**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ АСТРАХАНСКОЙ ОБЛАСТИ**

**Государственное бюджетное профессиональное образовательное учреждение Астраханской области «Астраханский колледж вычислительной техники»**

**Допустить к защите**

**«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.**

**Зам. директора по УМВР**

**\_\_\_\_\_\_\_\_\_\_\_С.В. Расторгуева**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**разработка автоматизированной системы   
«САЛОН КРАСОТЫ»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Пояснительная записка** | | | | | |
| **АКВТ.09.02.03.ДП00.12ПЗ.2020** | | | | | |
|  | ***Нормоконтролер*** | |  | ***Разработчик ВКР, студентка гр. ПБ-42*** |  | |
|  |  | Ю.С. Андрианова |  |  | И.О. Сарычев | |
| «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. | | | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. | | | |
|  | | |  | | | |
|  |  | |  | ***Руководитель ВКР*** | | |
|  |  |  |  |  | Р.Р. Заитова | |
|  | | | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020г. | | | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
|  |  |  |  | | | |

**2020**

Реферат

Объектом проектирования является процесс создания персонажа для настольной ролевой игры.

Разработанная система предназначена для автоматизации процессов:

* выбора расы героя;
* выбора класса героя;
* выбора параметров героя;
* выбора мировоззрения героя;
* выбора предыстории героя;
* выбора навыков;
* хранения информации о созданных персонажей;

Использование данного приложения позволит за короткое время создать собственного персонажа для игры в настольно ролевую игру без необходимости читать несколько книг с правилами, тем самым быстрее начав получать удовольствия от процесса игры.

Пояснительная записка представлена на 43 страницах, включает 5 таблиц, 41 рисунок.

Содержание

[Введение 5](#_Toc514352493)

[1 Исследовательский раздел 6](#_Toc514352494)

[1.1 Описание предметной области 6](#_Toc514352495)

[1.2 Описание технологии обработки информации 6](#_Toc514352496)

[1.3 Информационно-логическая модель системы 8](#_Toc514352497)

[1.4 Требования к техническому и программному обеспечению 10](#_Toc514352501)

[2 Технологический раздел 12](#_Toc514352504)

[2.1 Общие сведения о работе программы 12](#_Toc514352505)

[2.2 Функциональное назначение 12](#_Toc514352506)

[2.3 Инсталляция и выполнение разработанного программного продукта 12](#_Toc514352507)

[2.4 Даталогическая модель данных 13](#_Toc514352508)

[2.5 Общий алгоритм программного продукта 16](#_Toc514352509)

[2.6 Программные коды 16](#_Toc514352510)

[2.7 Руководство пользователя 19](#_Toc514352513)

[2.8 Программа и методика испытаний 40](#_Toc514352518)

[Заключение 42](#_Toc514352519)

[Список используемых источников 43](#_Toc514352520)

Введение

В век, когда большая половина людей на земле непосредственно связана с миром информационных технологий начиная с всемирной паутины где люди обмениваются новостями и просто общаются, заканчивая рабочим чатов для работников одного предприятия где важно вовремя передать важную информацию.

Выбранная тема исследования считается актуальной из-за стремительного роста игроков и полное отсутствия инструментов упрощения игры в русском сегменте.

Значимость темы исследования состоит в сокращении в трудовых и временных затрат на процесс создания персонажа тем самым понизив и без того высокий порог вхождения.

В условиях современного динамического развития общества и усложнения технической и социальной инфраструктуры информация становится таким же стратегическим ресурсом, как традиционные материальные и энергетические ресурсы. Современные информационные технологии, позволяющие создавать, хранить, перерабатывать и обеспечивать эффективные способы представления информационных ресурсов потребителю, стали важным фактором жизни общества и средством повышения эффективности управления всеми сферами общественной деятельности. Уровень использования информации становится одним из существенных факторов успешного экономического развития и конкурентоспособности региона, как на внутреннем, так и на внешнем рынке.

В настоящее время существуют уже готовые программные продукты, решающие довольно узкоспециализированные задачи. Но часто бывают случаи, когда данные программы не совсем подходят для конкретных условий предприятия. Возможно, они не полностью охватывают область задачи, либо, что бывает гораздо чаще, наоборот, затрагивают более широкую предметную область, что создаёт определённые трудности и неудобства в использовании. Кроме того, они, как правило, являются довольно дорогими и сложными. Часто такие программные продукты требуют специального обучения персонала, что так же стоит немалых затрат и требует времени на обучение. Именно поэтому и разрабатываются специальные программы, автоматизирующие деятельность конкретных предприятий. На основании решаемых задач они могут быть однопользовательскими и многопользовательскими экономическими информационными системами, автоматизированными рабочими местами и многими другими.

Выбранная тема исследования считается актуальной в связи со значительным упрощением и автоматизацией учёта и процесса реализации оказания услуг салоном красоты за счет использования информационных систем.

Значимость темы исследования состоит в сокращении трудовых и временных затрат на процессы, связанные с обслуживанием клиентов.

# **Исследовательский раздел**

## Описание предметной области

Деятельность салона красоты характеризуется следующими особенностями:

* основным направлением деятельности является оказание косметологических услуг;
* использованием широкого спектра форм расчетов, составлением договора публичный оферты;
* малочисленный штатный состав сотрудников, требующий качественной информационной поддержки.

Необходимо разработать программу, в которой сотрудники, регистрируясь, могли производить расчёт клиентов и регистрацию оказываемых услуг. При заключении сделки программа, на основе шаблона, должна составить договор.

Программа должна предусмотреть:

* общий интерфейс программы, целью, которой является создание программы в едином стиле (приятном для глаза пользователя, не вызывающего усталость и раздражение);
* осуществление быстрого и удобного выбора нужного пункта;
* хороший дизайн при исполнении программы.

Описание предметной области отражено в контекстной диаграмме (рисунок 1).

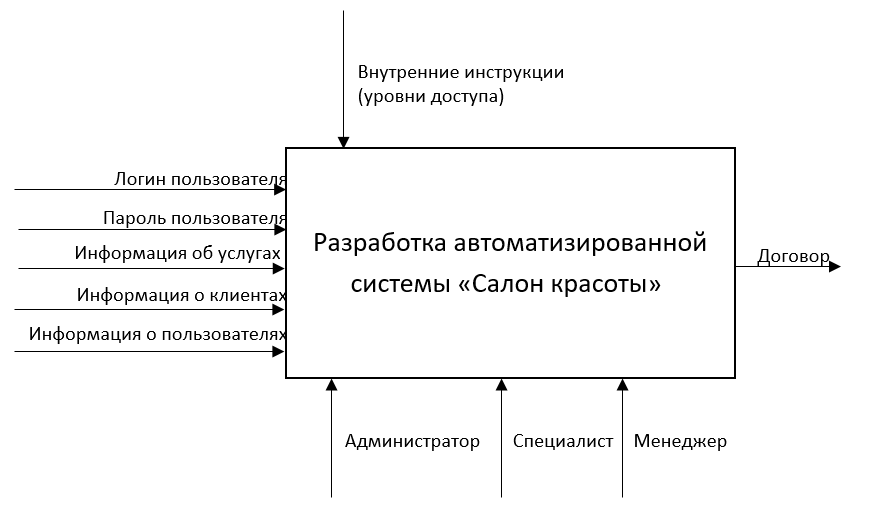


Рисунок 1 – Контекстная диаграмма

## Описание технологии обработки информации

На сегодняшний день инструментов для создания персонажей в настольно ролевой игре DnD 5e в российском сегменте отсутствуют, все используют английские источники, но в силу особенности языка происходит неправильный перевод. Из-за чего многие новички допускают колоссальное количество ошибок при заполнении листа персонажа, поэтому возник большой спрос на программный продукт для облегчения данной задачи.

Сложность добавляет не только в прочтении нескольких книг, но и в формулах, которые нужно запомнить и записать. Даже опытные мастера не могут запомнить все особенности формул.

На данный момент времени в салонах красоты нет специализированного программного обеспечения для учета оказываемых услуг.

При работе с данными клиента могут быть допущены ошибки, которые довольно проблематично устранить вручную. Также оформление договора публичной оферты занимает достаточно большое количество времени.

У руководства салона красоты не раз вставал вопрос о приобретении специализированного программного обеспечения для учета оказываемых услуг, однако он так и не был решен, вследствие отсутствия на рынке программных продуктов, полностью удовлетворяющих потребностям и финансовым возможностям организации.

Таким образом, проектируемая система решает основные проблемы: экономия времени менеджера и специалиста при работе с клиентом, записями. Поэтому была разработана и внедрена в организацию автоматизированная система с низкой себестоимостью и не уступающая в своих возможностях существующим программам аналогам.

Технология обработки информации представлена в виде диаграммы вариантов использования (рисунок 2).



Рисунок 2 – Диаграмма вариантов использования

## Информационно-логическая модель системы

### **Определение объектов**

В результате анализа процесса создания системы были выделены следующие сущности:

* Пользователь;

Сущность «Пользователи» предназначена для просмотра, создания персонажа и редактирования информации о персонаже.

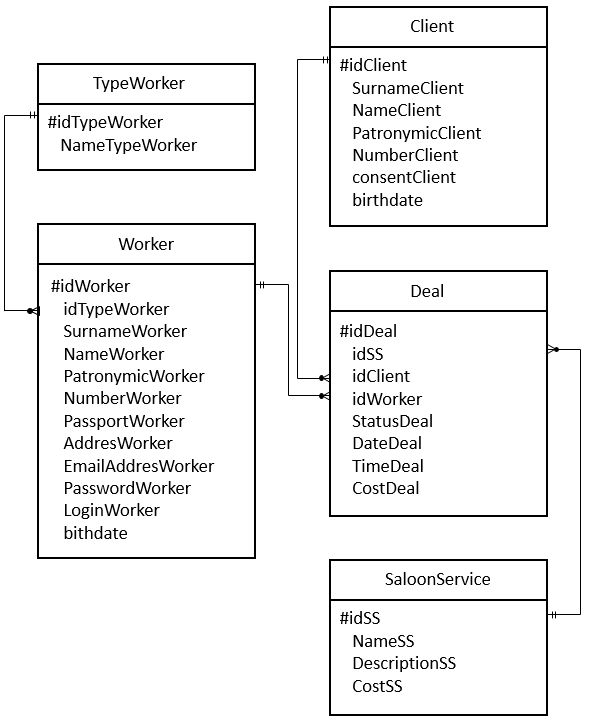


Рисунок 3 - ER-диаграмма

### **Требования к интерфейсу**

Для успешного применения разработанная система имеет удобный, интуитивно понятный интерфейс, выдержанный в спокойной цветовой гамме. Интерфейс проектируется для стандартного разрешения экрана 1366\*768, но также программа будет успешно отображаться и на прочих разрешениях.

Интерфейс не даёт пользователю ошибиться или предпринять неверное действие, т.к. он интуитивно понятен и не содержит ничего лишнего. Элементы управления должны обеспечивать все необходимые действия по работе с данными. Размещать на форме следует только логически связанные элементы управления. Все формы системы должны быть выполнены в едином стиле.

### **Описание параметров входной и выходной информации**

Входными данными для программы являются:

* данные о персонаже, вводимые пользователем с клавиатуры;

Выходной информацией для программы будет сформированные записи и договоры публичной оферты.

Выходная информация жестко закреплена, так как система не расширяемая и выходными данными являются чёткие понятия, применяемые в работе компании.

## Требования к техническому и программному обеспечению

### **Требования к техническому обеспечению**

Для работы программного продукта необходим компьютер со следующими требованиями:

* IBM-совместимый компьютер с МП Intel Pentium III или AMD Athlon с тактовой частотой 1000 МГц и выше;
* 512 Мб ОЗУ;
* SCSI или АТА100 жесткий диск объемом не менее 50 Мб;
* сетевая карта Ethernet 10/100 Мб;
* клавиатура;
* мышь.

### **Требования к программному обеспечению**

Для работы программного продукта необходимо следующее программное обеспечение:

Требования к серверу: ОС ­– Windows 7/8/8.1/10; MySQL Server 8.0; Microsoft SQL Server 2008; SQL Server Management Studio 2014.

Требования к клиенту: ОС ­– Windows 7/8/8.1/10; .NET Framework 4.7; DirectX.

# **Технологический раздел**

## Общие сведения о работе программы

Внедрение программного продукта позволит сократить время создания персонажа, не обращая внимания на многочисленные формулы и условности, которые происходят в момент создания персонажа.

В системе предусмотрены такие функции как:

* создание персонажа и добавления его в список персонажей;
* изменение данных о персонаже;
* удобный список уже созданных персонажей.

Внедрение автоматизированной системы «Салон красоты» позволит автоматизировать работу специалистов и администрации компании, предоставляя персоналу актуальную информацию об услугах, а также повысить скорость оформления договоров.

В системе предусмотрены такие функции как:

* регистрация клиента и сотрудника;
* редактирование и хранение информации о каждом клиенте;
* редактирование и хранение информации об оказываемых услугах;
* изменение персональных данных сотрудников;
* удобный поиск услуг.

Автоматизированная информационная система салона красоты была создана с помощью программного продукта Visual Studio 2019 и работает под управлением операционной системы MS Windows 7 и выше.

В качестве СУБД (система управления базами данных) используется Microsoft SQL Server.

Использование базы данных позволяет оперативно обрабатывать данные о недвижимостях, клиентах, сделок по запросу пользователя, а также способствует быстрому формированию сделок.

Сравнительно низкие системные требования позволяют использовать информационную систему на компьютерах практически с любой аппаратной конфигурацией.

При запуске программы откроется форма «Авторизация».

При нажатии на кнопку «Войти», открывается форма Главное меню.

## Функциональное назначение

Функциональным назначением программы является предоставление пользователям системы совершать запись клиентов на оказываемые услуги.

Данная система может эксплуатироваться только салонами красоты, имеющее соответствующее техническое и программное обеспечение.

## Инсталляция и выполнение разработанного программного продукта

Схемы взаимодействия программ (модулей) отображают путь активации программ и взаимодействии с соответствующими данными.

Каждая программа в схеме взаимодействия программ показывается только один раз (в схеме работы системы программа может изображаться более чем в одном потоке управления).

Схема взаимодействия программ состоит из:

* символов данных, указывающих на наличие данных;
* символов процесса, указывающих на операции, которые следует выполнить над данными;
* линейных символов, отображающих поток между процессами и данными, а также инициации процессов;
* специальных символов, используемых для облегчения написания и чтения схем.

Установка системы начинается с установки сервера. Так как система по стандарту работает с пакетом MySQL Server, в состав дистрибутива входит его установочный пакет.

MySQL – [свободная](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%9F%D0%9E) [реляционная система управления базами данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94).

База данных должна быть развернута на компьютере, который будет играть роль сервера. Для того чтобы компьютер стал играть роль сервера требуется:

* подключение компьютера к локальной сети предприятия;
* создать правила в брандмауэре разрешающие входящие подключения по порту 3306 и протоколам TCP и UDP;
* запустить службу Обозреватель MySQL Notifier.

Основным условием является обязательное нахождение всех ПК и ПК исполняющего роль сервера в локальной сети.

После распаковки система, не требует ни какой специальной настройки.

После распаковки в папке появится только один исполняемый файл, который и является распакованной системой.

При запуске программы появляется форма входа с расположенными на ней графическими элементами управления.

## Даталогическая модель данных

Для решения поставленной задачи на основании объектов автоматизации создано 5 таблиц, входящие в состав базы данных BeautyBD. База данных BeautyBD приведена к 3NF.

Целостность данных обеспечивается заложенными во время создания таблиц ограничениями на тип, размер и диапазон допустимых значений.

Данные о пользователях содержатся в таблице Worker. Структура и правила поддержки целостности данных в таблице Worker приводятся в таблице 1.

Мои таблицы (распишу чуть позже, мне было важно скинуть вам что-то)

Таблица 1 - классы

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| ID | int | Первичный |  |
| Class\_name | nvarchar(50) |  |  |
| Class\_description | varchar(MAX) |  |  |
| SkillCount | int |  |  |

Таблица 2 - расы

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| ID | int | Первичный |  |
| Name | varchar(50) |  |  |
| Description | varchar(MAX) |  |  |
| Speed | int |  |  |

Таблица 3 - броня

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| ID | int | Первичный |  |
| Name | varchar(100) |  |  |
| ArmorClass | int |  |  |
| Description | varchar(50) |  |  |
| Weight | float |  |  |

Таблица 4 - Chosable

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| ID | int |  |  |
| ClassID | int |  |  |
| SpecID | int |  |  |
|  |  |  |  |
|  |  |  |  |

Таблица 5 - ClassHasMagic

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| ID | int |  |  |
| ClassID | int |  |  |
| SpellCount | int |  |  |
| Cantrip | int |  |  |
| Lvl\_1 | int |  |  |
| HaveFormul | bit |  |  |

Таблица 6 - Items

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| ID | int |  |  |
| Name | varchar(MAX) |  |  |
| Description | varchar(MAX) |  |  |
| Weight | float |  |  |
|  |  |  |  |

Таблица 7 - Origins

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| ID | int |  |  |
| Name | varchar(50) |  |  |
| Descriptions | varchar(MAX) |  |  |
|  |  |  |  |
|  |  |  |  |

Таблица 8 - OriginsItems

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| ID | int |  |  |
| OriginID | int |  |  |
| ItemID | int |  |  |
| Count | int |  |  |
|  |  |  |  |

Таблица 9 - OriginSkill

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| ID | int |  |  |
| OriginID | int |  |  |
| OriginSkill | int |  |  |
|  |  |  |  |
|  |  |  |  |

Таблица 10 - RaceParam

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| ID | int |  |  |
| RaceID | int |  |  |
| Param | varchar(50) |  |  |
| Bonus | int |  |  |
| IsChoosable | bit |  |  |

Таблица 11 - RaceSkills

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| ID | int |  |  |
| RaceID | int |  |  |
| SkillID | int |  |  |
| IsChoosable | bit |  |  |
|  |  |  |  |

Таблица 12 - RaceSpells

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| ID | int |  |  |
| RaceID | int |  |  |
| SpellID | int |  |  |
|  |  |  |  |
|  |  |  |  |

Таблица 13 - RaceSpells

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| ID | int |  |  |
| ClassID | int |  |  |
| SkillID | int |  |  |
|  |  |  |  |
|  |  |  |  |

Таблица 14 - Skills

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| ID | int |  |  |
| Name | varchar(50) |  |  |
| Description | varchar(MAX) |  |  |
|  |  |  |  |
|  |  |  |  |

Таблица 15 - Spec

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| ID | int |  |  |
| Name | varchar(100) |  |  |
| Description | varchar(MAX) |  |  |
|  |  |  |  |
|  |  |  |  |

Таблица 16 - SpellClass

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| ID | int |  |  |
| ClassID | int |  |  |
| SpellID | int |  |  |
|  |  |  |  |
|  |  |  |  |

Таблица 17 - Spells

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| ID | int |  |  |
| Name | varchar(MAX) |  |  |
| Lvl | int |  |  |
| SchoolName | varchar(50) |  |  |
| Description | varchar(MAX) |  |  |

Таблица 18 - Weapons

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| ID | int |  |  |
| Name | varchar(MAX) |  |  |
| Damage | varchar(10) |  |  |
| Description | varchar(MAX) |  |  |
| Weight | float |  |  |
| TwoHanded | bit |  |  |
| Long | bit |  |  |
| Remote | bit |  |  |
| Lung | bit |  |  |
| Throwing | bit |  |  |
| Special | bit |  |  |
| Rechargeable | bit |  |  |
| Heavy | bit |  |  |
| Universal | bit |  |  |
| Fencing | bit |  |  |

Таблица 1 - Структура таблицы Worker

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| idWorker | int | Первичный | Идентификатор пользователя |
| idTypeWorker | int | Вторичный | Идентификатор типа пользователя |
| SurnameWorker | varchar(50) |  | Фамилия пользователя |
| NameWorker | varchar(50) |  | Имя пользователя |
| PatronymicWorker | varchar(50) |  | Отчество пользователя |
| NumberWorker | varchar(20) |  | Номер телефона пользователя |
| PassportWorker | varchar(50) |  | Паспортные данные пользователя |
| AddresWorker | varchar(50) |  | Адрес пользователя |
| EmailAddresWorker | varchar(50) |  | Почта пользователя |
| PasswordWorker | varchar(50) |  | Пароль пользователя |
| LoginWorker | varchar(50) |  | Логин пользователя |
| birthdate | date |  | Дата рождения пользователя |

Данные о типах пользователей в таблице TypeWorker. Структура и правила поддержки целостности данных в таблице employee приводятся в таблице 2.

Таблица 2 - Структура таблицы TypeWorker

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| idTypeWorker | int | Первичный | Идентификатор типа пользователя |
| NameTypeWorker | varchar(50) |  | Название типа пользователя |

Данные о клиентах содержатся в таблице Сlient. Структура и правила поддержки целостности данных в таблице Сlient приводятся в таблице 3.

Таблица 3 - Структура таблицы Сlient

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| idClient | int | Первичный | Идентификатор клиента |
| NameClient | varchar(50) |  | Имя клиента |
| SurnameClient | varchar(50) |  | Фамилия клиента |
| PatronymicClient | varchar(50) |  | Отчество клиента |
| NumberClient | varchar(20) |  | Телефон клиента |
| dateBirth | date |  | Дата рождения клиента |
| consentClient | varchar(30) |  | Согласие на обработку персональных данных |

Данные об услугах в таблице SaloonService. Структура и правила поддержки целостности данных в таблице SaloonService приводятся в таблице 4.

Таблица 4 - Структура таблицы SaloonService

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| idSS | int | Первичный | Идентификатор услуги |
| NameSS | varchar(50) |  | Наименование услуги |
| DescriptionSS | varchar(50) |  | Описание услуги |
| CostSS | decimal(18, 0) |  | Стоимость услуги |

Данные о сделках (записях) в таблице Deal. Структура и правила поддержки целостности данных в таблице Deal приводятся в таблице 5.

Таблица 5 - Структура таблицы Deal

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Ключ** | **Назначение** |
| idDeal | int | Первичный | Идентификатор сделки |
| idSS | int | Вторичный | Идентификатор услуги |
| idWorker | int | Вторичный | Идентификатор специалиста |
| idClient | int | Вторичный | Идентификатор клиента |
| StatusDeal | varchar(50) |  | Статус сделки |
| DateDeal | date |  | Дата сделки |
| TimeDeal | varchar(50) |  | Время сделки |
| CostDeal | decimal(18, 0) |  | Стоимость сделки |

## Общий алгоритм программного продукта

Алгоритм данной системы основан на использовании технологии «окно в окне».

Имеется главное приложение, которое выступает в роли оболочки для других форм.

Это позволило создать очень удобный интерфейс и не перегружать форму лишней информацией.

Данная система имеет следующие характеристики:

* обладает интерактивностью, то есть обратной связью с пользователем;
* обеспечивает удобный интерфейс между программой и пользователем;
* удобное нахождение и открытие необходимой информации.

Автоматизированная система салона красоты включает три основные подзадачи:

* первая подзадача позволяет оформлять сделки;
* вторая подзадача обеспечивает долгосрочное хранение информации о клиентах, услугах, пользователях, сделок;
* третья подзадача позволяет формирование договоров публичной оферты.

## Программные коды

### **Среда разработки**

Автоматизированная система салона красоты разрабатывалась в среде программирования Visual Studio 2019 Сommunity.

Visual Studio – линейка продуктов компании Microsoft, включающих [интегрированную среду разработки](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные [приложения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), так и приложения с [графическим интерфейсом](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F), в том числе с поддержкой технологии Windows Forms, WPF, а также [веб-сайты](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82), [веб-приложения](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5), [веб-службы](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%BB%D1%83%D0%B6%D0%B1%D0%B0) как в [родном](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4), так и в [управляемом](https://ru.wikipedia.org/wiki/%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%8F%D0%B5%D0%BC%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, [.NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework), Xbox, Windows Phone .NET Compact Framework и Silverlight.

Visual Studio включает в себя [редактор исходного кода](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80_%D0%B8%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BE%D0%B4%D0%B0) с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный [отладчик](https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio_Debugger) может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер [классов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) и дизайнер [схемы базы данных](https://ru.wikipedia.org/wiki/%D0%A1%D1%85%D0%B5%D0%BC%D0%B0_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). Visual Studio позволяет создавать и подключать сторонние дополнения ([плагины](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%B3%D0%B8%D0%BD)) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем [контроля версий исходного кода](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%8F%D0%BC%D0%B8) (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на [предметно-ориентированных языках программирования](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%BC%D0%B5%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA)) или инструментов для прочих аспектов [процесса разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F) (например, клиент Team Explorer для работы с Team Foundation Server).

### **Используемые технологии**

При разработке автоматизированной системы салона красоты использовалась технология WPF.

Технология WPF (Windows Presentation Foundation) является часть экосистемы платформы .NET и представляет собой подсистему для построения графических интерфейсов.

Если при создании традиционных приложений на основе WinForms за отрисовку элементов управления и графики отвечали такие части ОС Windows, как User32 и GDI+, то приложения WPF основаны на DirectX. В этом состоит ключевая особенность рендеринга графики в WPF: используя WPF, значительная часть работы по отрисовке графики, как простейших кнопочек, так и сложных 3D-моделей, ложиться на графический процессор на видеокарте, что также позволяет воспользоваться аппаратным ускорением графики.

Одной из важных особенностей является использование языка декларативной разметки интерфейса XAML, основанного на XML: вы можете создавать насыщенный графический интерфейс, используя или декларативное объявление интерфейса, или код на управляемых языках C# и VB.NET, либо совмещать и то, и другое.

Преимущества WPF

Использование традиционных языков .NET-платформы - C# и VB.NET для создания логики приложения

Возможность декларативного определения графического интерфейса с помощью специального языка разметки XAML, основанном на xml и представляющем альтернативу программному созданию графики и элементов управления, а также возможность комбинировать XAML и C#/VB.NET.

Независимость от разрешения экрана: поскольку в WPF все элементы измеряются в независимых от устройства единицах, приложения на WPF легко масштабируются под разные экраны с разным разрешением.

Новые возможности, которых сложно было достичь в WinForms, например, создание трехмерных моделей, привязка данных, использование таких элементов, как стили, шаблоны, темы и др.

Хорошее взаимодействие с WinForms, благодаря чему, например, в приложениях WPF можно использовать традиционные элементы управления из WinForms.

Богатые возможности по созданию различных приложений: это и мультимедиа, и двухмерная и трехмерная графика, и богатый набор встроенных элементов управления, а также возможность самим создавать новые элементы, создание анимации, привязка данных, стили, шаблоны, темы и многое другое.

Аппаратное ускорение графики - вне зависимости от того, работаете ли вы с 2D или 3D, графикой или текстом, все компоненты приложения транслируются в объекты, понятные Direct3D, и затем визуализируются с помощью процессора на видеокарте, что повышает производительность, делает графику более плавной.

Создание приложений под множество ОС семейства Windows - от Windows XP до Windows 10.

В тоже время WPF имеет определенные ограничения. Несмотря на поддержку трехмерной визуализации, для создания приложений с большим количеством трехмерных изображений, прежде всего игр, лучше использовать другие средства - DirectX или специальные Фреймворки, такие как Monogame или Unity.

Также стоит учитывать, что по сравнению с приложениями на Windows Forms объем программ на WPF и потребление ими памяти в процессе работы в среднем несколько выше. Но это с лихвой компенсируется более широкими графическими возможностями и превышенной производительностью при отрисовке графики.

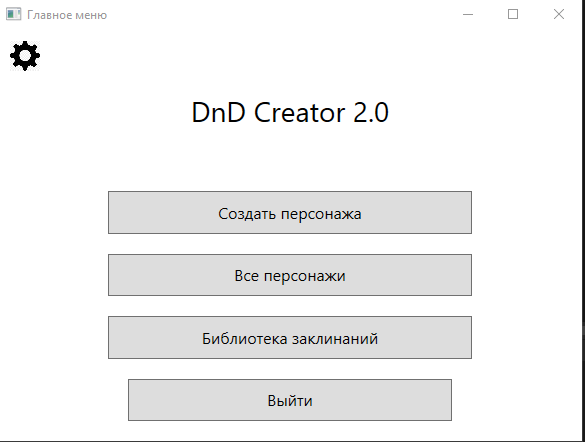
История развития

WPF является частью экосистемы .NET и развивается вместе с Фреймворком .NET и имеет те же версии. Первая версия WPF 3.0 вышла вместе с .NET 3.0 и операционной системой Windows Vista в 2006 году. С тех пор платформа последовательно развивается. Последняя версия WPF 4.6 вышла параллельно с .NET 4.6 в июле 2015 года, ознаменовав девятилетие данной платформы.

## Руководство пользователя

### **Главное меню**

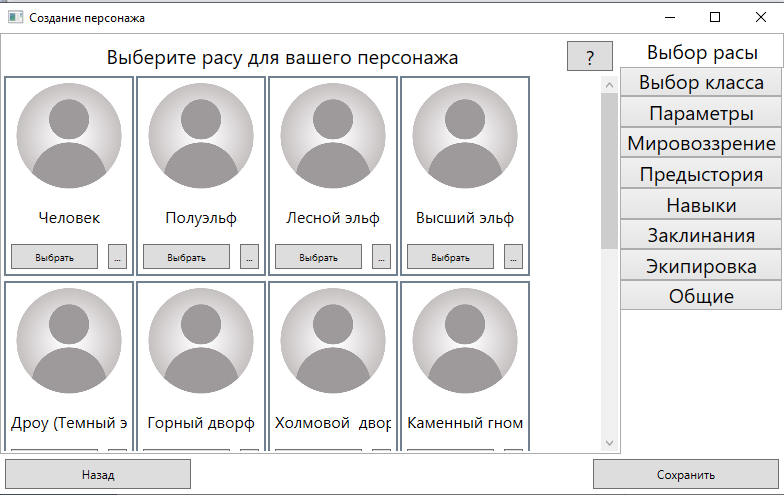
В главном меню у нас появляются такие возможности: создание персонажа, просмотр персонажей и просмотра списка заклинаний.



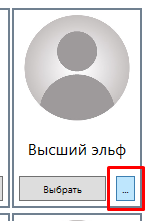
### **Создание персонажа**

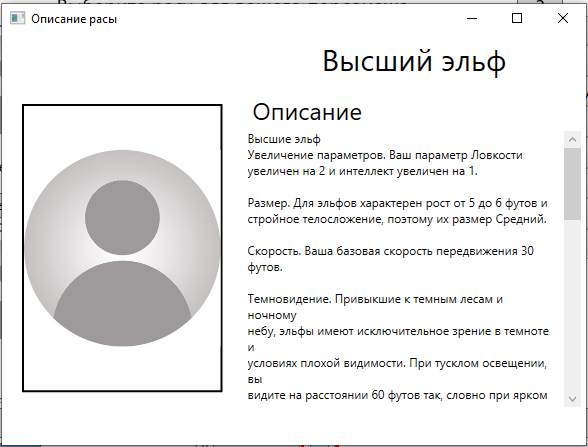
Пользователю открывается окно персонажей, которое содержит 8 вкладок.

В вкладе под названием «Выбор расы» мы выбираем для себя какую расу приобретет наш персонаж. От выбора раса зависит много параметров: отношение к вам; умения в навыках свойственные выбранной расе; ваши внешние параметры и физические параметры.

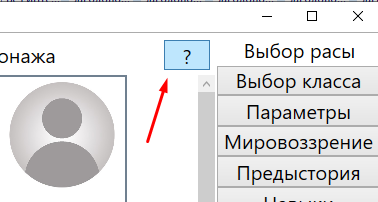


Чтобы узнать подробнее о расе можно нажать на кнопку «…» для открытия окна с краткой информацией.

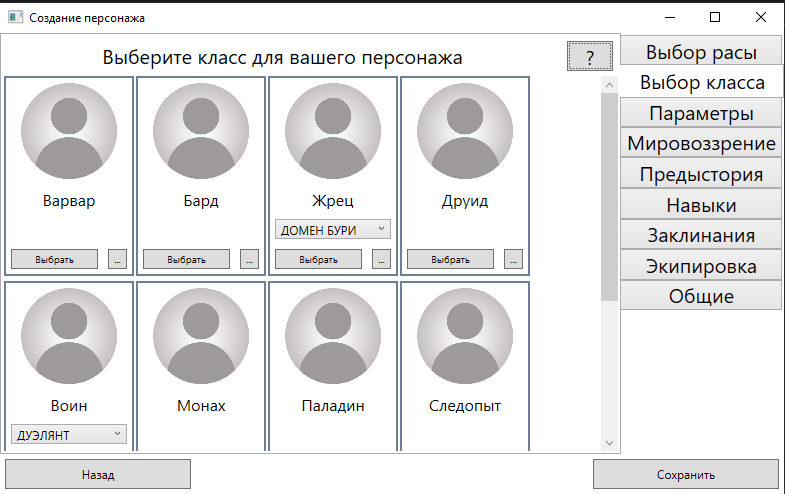




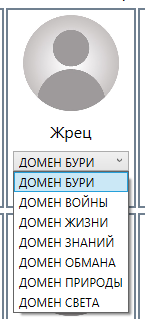
Так же на каждой из вкладок присутствует кнопка со справкой об этом вкладке.



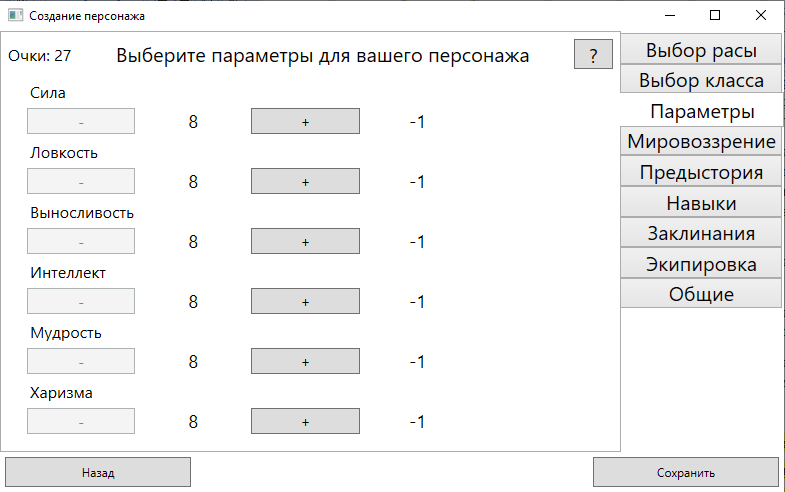
Во вкладке под названием «Выбор класса» мы выбираем какую специализацию выбирает наш персонаж. От этого зависит количество навыков которые вы можете владеть.



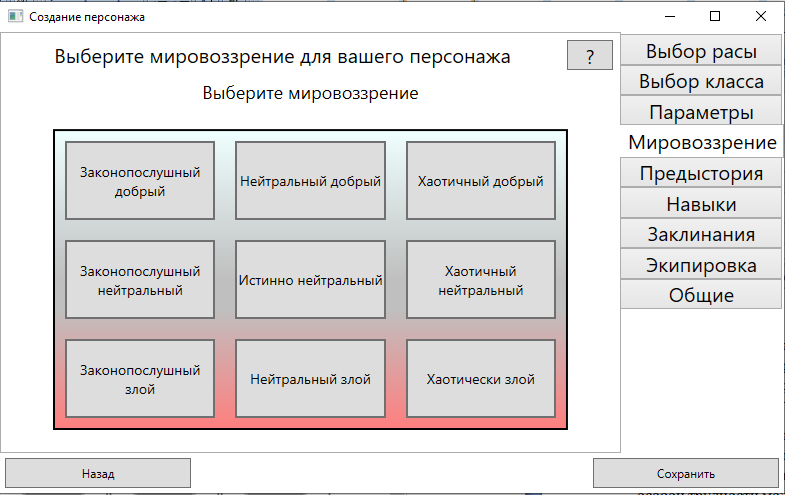
У некоторых классов есть подтипы. Когда вы выбрали подтип вашего класса, вы можете узнать особенность вашего подтипа в описании класса.



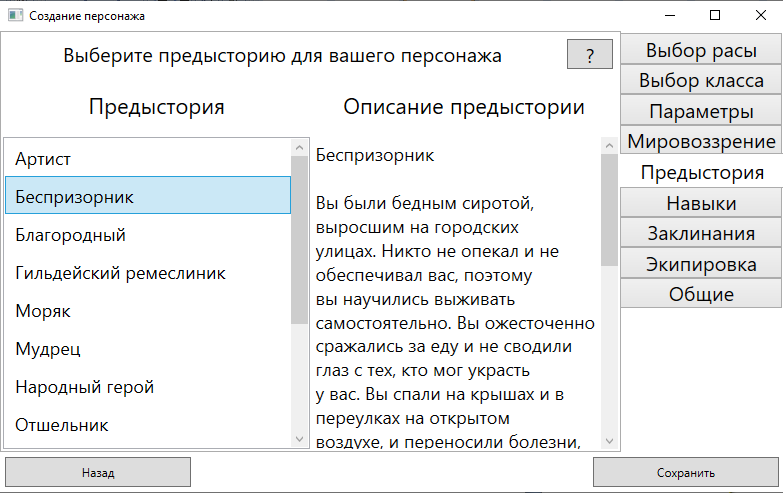
Во вкладе под названием «Параметры» мы распределяем очки способностей. Которые будут нас не раз выручать



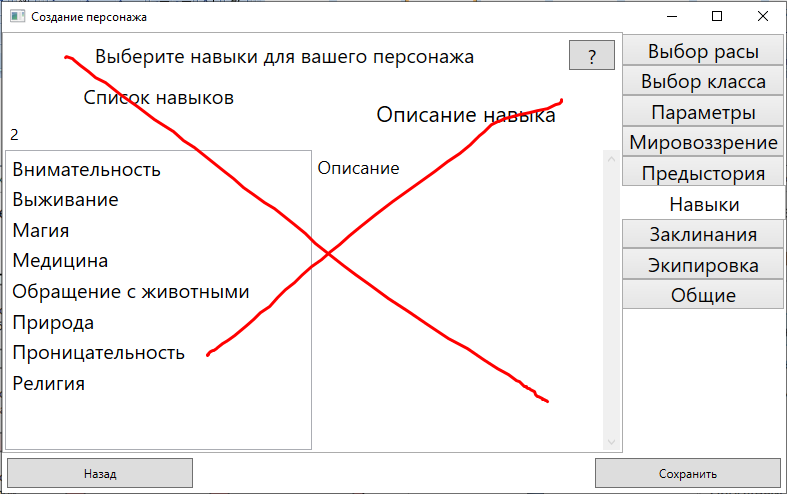
Во вкладе под названием «Мировоззрение» мы выбираем какое мировоззрение будет у нашего героя. Его мировоззрение влияет на поступки которые он может совершать. Законопослушно добрый никогда не станет воровать или обманывать людей, а хаотично-злой вряд ли будет помогать без собственной выгоды.



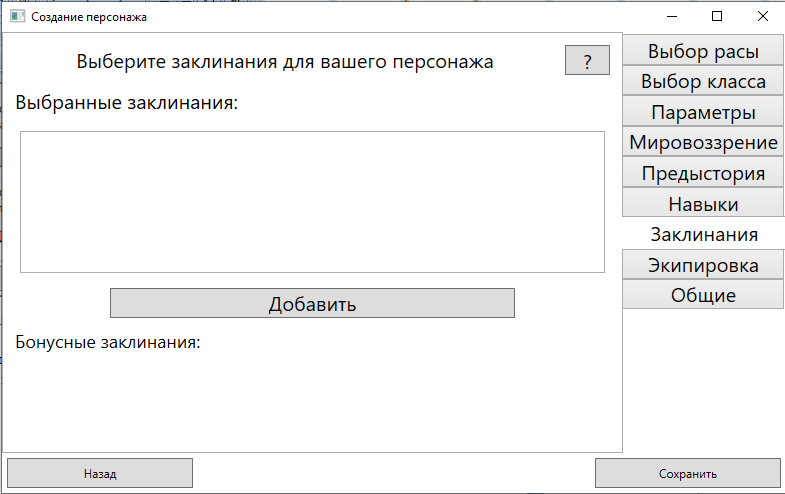
Во вкладе под названием «Предыстория» мы выбираем какое прошлое было у нашего персонажа, что повлияло на становления на выбранный им путь. От предыстории так же зависит какие навыки мы получим. Например выбрав предысторию «Артист» наш герой без особой трудности может показать трюк, умел в акробатике и умеет играть на одном из инструментов.



Во вкладе под названием «Навыки» мы выбираем ряд дополнительных навыков в которых мы умелы.



Во вкладе под названием «Заклинания» нужно будет добавить те заклинания, который ваш персонаж будет использовать на протяжении игры

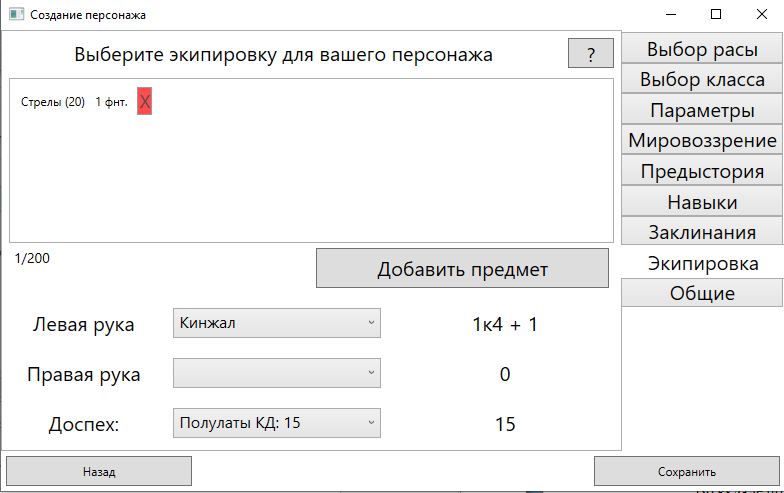


Не у каждого класса есть магия, если кнопка «добавить» вам недоступна после выбора класса, то смело можете переходить к следующей вкладке.

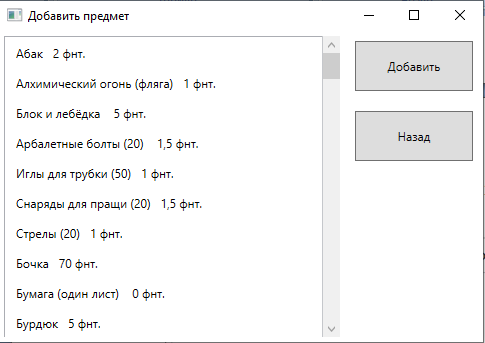
А если кнопка доступна, значит вы можете открыть окно выбора заклинаний, в этом окне вам будет показано сколько заклинаний вы можете выбрать для вашего персонажа



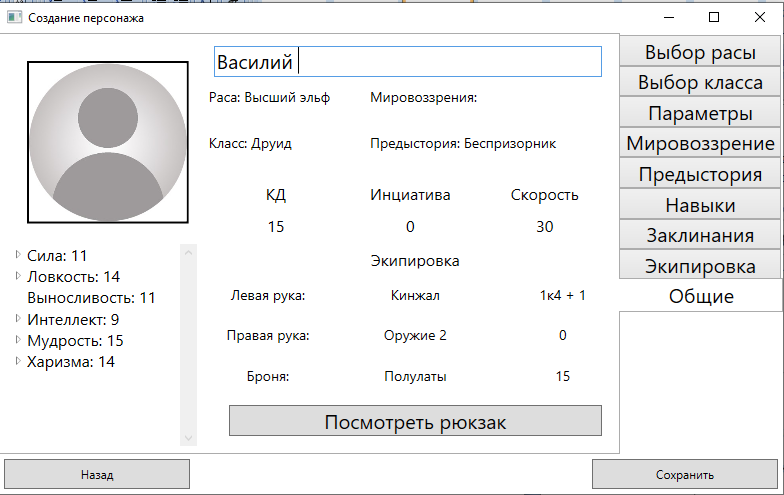
Во вкладе под названием «Экипировка» нам предстоит выбрать какое снаряжение у нас будет. Оружие, броня, рюкзак с вещами, все это имеет вес.



Вы можете добавить предмет себе в инвентарь если захотите для этого нажмите кнопку «Добавить предмет»



Во вкладе под названием «Общие» у нас выводиться вся основная информация о персонаже, которого мы сделали



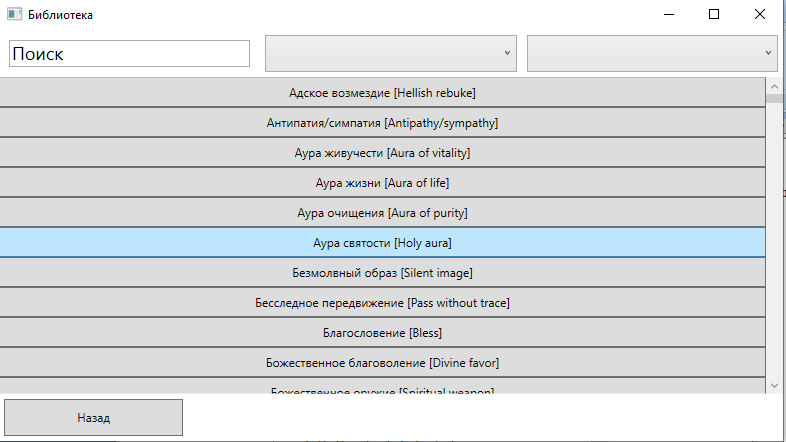
Как только вы дошли до этой вкладки и убедились в том что все выбрали, можете нажать кнопку «Сохранить» чтобы информация о вашем персонаже сохранилась

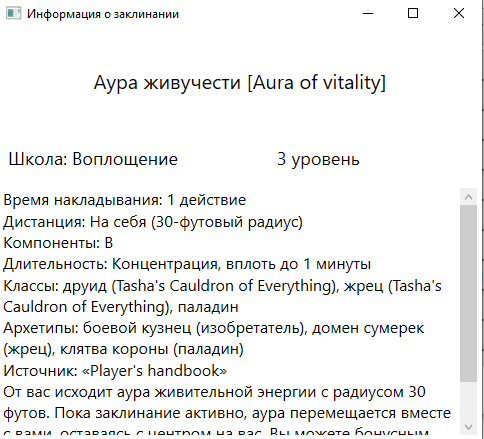
### **Все персонажи**

…

### **Библиотека заклинаний**

В этом окне вы сможете найти заклинания о которых хотите узнать





Если авторизованный пользователь является администратором, то помимо возможностей менеджера, у администратора появляются такие возможности, как: редактирование, добавление и удаления услуг и сотрудников.

При переходе из Главного меняю во вкладку «Сотрудники» открывается окно с данными всех сотрудников, включая пароли, доступных к изменению, добавлению и удалению.

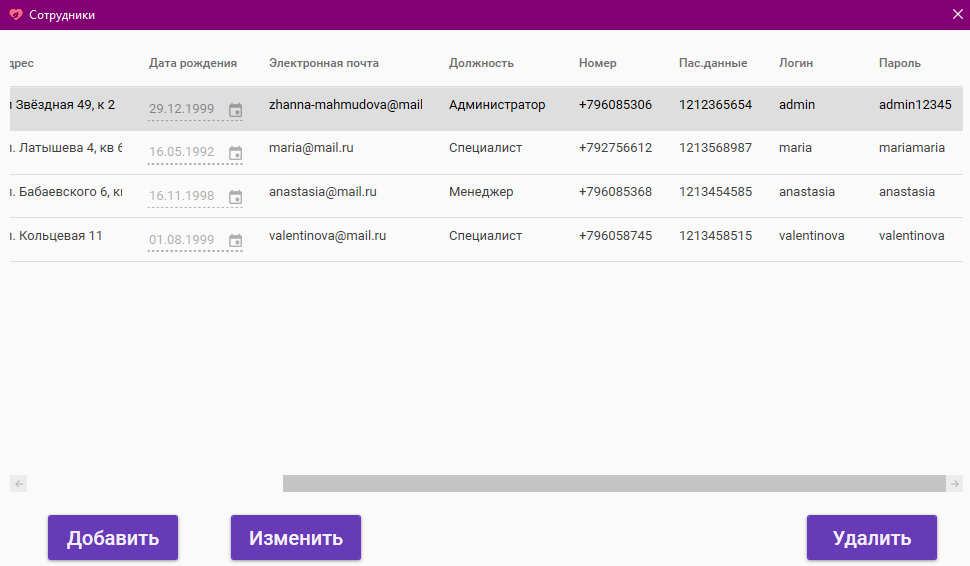


Рисунок 33 – Окно «Сотрудники» для администратора

При переходе из Главного меняю во вкладку «Сотрудники» открывается окно с данными всех сотрудников, доступных к чтению, добавлению, удалению и редактированию.

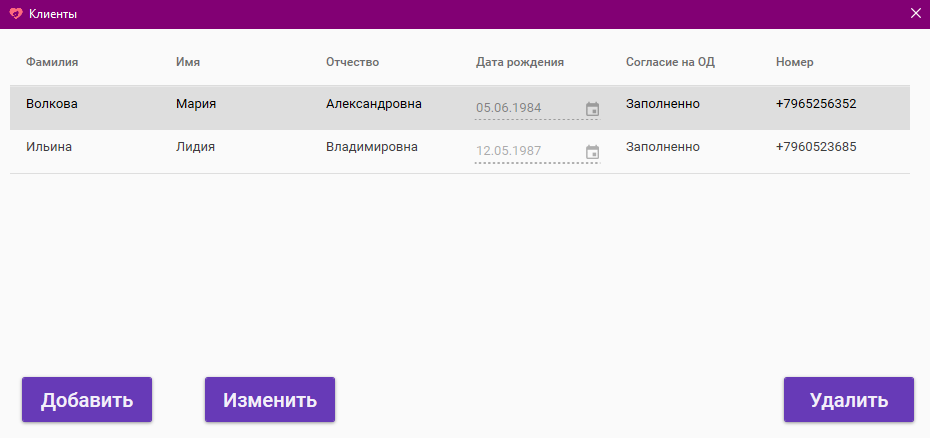


Рисунок 34 – Окно «Сотрудники» для администратора

При нажатии на кнопку «Добавить» предоставляется возможность зарегистрировать сотрудника. Нажимая кнопку «Принять» данные сохраняются в БД и обновляются на предыдущем окне.

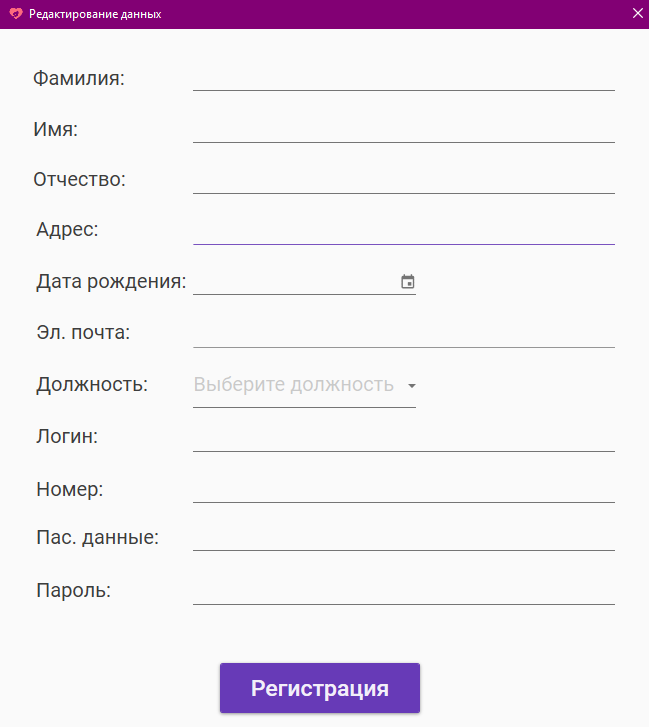


Рисунок 35 – Окно «Редактирование данных» сотрудника для администратора

Если в окне «Сотрудники» выбрать существующего сотрудника и нажать на кнопку «Изменить», то данные откроются в новом окне «Редактирование данных» и будут доступны для редактирования.

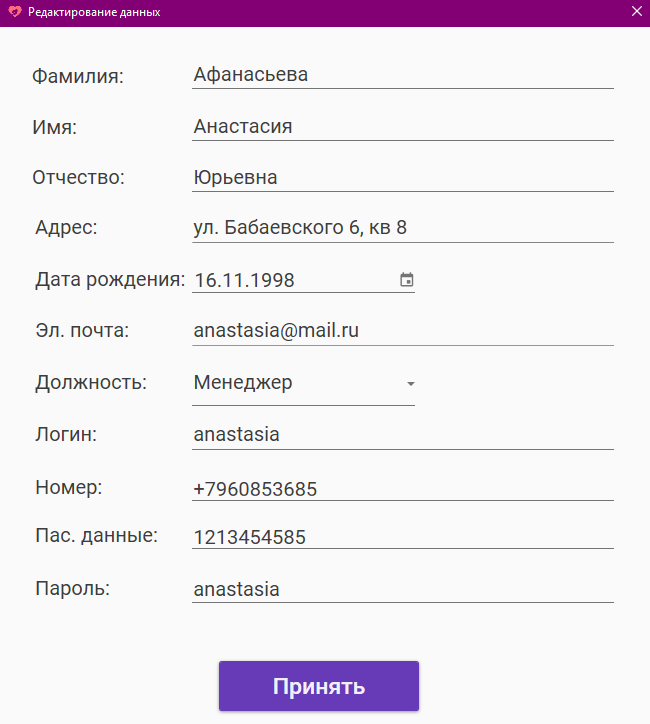


Рисунок 36 – Окно «Редактирование данных» сотрудника для администратора

Если в окне «Сотрудники» выбрать существующую запись и нажать на кнопку «Удалить» появится предупреждающее уведомление.

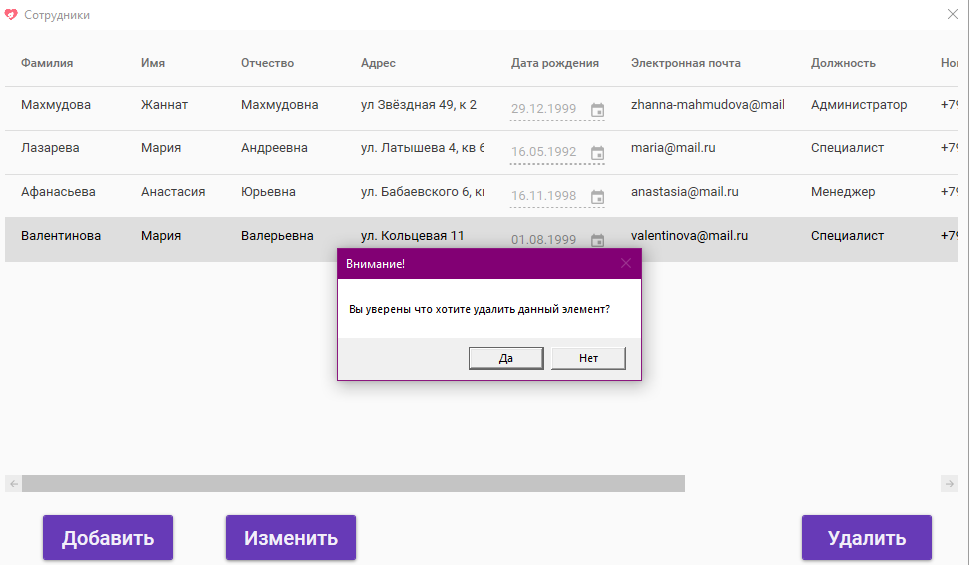


Рисунок 37 – Окно «Сотрудники» для администратора. Уведомление об удалении

При переходе из Главного меняю во вкладку «Услуги» открывается окно с данными о услугах, доступных к чтению, добавлению, удалению и редактированию.

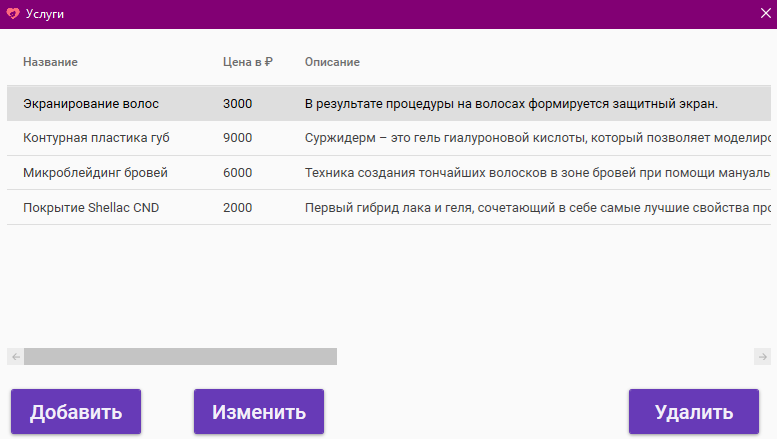


Рисунок 38 – Окно «Услуги» для администратора

При нажатии на кнопку «Добавить» предоставляется возможность добавить услугу. Нажимая кнопку «Принять» данные сохраняются в БД и обновляются на предыдущем окне.

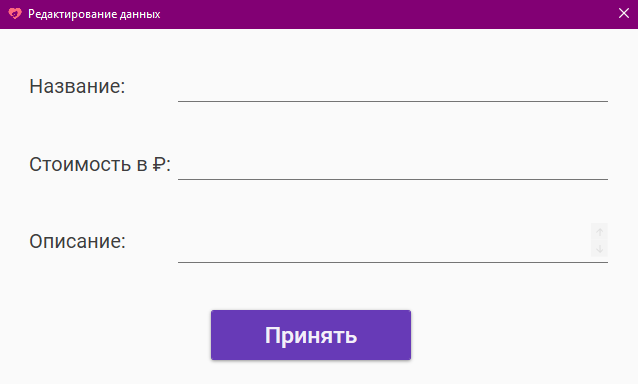


Рисунок 39 – Окно «Редактирование данных» услуги для администратора

Если в окне «Услуги» выбрать существующую услугу и нажать на кнопку «Изменить», то данные откроются в новом окне «Редактирование данных» и будут доступны для редактирования.

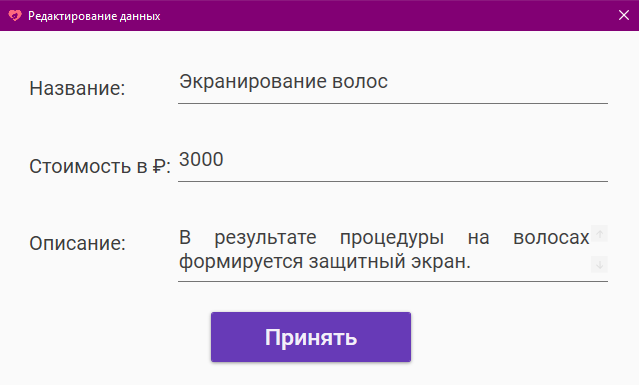


Рисунок 40 – Окно «Редактирование данных» услуги для администратора

Если в окне «Услуги» выбрать существующую услугу и нажать на кнопку «Удалить» появится предупреждающее уведомление.

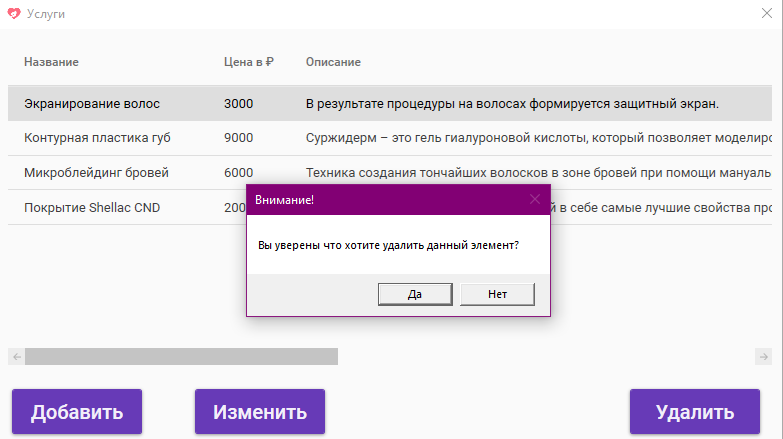


Рисунок 41– Окно «Услуги» для администратора. Уведомление об удалении

## Программа и методика испытаний

Проверка программного продукта должна подтвердить правильность выполнения следующих операций:

* проверка формирования экранных форм на основе, имеющейся в системе информации проводить путем запуска и программы и выбора любого пункта меню;
* после щелчка на подразделе меню появляется форма, содержащая некоторые компоненты (текст, кнопки, таблицы и т.д.);
* проверка корректного формирования сделки.

Проверка программного продукта должна подтвердить правильность выполнения следующих операций:

* наполнение введенными данными;
* формирование экранных форм на основе, имеющейся в системе информации.

Корректность наполнения базы данных введенными данными проверяется путем ввода информации о клиентах, с последующим повторным вызовом соответствующих форм редактирования данных.

Проверка формирования экранных форм необходима для контроля корректности предоставления информации сотруднику компании.

Проверка клиентского модуля сотрудником:

* запустить исполняемый модуль SaloonBeaty.exe;
* авторизоваться;
* на главной форме нажать кнопку «Запись»;
* нажать на кнопку «Добавить»;
* заполнить необходимые данные в полях для ввода данных записи на услугу;
* нажать на кнопку «Принять»;
* проверить добавление новой записи на услугу.

# **Заключение**

В результате выполнения выпускной квалификационной работы была разработана автоматизированная система «Салон красоты».

Разработанная система полностью удовлетворяет требованиям задания и охватывает полный спектр задач.

Разработанная система обеспечивает автоматизацию следующих процессов:

* регистрация работников салона красоты;
* регистрация клиентов;
* добавление услуг;
* редактирование данных работников салона красоты;
* редактирование данных клиентов;
* редактирование услуг;
* управление записями оказания услуг;
* управление клиентами;
* формирование договора оказания услуг;
* поиск и фильтрация записей оказания услуг.

Используя данную программу, пользователю предоставляется возможность быстрого и удобного управления клиентами, услугами, записями.

Разработанная система успешно решает все поставленные задачи. Программный продукт отличается дружественным интерфейсом, легкой структурой.

Внедрение автоматизированной системы позволит разгрузить и облегчить работу сотрудников, справиться с большим объемом информации и совершать меньше ошибок, а администрации эффективно управлять компанией.

# **Список используемых источников**

1. Кормен Т. Алгоритмы. Вводный курс – Вильямс – 2016, 208 c.
2. Майлз Р.C# Programming Yellow Book: Учебное пособие – Отдел компьютерных наук Университета Халла – 2016, 344 c.
3. Тюкачев Н., Хлебостроев В. C#. Основы программирования – Лань – 2017, 415 c.
4. Фиайли Крис SQL – ДМК Пресс – 2013, 456 c.
5. Фримен Э. Паттерны проектирования – Питер – 2017, 656 с.