Федеральное государственное бюджетное профессиональное образовательное учреждение

Торжокский политехнический колледж

Федерального агентства по государственным резервам

Индивидуальное задание по МДК.01.01

«Разработка программных модулей»

Выполнил студент группы П-30

Павлов В.А.

Гордиенко В.В.

2025

Содержание

**Раздел 1.** Практическая значимость разрабатываемого ПС

**1.1 Основные цели и задачи разработки ПС**

Программное средство разрабатывается для автоматизации процессов управления салоном красоты, улучшения клиентского обслуживания, повышения эффективности работы сотрудников и увеличения прибыли. Основные функции программного обеспечения могут включать управление записью клиентов, ведение базы данных клиентов, управления запасами, создание отчетов и анализ эффективности работы.

**1.2 Пользователи ПС**

1. Администраторы салона — отвечают за организацию работы, записи клиентов, управление расписанием и персоналом.

2. Сотрудники (парикмахеры, косметологи и т.д.) — используют ПС для ведения расписания, получения информации о клиентах и услугам.

3. Клиенты — могут записываться на услуги через приложение или веб-интерфейс, проверять расписание и получать уведомления.

4. Руководство — использует данные для анализа и принятия стратегических решений, мониторинга эффективности бизнеса.

Если бы данного программного средства не существовало, салон красоты столкнулся бы с рядом негативных последствий: - Сложности в управлении записью — отсутствие автоматизации может привести к путанице в записях, пропущенным записям и недовольству клиентов. - Низкая эффективность работы — отсутствие инструментов для анализа работы может тормозить развитие бизнеса и усложнять оптимизацию процессов. - Увеличение человеческих ошибок — ручное ведение учета клиентов, услуг и финансовых операций увеличивает вероятность ошибок. - Сложности в управлении запасами — без программы сложно отслеживать наличие товаров и материалов, что может привести к их дефициту или избытку. Таким образом, разработка ПС для салона красоты значительно облегчает работу, улучшает качество обслуживания и способствует росту прибыли.

**Раздел 2.** Функциональные требования к ПС

**2.1 Основные функции системы**

1. Управление записями клиентов. Пользователи должны иметь возможность записывать новых клиентов на услуги, вносить изменения и отменять записи. Это может быть реализовано через простой интерфейс, где администраторы и мастера могут видеть текущее расписание и доступные временные слоты.

2. Ведение базы данных клиентов. Система должна предоставлять возможность добавления и редактирования информации о клиенте, включая контактные данные, историю посещений и предпочтения. Пользователи могут легко искать и открывать карточки клиентов для обновления или просмотра информации.

3. Управление услугами и ценами. Пользователи должны иметь возможность добавлять новые услуги, устанавливать и изменять их цены, а также задавать временные рамки для каждой услуги.

4. Отчеты и аналитика. Программное средство должно обеспечивать генерацию отчетов о доходах, клиентской базе и эффективности работы сотрудников.

**2.2 Как пользоваться функциями системы**

Для доступа к функционалу системы пользователям необходимо будет запустить программу. После этого они смогут легко ориентироваться в меню, находя нужные функции: добавлять записи, услуги и генерировать отчеты с помощью интуитивно понятного интерфейса.

**Раздел 3.** Модульная структура проектируемого ПС

Диаграмма UML

В данной программе 4 класса

Класс Service отвечает за представление услуги, т.е. представляет отдельную услугу, оказываемую в салоне красоты.

|  |
| --- |
| Service |
| public string Name { get; set; }  public decimal Price { get; set; }  public List<string> Masters { get; set; } |

Класс Client отвечает за представление клиента, т.е. представляет имя клиента и возвращает его.

|  |
| --- |
| Client |
| public string Name { get; set; } |
| public new string ToString() |

Класс Appointment отвечает за представление визита, т.е. представляет запись на конкретную услугу конкретным клиентом у конкретного мастера в определенное время.

|  |
| --- |
| Appointment |
| public Client Client { get; set; }  public Service Service { get; set; }  public string Master { get; set; }  public DateTime Date { get; set; } |

Класс Salon отвечает за управление салоном, т.е. отвечает за сам салон красоты и управляет всеми данными и операциями, связанными с ним. Это основной класс, который координирует работу с услугами, клиентами и записями.

|  |
| --- |
| Salon |
| public void LoadServicesFromFile(string fileName)  public void SaveServicesToFile(string fileName)  public void AddService(Service service)  public void AddClient(Client client)  public Client GetOrCreateClient(string name)  public Service GetServiceByName(string name)  public void AddAppointment(Appointment appointment)  public string GenerateClientRating()  public string GenerateRevenueReport(DateTime startDate, DateTime endDate)  public string GenerateEmployeeReport(DateTime startDate, DateTime endDate)  public void SaveReportToFile(string report, string fileName) |

**Раздел 4.** Описание алгоритмов

Алгоритм генерации рейтинга клиентов по частоте посещений (метод GenerateClientRating() класса Salon):

Описание по шагам:

1. Группировка посещений по клиентам: Сначала группируем все записи (appointments) по клиентам. Это позволяет узнать, сколько раз каждый клиент посетил салон.
2. Подсчет количества посещений для каждого клиента: Для каждой группы, созданной на предыдущем шаге, подсчитываем количество записей (то есть количество посещений).
3. Создание словаря “Клиент - Количество посещений”: Создаем словарь, где ключом является объект Client, а значением - количество его посещений.
4. Сортировка словаря по количеству посещений в порядке убывания: Сортируем словарь, чтобы клиенты с наибольшим количеством посещений были в начале списка.
5. Формирование текстового отчета: Создаем строку, которая будет содержать текстовый отчет.
6. Итерация по отсортированному словарю: Проходим по каждому элементу отсортированного словаря.
7. Добавление информации о клиенте и количестве посещений в отчет: Для каждого клиента добавляем в отчет строку вида “Имя клиента: Количество посещений”.
8. Возврат текстового отчета: Возвращаем строку, содержащую сформированный текстовый отчет.

Блок-схема

Алгоритм генерации отчета о выручке за период (метод GenerateRevenueReport(DateTime startDate, DateTime endDate) класса Salon):

Описание по шагам:

1. Фильтрация записей по дате: Выбираем из списка всех записей (appointments) только те, которые находятся в заданном периоде (между startDate и endDate).
2. Расчет суммарной выручки: Для каждой отобранной записи получаем стоимость услуги (Service.Price) и суммируем их.
3. Формирование текстового отчета: Создаем строку, которая будет содержать текстовый отчет с указанием периода и суммарной выручки.
4. Возврат текстового отчета: Возвращаем строку, содержащую сформированный текстовый отчет.

Блок-схема