Министерство образования Московской области

Государственное образовательное учреждение высшего образования Московской области

«Государственный гуманитарно-технологический университет»

**Ликино-Дулевский политехнический колледж – филиал ГГТУ**

**КУРСОВОЙ ПРОЕКТ**

«Разработка программного модуля для реализации операций с базой данных игрового приложения»

МДК 01.01 «Разработка программных модулей»

**Выполнила:**

Синюкова Ирина Андреевна\_\_\_\_\_\_\_\_\_ \_\_\_

студент группы ИСП.20А\_\_\_\_\_\_\_\_\_\_\_\_\_

09.02.07 Информационные системы и программирование)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

очной формы обучения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Руководитель:**

Кузьмина Елена Евгеньевна­­­­­­­­­­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка

\_\_\_\_\_\_\_\_\_/\_\_\_­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Подпись руководителя

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ликино-Дулево

2023 год

Оглавление

Введение1

1. Назначение разработки4
2. Требования к программному модулю4

2.1 Требования к функциональным характеристикам5

2.2 Требования к составу и параметрам технических средств6

2.3 Требования к информационной и программной совместимости6

1. Руководство программиста4

3.1 Разработка базы данных5

3.2 Разработка программного модуля6

3.3 Текст программы6

1. Руководство пользователя4

4.1 Выполнение программного модуля5

4.2 Сообщения пользователю6

1. Тестирование программного модуля4

5.1 Модульное тестирование5

5.2 Тестовые сценарии6

1. Отладка программного модуля4
2. Заключение4
3. Список литературы4

**Введение:**

Игровое приложение — это объединенный в единое целое комплекс прикладных программ, с помощью которых обеспечивается графическая визуализация, звуковое сопровождение, перемещение внутриигровых персонажей, их действия в соответствии со скриптами.

Данное приложение написано под РПГ-игру.

РПГ (RPG «Role—Playing Game») – знаменитый жанр компьютерных и видео игр, где основой игрового процесса является отогревание определенной роли. Игрок берёт под контроль определённого героя или героиню, с набором стандартных навыков, характеристик и умений.

Играм этого жанра свойственны очень глубокие диалоги, нелинейное прохождение, использование продвинутого инвентаря и большие, и открытые для исследования игровые пространства.

Копнём немного дальше и рассмотрим по отдельности именно персонажей, чему и будет в основном посвящена программа.

Игровой персонаж — это вымышленная фигура в видеоигре, действия которой контролируются игроком, а не правилами игры.

Персонажи, которые не контролируются игроком, называются неигровыми персонажами (NPC - Non-player character).

1. **Назначение разработки**

Данная программа используется в работе с данными части игры, а именно раздел персонажей. Программа предоставляет использовать различные функции: добавление, редактирование, удаление.

Для удобности есть следующие функции: сортировка, фильтрация, поиск.

Программа является частью большой системы (полноценной игры). Следовательно она может быть использована в разработке РПГ-игры.

1. **Требования к программному модулю**

**2.1 Требования к функциональным характеристикам**

В данной области есть сущность «Персонаж», который делится на игрового и НПС. Игровым персонажем управляют, поэтому ему предоставлены оружия и сила. У каждого оружия есть своё название и свой навык. Сила также имеет свою уникальность и название. Они изначально закреплены игрой, поменять их никак нельзя. У НПС – это вражеские фигуры, с которыми игровым персонажем надо сражаться – есть свой уровень сложности.

Базой могут пользоваться разработчики.

Есть разработчик (1) – работает с персонажами и разработчик (2), работающий с оружием. Разработчики имеют право добавлять, изменять, удалять персонажей/оружие, но согласовавшись между собой, т.к. эти области сильно между собой связано.

Для использования программы нужен файл (.exe).

Входные данные берутся из базы данных. Выходные – печать в Excel.

**2.2 Требования к составу и параметрам технических средств**

Таблица 1

|  |  |
| --- | --- |
| Название | Характеристики |
| Процессор | Частота сигналов: 2.80 GHz  Макс. Объем памяти: 8 GB  Кол-во потоков: 2 |
| Оперативная память | Кол-во модулей в комплекте: 1  Объем одного модуля памяти: 8 GB  Тактовая частота: 1600 МГц |
| Разрешение экрана | 1024х768 |
| Жёсткий диск | Емкость: 1 ТБ  Объем буфера: 64 МБ  Потребляемая мощность: 5.3 Вт |
| Устройства ввода | Клавиатура и мышь |
| ОС | Windows |
| Устройство вывода | Монитор |
| Принтер | Макс. Объем памяти: 128  Процессор: 600 МГц  Нагрузка: 15000 стр/мес |

**2.3 Требования к информационной и программной совместимости.**

Для корректной работы программы необходимо: ОС Windows 7/8/8.1/10, Microsoft Excel 2016 (и выше), Microsoft SQL Server Management Studio 18 (и выше).

**3. Руководство программиста**

**3.1 Разработка базы данных**

Прежде чем приступать к созданию базы в SQL Server, были созданы ER-диаграмма и UseCase-диаграмма для понимания предметной области:

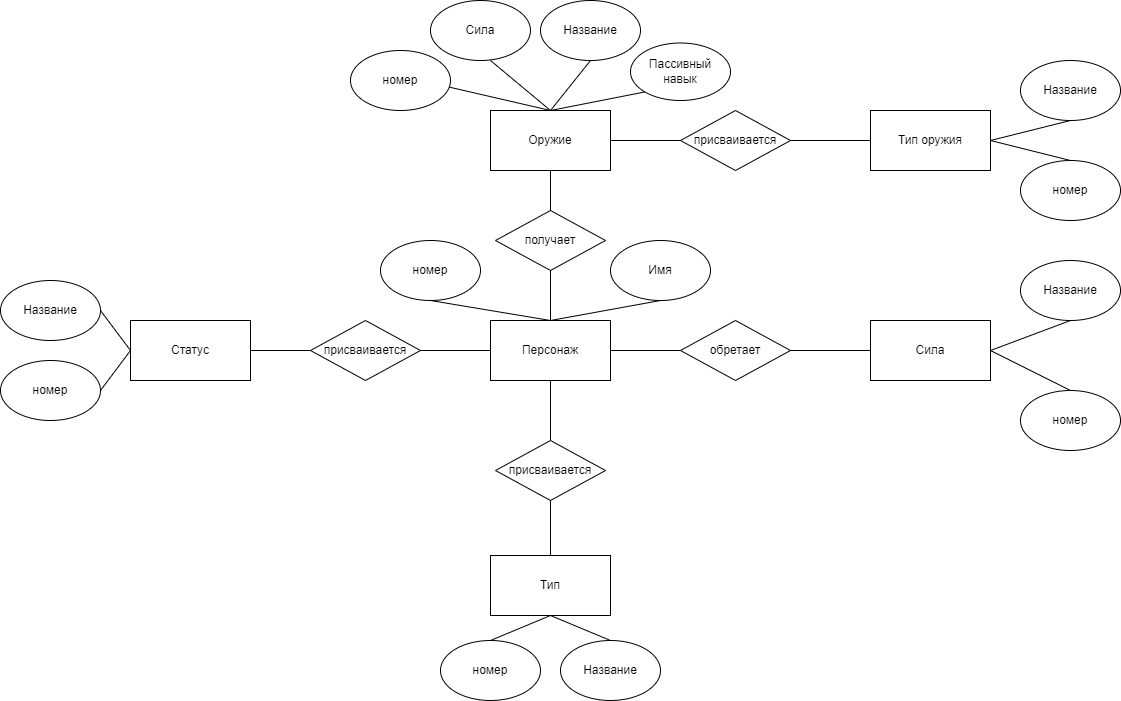


Рисунок 1 «ER-диаграмма»

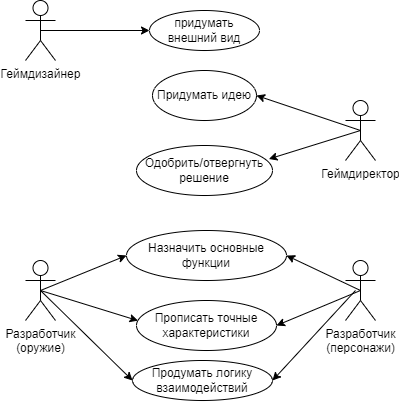


Рисунок 2 «UseCase-диаграмма»

После чего была создана база данных со следующей схемой данных:

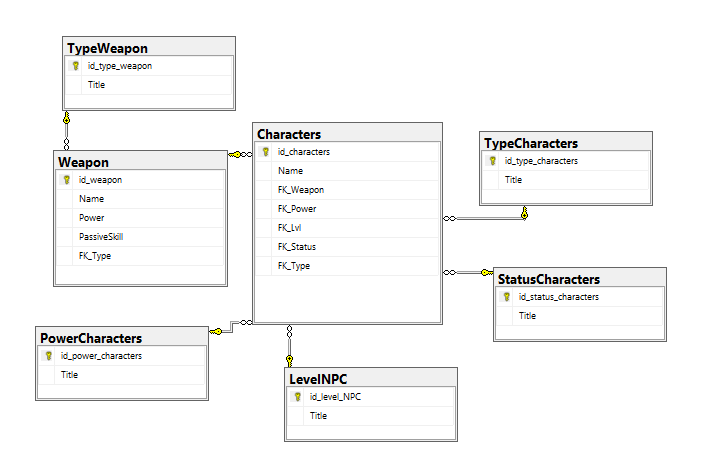
****

Рисунок 3 «Схема данных»

Таблицы имеют следующие типы данных:

Characters:

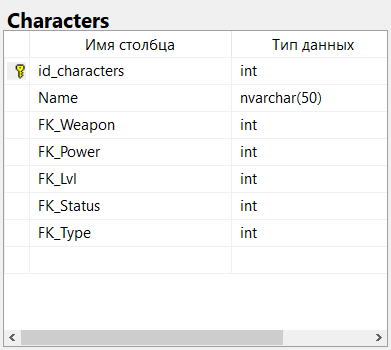


Рисунок 4 «Типы данных таб.Characters»

TypeCharacters:

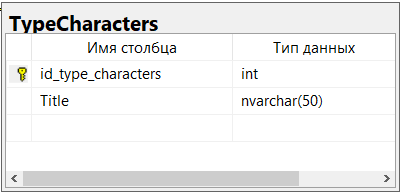


Рисунок 5 «Типы данных таб.TypeCharacters»

PowerCharacters:

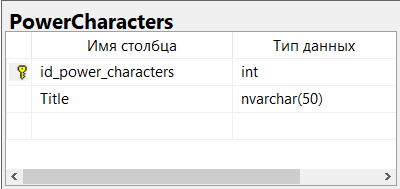


Рисунок 6 «Типы данных таб.PowerCharacters»

StatusCharacters:

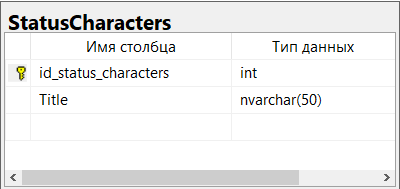


Рисунок 7 «Типы данных таб.StatusCharacters»

LevelNPC:

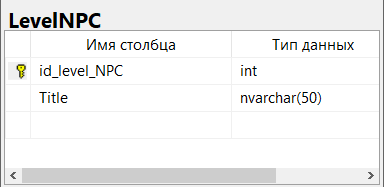


Рисунок 8 «Типы данных таб.LevelNPC»

Weapon:

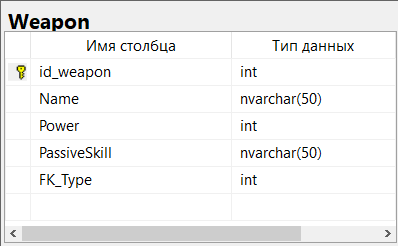


Рисунок 9 «Типы данных таб.Weapon»

TypeWeapon:

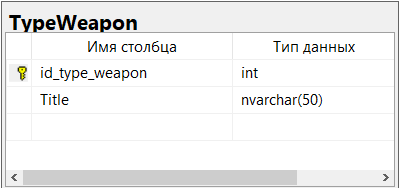


Рисунок 10 «Типы данных таб.TypeWeapon»

**3.2 Разработка программного модуля**

Для разработки программного модуля использовалась программа Visual Studio 2022. Программа писалась на языке C# с помощью «Приложение WPF (.NET Framework)».

Программа включает в себя страницы функционирования с данными:

1. PageCharacters
2. PageLevelNPC
3. PagePower
4. PageStatusChar
5. PageType
6. PageWeapon

Страницы добавления/редактирования данных:

1. AddEditCharacters
2. AddEditLevelNPC
3. AddEditPower
4. AddEditStatus
5. AddEditType
6. AddEditWeapon

Класс:

1. ClassFrame

Подключённую базу данных:

1. ModelBase (технология Entities Framework)
   1. **Текст программы**

После подключения базы данных через Модель ADO.NET EDM следует дополнить код в ModelBase.Context.cs для обработки поля:

private static GameCharactersEntities \_context;

public GameCharactersEntities()

: base("name=GameCharactersEntities")

{

}

public static GameCharactersEntities GetContext()

{

if (\_context == null)

\_context = new GameCharactersEntities();

return \_context;

}

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

throw new UnintentionalCodeFirstException();

}

Класс ClassFrame – элемент управления, поддерживающий навигацию, служит для отображения других элементов проекта.

internal class ClassFrame

{

public static Frame frmObj { get; set; }

}

Следующий код используется повторно во всех страницах проекта, поэтому разберём на примере двух экземплярах: PageCharacters и AddEditCharacters.

Для всех страниц использовались стили. Вот их пример:

<Style x:Key="DataGridStyle1" TargetType="{x:Type DataGrid}">

<Setter Property="ColumnHeaderStyle" Value="{DynamicResource ColumnHeaderStyle1}"/>

</Style>

<Style x:Key="ColumnHeaderStyle1" TargetType="DataGridColumnHeader">

<Setter Property="Height" Value="30"/>

<Setter Property="Foreground" Value="Purple"/>

<Setter Property="FontSize" Value="16"/>

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="FontFamily" Value="Century Gothic"/>

</Style>

<Style x:Key="Button" TargetType="Button">

<Setter Property="Background" Value="LightBlue"/>

<Setter Property="FontWeight" Value="DemiBold"/>

<Setter Property="Height" Value="30"/>

<Setter Property="Width" Value="347"/>

<Setter Property="FontSize" Value="18"/>

</Style>

<Style x:Key="TextBlock" TargetType="TextBlock">

<Setter Property="FontFamily" Value="Century Gothic"/>

<Setter Property="FontSize" Value="13"/>

<Setter Property="Foreground" Value="HotPink"/>

<Setter Property="FontWeight" Value="Bold"/>

</Style>

С помощью DataGrid формируется таблица с данными. Прописывается определенный интерфейс с помощью различных параметров:

<DataGrid Name="DGridCharacter"

Grid.Column="2"

AutoGenerateColumns="False"

IsReadOnly="True" Margin="0,58,19,58"

Background="LightGoldenrodYellow"

BorderBrush="MediumPurple"

BorderThickness="5"

RowBackground="LemonChiffon"

Для отображения каждого столбца с данными используется привязка с помощью Binding ():

<DataGridTextColumn

Header="Имя" Width="\*"

FontSize="17"

FontFamily="Century Gothic"

Binding="{Binding Name}"/>

Если нужно отобразить данные, которые берутся из другой таблицы с помощью вторичного ключа, то код немного меняется (указывается название таблицы и поле, которое нужно вывести):

<DataGridTextColumn

Header="Статус" Width="\*"

FontSize="17"

FontFamily="Century Gothic"

Binding="{Binding StatusCharacters.Title}"/>

IsVisibleChanged="Page\_IsVisibleChanged"

**4. Руководство пользователя**

**4.1 Выполнение программного модуля**

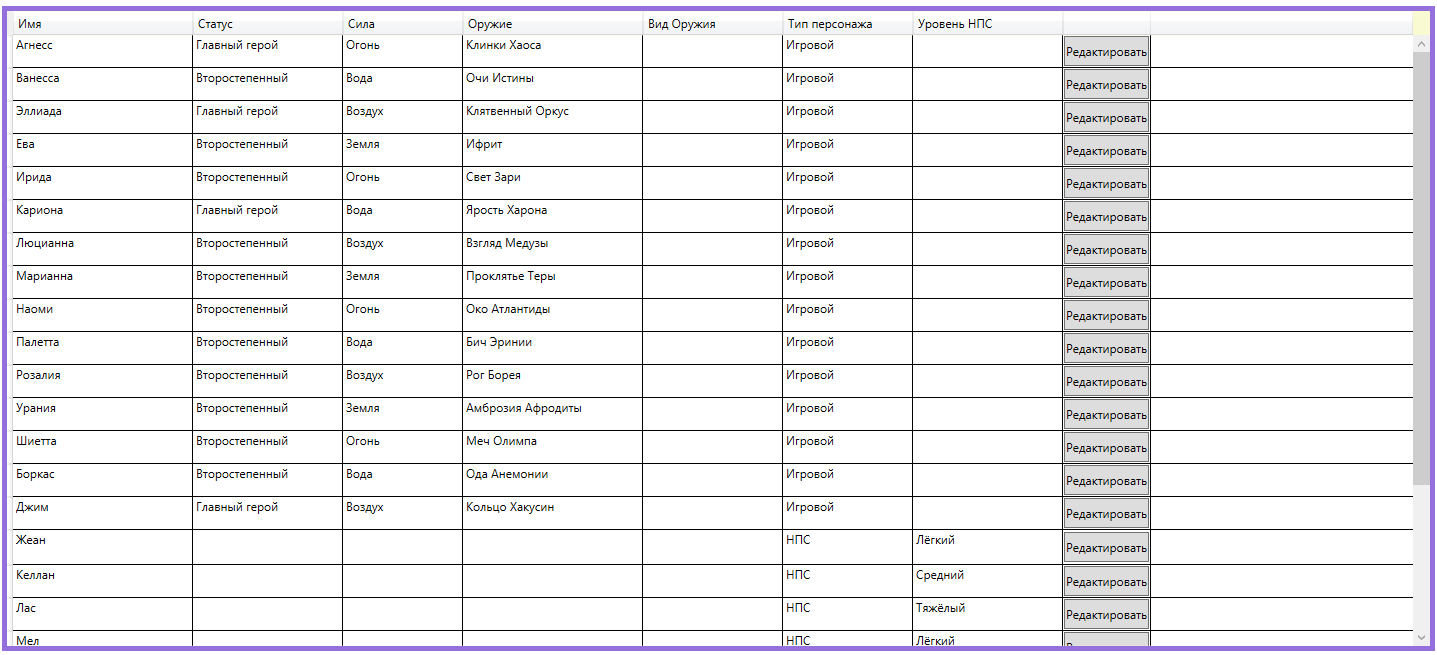
**4.2 Сообщения пользователю**

1. **Тестирование программного модуля**
   1. **Модульное тестирование**
   2. **Тестовые сценарии**
2. **Отладка программного модуля**

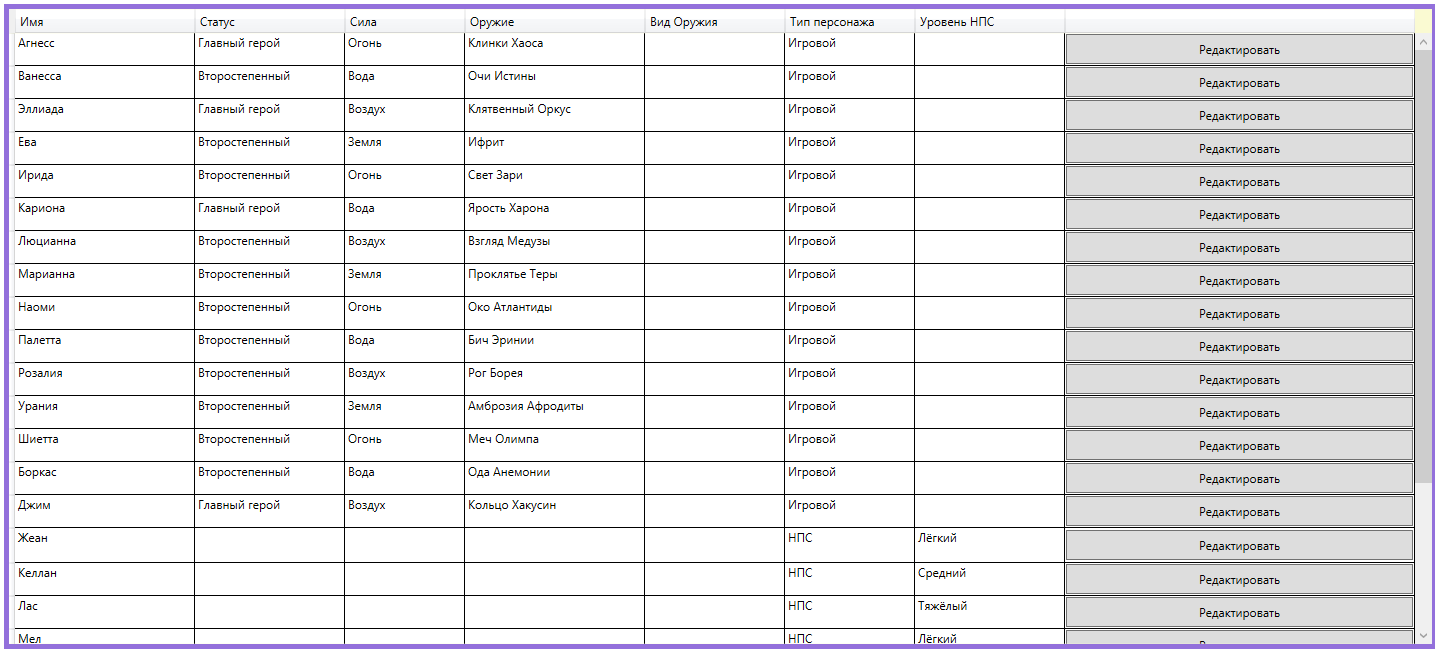
1) Была допущена ошибка при создании проекта, назвав его также, как и базу данных, из-за чего проект не работал. Решением было – создать новый проект с другим названием.

2) Лишний пустой столбец в таблице. Было принято решение избавиться от него, расширив при этом другие столбцы – в данном случае столбец с кнопкой «Редактирование».

До:



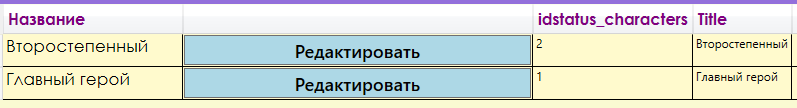
После:



3) При эксперименте с таблицей, была изменена функция «AutoGenerateColumn = True».



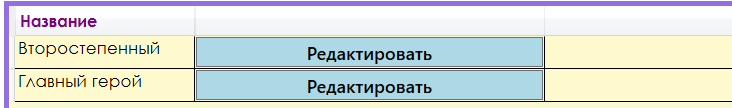
После чего стали высвечиваться дополнительные столбцы с ключами таблиц, чегоне должен видеть пользователь



Эта проблема решилась при изменении данной строчки:



Лишние столбцы пропали:



1. **Заключение**
2. **Список литературы**

## Приложение 2. SQL код создания БД «Auto»

USE [master]

GO

Цвцйвйвцй

Вйцвйцвйвй

Йцвцйвйвйв

Йвйвййв

йвцйвйвйв