добавлен в систему.")
    beauty_salon.save_customers_data()

elif choice == '2':
    beauty_salon.display_customers()

elif choice == '3':
    beauty_salon.display_customers()

    customer_name = input("Введите имя клиента: ")
    service_name = input("Введите название услуги: ")
    service_price = float(input("Введите цену услуги: $"))
    service = BeautyService(service_name, service_price)

    customer = next((c for c in beauty_salon.customers if c.name ==
customer_name), None)

    if customer:
        beauty_salon.record_appointment(customer, service)
    else:
        print(f"Клиент {customer_name} не найден.")

elif choice == '4':
    discount_rate = float(input("Введите новую скидку в процентах: ")) /
100

    beauty_salon.set_discount_rate(discount_rate)

```

```

elif choice == '5':

    beauty_salon.display_customer_services()

elif choice == '6':

    # Сохранение данных о клиентах перед выходом

    print("Данные сохранены. До свидания!")

    break

else:

    print("Некорректный ввод. Попробуйте еще раз.")

if __name__ == "__main__":

    main()

```

### 1. Называем салон и добавляем клиента:

```

Введите название салона красоты: Даша

1. Добавить клиента
2. Показать клиентов
3. Записать на услугу
4. Установить скидку
5. Показать клиентов и услуги
6. Сохранить и выйти
Введите номер действия: 1
Введите имя клиента: Зия
Введите номер телефона клиента: 789
Зия добавлен в систему.

```

### 2. Выводим всех клиентов из базы данных салона:

```
1. Добавить клиента
2. Показать клиентов|
3. Записать на услугу
4. Установить скидку
5. Показать клиентов и услуги
6. Сохранить и выйти
Введите номер действия: 2
Клиенты салона 'Даша':
```

```
Клиент: ДАША - Телефон: 222 - Баллы лояльности: 10
Клиент: КАТЯ - Телефон: 888 - Баллы лояльности: 5
Клиент: Дима - Телефон: 777 - Баллы лояльности: 0
Клиент: Света - Телефон: 666 - Баллы лояльности: 0
Клиент: Оксана - Телефон: 333 - Баллы лояльности: 0
Клиент: Вероника - Телефон: 234 - Баллы лояльности: 0
Клиент: Гульшат - Телефон: 235 - Баллы лояльности: 0
Клиент: Марина - Телефон: 5830 - Баллы лояльности: 0
Клиент: Ирина - Телефон: 499 - Баллы лояльности: 0
Клиент: Яна - Телефон: 345 - Баллы лояльности: 0
Клиент: Зия - Телефон: 789 - Баллы лояльности: 0
```

### 3. Записываем клиента на услугу:

```
1. Добавить клиента
2. Показать клиентов
3. Записать на услугу
4. Установить скидку
5. Показать клиентов и услуги
6. Сохранить и выйти
Введите номер действия: 3
```

```
Введите имя клиента: Зия
Введите название услуги: педикюр
Введите цену услуги: $800
Клиент Зия записан на услугу: <__main__.BeautyService object at 0x0000013DC5799FF0>
Зия получил 5 баллов лояльности.
С учетом скидки, клиент заплатит: $720.0
```

#### 4. Устанавливаем скидку на услуги салона:

```
1. Добавить клиента
2. Показать клиентов
3. Записать на услугу
4. Установить скидку
5. Показать клиентов и услуги
6. Сохранить и выйти
Введите номер действия: 4
Введите новую скидку в процентах: 8
Скидка для всех клиентов салона 'Даша' установлена на 8.0%.
```

#### 5. Выводим клиентов, записавшихся на услуги:

```
1. Добавить клиента
2. Показать клиентов
3. Записать на услугу
4. Установить скидку
5. Показать клиентов и услуги
6. Сохранить и выйти
Введите номер действия: 5
Имя: ДАША Услуга: маникюр цена: 54
Имя: КАТЯ Услуга: шугаринг цена: 81
Имя: Зиля Услуга: педикюр цена: 720
```

#### 6. Выходим из программы:

```
1. Добавить клиента
2. Показать клиентов
3. Записать на услугу
4. Установить скидку
5. Показать клиентов и услуги
6. Сохранить и выйти
Введите номер действия: 6
Данные сохранены. До свидания!

Process finished with exit code 0
```



**В Базе данных следующие записи:**

beauty\_salon\_data.xlsx - Excel

Файл Главная Вставка Разметка с Формулы Данные Рецензиро

Буфер обмена Шрифт Выравнивание Число Условное формат Форматировать к Стили ячеек

Стили

G10

	A	B	C	D	E
1	ДАША	маникюр	54		
2	КАТЯ	шугаринг	81		
3	Зиля	педикюр	720		
4					

customers.xlsx - Excel

Файл Главная Вставка Разметка с Формулы Данные Рецензиро

Буфер обмена Шрифт Выравнивание Число Условное форм Форматировать Стили ячеек

Стили

6

	A	B	C	D	E
	Name	Phone	Loyalty Points		
	ДАША	222	10		
	КАТЯ	888	5		
	Дима	777	0		
	Света	666	0		
	Оксана	333	0		
	Вероника	234	0		
	Гульшат	235	0		
	Марина	5830	0		
	Ирина	499	0		
	Яна	345	0		
	Зиля	789	5		

## **Заключение**

В данной работе разобрано моделирование бизнес-процесса салона красоты, предложены пути оптимизации рабочего процесса, разработана модель «как должно быть». Показаны основные этапы разработки информационной системы, предложена платформа для разработки, выбор платформы мотивирован и обоснован, разобраны основные объекты метаданных, показаны интерфейс и логика работы с информационной системой. Применение данной информационной системы позволит избавиться от бумажной работы специалистов, оптимизирует работу администратора салона, облегчит руководителю работу по расчету заработной платы и позволит оперативно принимать управленческие решения.

Интерфейс программы не сложен в освоении и не требует каких-то специальных навыков. В случае внедрения, в проект легко можно добавить новые функциональные возможности.

## **Список используемой литературы**

1. Култыгин О.П. «Методы и средства проектирования информационных систем»
2. Радченко М.Г. «1С:Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы»
3. Чистов Д.В. «Информационные системы в экономике: Учебное пособие»