**Главное управление по образованию Витебского облисполкома**

**Учреждение образование   
«Полоцкий государственный экономический колледж»**

**О Т Ч Е Т  
по технологической практике**

учащейся 4 курса группы П41  
специальности 2-40 01 01  
 «Программное обеспечение информационных технологий»

Сятковского Николая Васильевича

ООО «Фьюри Лаин Девелопмент»

за период с 03 октября 2022 г. по 26 ноября 2022 г.

|  |  |  |
| --- | --- | --- |
| Руководители практики: |  |  |
| от предприятия  инженер-программист, Мельников Евгений Юрьевич | \_\_\_\_\_\_\_\_\_\_\_  *(подпись)* | \_\_\_\_\_\_\_\_\_\_\_  *(отметка)* |
| от колледжа |  |  |
| преподаватель ИТ,  Шунто Мария Игоревна | \_\_\_\_\_\_\_\_\_\_\_  *(подпись)* | \_\_\_\_\_\_\_\_\_\_\_  *(отметка)* |

Дата сдачи отчета \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Итоговая отметка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Полоцк 2022

СОДЕРЖАНИЕ

[1 РЕЗУЛЬТАТЫ ПРОХОЖДЕНИЯ ОБЩЕЙ ОЗНАКОМИТЕЛЬНОЙ ЧАСТИ ПРАКТИКИ 4](#_Toc120408384)

[1.1 Функции предприятия, его организационная структура 4](#_Toc120408385)

[1.3 Должностные обязанности персонала рабочих мест 4](#_Toc120408387)

[1.4 Компьютерная сеть предприятия 5](#_Toc120408388)

[1.5 Организация взаимодействия с глобальными сетями 6](#_Toc120408389)

[1.6 Эксплуатируемое прикладное ПО 6](#_Toc120408390)

[1.7 Методы и средства разработки ПО 6](#_Toc120408391)

[1.8 Организация электронного документооборота 6](#_Toc120408392)

[1.9 Методы защиты информации 7](#_Toc120408393)

[1.10 Организация устранения сбоев и ошибок ПО и аппаратуры 7](#_Toc120408394)

[2 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ ТЕХНОЛОГИЧЕСКОЙ ПРАКТИКИ 8](#_Toc120408395)

[2.1 Анализ исходных данных и постановка задачи 8](#_Toc120408396)

[2.2 Описание предметной области 9](#_Toc120408400)

[2.3 Сравнительный анализ аналогов и прототипов 11](#_Toc120408401)

[2.4 Разработка технического задания 13](#_Toc120408404)

[2.5 Проектирование программного средства 17](#_Toc120408409)

[2.6 Проектирование структуры данных 17](#_Toc120408410)

[2.7 Проект интерфейса программного средства 18](#_Toc120408411)

[2.8 Реализация и тестирование программного средства 20](#_Toc120408414)

[2.9 Описание реализации программного средства 21](#_Toc120408415)

[2.10 Тестирование программного средства 21](#_Toc120408416)

[2.11 Установка и эксплуатация ПО 21](#_Toc120408417)

[ПРИЛОЖЕНИЕ А Диаграмма вариантов использования 22](#_Toc120408420)

[ПРИЛОЖЕНИЕ Б Техническое задание 24](#_Toc120408421)

[ПРИЛОЖЕНИЕ В Диаграмма Сущность-связь 27](#_Toc120408437)

[ПРИЛОЖЕНИЕ Г Диаграмма Развёртывания 28](#_Toc120408438)

[ПРИЛОЖЕНИЕ Д Программа и методика испытаний 29](#_Toc120408439)

[ПРИЛОЖЕНИЕ Е Результаты тестирования 32](#_Toc120408440)

[ПРИЛОЖЕНИЕ Ж Руководство оператора 33](#_Toc120408441)

[ПРИЛОЖЕНИЕ И Содержание электронного носителя 36](#_Toc120408442)

# РЕЗУЛЬТАТЫ ПРОХОЖДЕНИЯ ОБЩЕЙ ОЗНАКОМИТЕЛЬНОЙ ЧАСТИ ПРАКТИКИ

## Функции предприятия, его организационная структура

FuryLion – молодая и перспективная компания, работающая в сфере IT-технологий. Деятельность организации основывается на создании современного программного обеспечения, компьютерных игр и обучающих программ для ПК, а также комиксов.

Концентрация усилий в данной сфере позволяет компании добиваться серьезных успехов, что сказывается на росте ее популярности. Опытные профессионалы, работающие в FuryLion, генерируют и воплощают в жизнь самые смелые, широкомасштабные и захватывающие идеи. Динамично развивающаяся компания активно сотрудничает со своими потребителями, а потому всегда старается, в первую очередь, ориентироваться на клиентские предложения и пожелания.

Компания FuryLion – перспективная команда единомышленников, работающих во всех областях, необходимых для разработки компьютерных игр и высококачественного программного обеспечения. Талантливые и обладающие необходимым опытом программисты, сценаристы, геймдизайнеры, художники, менеджеры, композиторы и продюсеры создают актуальный и качественный продукт, обеспечивающий компании стремительное развитие.

Ядро FuryLion составляют настоящие профессионалы, благодаря чему продукция компании отвечает всем потребностям своей целевой аудитории, независимо от жанровых предпочтений и возраста клиентов.

## Функции, цели и задачи подразделения

Функциями, целями и задачами отдела программирования Unity разработчики для работы над казуальными играми является:

* разработки игр на Unity3D;
* знание среды разработки Visual Studio (C#);
* владение контролем скорости версии (git);
* владение английским языком.

## Должностные обязанности персонала рабочих мест

Для выполнения функций техник-программист обязан:

* выполнять работу по обеспечению механизированной и автоматизированной обработки поступающей в вычислительный (информационно-вычислительный) центр (ВЦ, ИВЦ) информации, разработки технологии решения экономических и других задач производственного и научно-исследовательского характера;
* принимать участие в проектировании систем обработки данных и систем математического обеспечения машины;
* выполнять подготовительные операции, связанные с осуществлением вычислительного процесса, вестти наблюдение за работой машин;
* составлять простые схемы технологического процесса обработки информации, алгоритмы решения задач, схемы коммутации, макеты, рабочие инструкции и необходимые пояснения к ним;
* разрабатывать программы решения простых задач, проводить их отладку и экспериментальную проверку отдельных этапов работ;
* выполнять работу по подготовке технических носителей информации, обеспечивающих автоматический ввод данных в вычислительную машину, по накоплению и систематизации показателей нормативного и справочного фонда, разработке форм исходящих документов, внесению необходимых изменений и своевременному корректированию рабочих программ;
* участвовать в выполнении различных операций технологического процесса обработки информации (прием и контроль входной информации, подготовка исходных данных, обработка информации, выпуск исходящей документации и передача ее заказчику);
* вести учет использования машинного времени, объемов выполненных работ;

## Компьютерная сеть предприятия

На производстве ООО FuryLion используется локальное соединение компьютеров. Локальная вычислительная сеть (Local Area Network), именуемая в дальнейшем LAN ­ это совокупность компьютеров и других средств вычислительной техники (активного сетевого оборудования, принтеров, факсов, модемов и т. п.), объединенных в вычислительную сеть с помощью кабелей и сетевых адаптеров и работающих под управлением сетевой операционной системы. Вычислительные сети создаются с целью совместного использования общих сетевых ресурсов (дисковое пространство, принтеры, модемы и другая техника), совместную работу с общими базами данных, уменьшении лишних и мешающих работе передвижений внутри помещений. Каждый компьютер в сети оснащается сетевым адаптером, адаптеры соединяются с помощью сетевых кабелей или беспроводных технологий и тем самым объединяют компьютеры в единую вычислительную сеть. Компьютер, подключенный к сети, называется рабочей станцией или сервером, в зависимости от выполняемых их функций. Сеть состоит из 20 (и более) рабочих станций.

## Организация взаимодействия с глобальными сетями

Доступ в интернет производится через подключение по сети Wi-Fi. Скорость интернета ограничена провайдером.

## Эксплуатируемое прикладное ПО

Список софта на ПК пользователей составил:

* Unity, Unreal Engine
* Spine, Kubernetes, Docker, MySQL;
* C#, C++, Java, AWS, GCP, Go, JavaScript;
* CI CD.

## Методы и средства разработки ПО

Метод нисходящего проектирования. Суть метода заключается в определении спецификаций компонентов системы путем последовательного выделения в ее составе отдельных составляющих и их постепенной детализации до уровня, обеспечивающего однозначное понимание того, что и как необходимо разрабатывать и реализовывать.

## Организация электронного документооборота

При подготовке, оформлении документов в электронном виде и их отправке по электронной почте используется следующий порядок:

1. Составление и оформление проекта документа в электронном виде осуществляется исполнителем при помощи компьютера на соответствующем бланке. Документ в электронном виде должен иметь все реквизиты, установленные для аналогичного документа на бумажном носителе.
2. Подготовленный и оформленный проект документа в электронном виде распечатывается на бумажном носителе и представляется руководителю. После подписания (утверждения) руководителем или иными уполномоченными лицами в соответствии с их компетенцией документ на бумажном носителе и его копия в электронном виде направляются лицу, ответственному за электронный документооборот.
3. Ответственный за документооборот в электронном виде вставляет в электронную версию документа в качестве эквивалента собственноручной подписи руководителя сканированную подпись, сохраняет документ в формате PDF и регистрирует его в системе.
4. После регистрации документа в системе, его электронная копия и реквизиты регистрационной карточки отправляются по электронной почте внешнему адресату (сторонней организации).
5. Регистрация и отправка документов в электронном виде осуществляются в день их подписания.
6. Для осуществления контроля за прохождением электронной почты на компьютере работника, ответственного за документооборот в электронном виде, в почтовой программе должны быть включены опции запроса факта получения (прочтения) электронных сообщений с данного почтового ящика, а также отправка подтверждения факта получения (прочтения) в автоматическом режиме.
7. Поступивший по электронной почте документ, если он был отправлен из системы, регистрируется в автоматизированном режиме. Многие из реквизитов регистрационной карточки устанавливаются на основе данных, полученных вместе с электронным документом (в зависимости от установленных отправителем параметров отправки). При необходимости полученные в электронном виде документы распечатываются на бумажном носителе.
8. При хранении и использовании документов в электронном виде, отправленных (полученных) по электронной почте, обеспечивается их доступность только для работников, которым поручено исполнение этих документов.

## Методы защиты информации

На практике используют несколько групп методов защиты, в том числе:

* препятствие на пути предполагаемого похитителя, которое создают физическими и программными средствами;
* управление, или оказание воздействия на элементы защищаемой системы;
* маскировка, или преобразование данных, обычно – криптографическими способами;
* регламентация, или разработка нормативно-правовых актов и набора мер, направленных на то, чтобы побудить пользователей, взаимодействующих с базами данных, к должному поведению;
* принуждение, или создание таких условий, при которых пользователь будет вынужден соблюдать правила обращения с данными;
* побуждение, или создание условий, которые мотивируют пользователей к должному поведению.

## Организация регистрации и устранения сбоев и ошибок ПО и аппаратуры

Для регистрации ошибок на предприятии используется централизованная служба Stackify. Централизованные службы регистрации ошибок помогают просматривать сообщения регистрации во всех приложениях, на всех серверах и во всех средах. В службу Stackify заносится ошибка, присваивается уровень важности и передаётся специалисту.

# РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ ТЕХНОЛОГИЧЕСКОЙ ПРАКТИКИ

## Анализ исходных данных и постановка задачи

### Тема работы

«Игровое приложение «Эко-ферма». Модуль управления данными».

### Исходные данные к проекту

Функциональное назначение системы – автоматизация формирования каталога игровых сущностей и статистики игрового процесса.

Эксплуатационное назначение системы – использование администратором базы данных игрового приложения «Эко-ферма».

#### Требования к пользователям системы

В системе должны быть реализованы уровни доступа к информации под следующими ролями: Администратор, Система.

Количество потенциально-возможных пользователей информационной системы в один момент времени – 1.

#### Функциональные требования

В системе должна быть реализованы возможности:

* Для администратора: формирования каталога игровых сущностей;
* Для системы: формирование статистики игровых сессий на основе каталога и игрового процесса.

В системе должна быть реализована функция поиска информации в каталоге (критерии поиска уточняются в процессе разработки).

##### Отчетная документация должна быть представлена в составе:

* Экспорт результатов поиска в Excel.

### Перечень подлежащих разработке вопросов

#### Анализ исходных данных и постановка задачи.

* Описание предметной области.
* Разработка технического задания.
* Проектирование программного средства
* Проектирование логической структуры данных (результат – ER-диаграмма).
* Проектирование физической структуры данных.
* Проектирование интерфейса программного средства.
* Реализация и тестирование программного средства
* Оформление пояснительной записки и сопроводительной документации КП.

## Описание предметной области

Предметной областью, для которой ведется разработка является индустрия компьютерных игр. [5]

Индустрия компьютерных игр – сектор экономики, связанный с разработкой, продвижением и продажей компьютерных игр. В неё входит большое количество специальностей, по которым работают тысячи человек по всему миру.

Бен Сойер из Digitalmill рассматривает цепочку ценности игровой индустрии, которая составлена из шести связанных, но хорошо различимых слоев:

– уровень издательства и капитала: издательства оплачивают разработку новых проектов и извлекают прибыль посредством лицензирования наименований;

– уровень продукта и талантов: включает в себя разработчиков, дизайнеров, художников, композиторов, которые могут работать как по индивидуальным контрактам, так и в составе группы разработчиков;

– уровень создания и технологий: является источником средств разработки игр, настраиваемых и расширяемых игровых движков, связующего ПО, а также средств управления разработкой;

– уровень распространения: создание перечней игр и их продвижение в розничных и интернет-магазинах;

– уровень аппаратного и программного обеспечения: сюда входят аппаратные базисы-платформы, в том числе такие как консоли и мобильные устройства. В этот уровень сейчас входят также и неаппаратные платформы, такие как виртуальные машины или программные платформы, такие как браузеры или Facebook;

– уровень конечных пользователей. Иначе говоря, потребителей игр или геймеров.

Одновременно с развитием технологий происходило и развитие рынка игр. Так в начале периода стали более успешными лицензированные игры, также как и сиквелы игр.

В 1993 году продажи компьютерных игр в мире составили $19,8 млрд. ($31 млрд. в ценах 2011), $20,8 млрд. в 1994 ($32 млрд. в ценах 2011) и приблизительно $30 млрд. в 1998 ($41,5 млрд. в ценах 2011). Суммарные продажи игровой индустрии США более чем в 2,5 раза превысили продажи кинематографа в США.

В 2000-е было создано и стали популярными множество казуальных и инди-игр. Также крепнет направление игр для мобильных платформ, появилось направление создания игр для социальных сетей. Особенно известен разработчик Zynga игр для социальной сети Facebook. Другим примером успешных платформ для компьютерных игр являются iOS и Android.

На начальном этапе существования компьютерных игр стоимость разработки была минимальной, поэтому это был прибыльный бизнес. Игры, разработанные единственным программистом или небольшой группой, состоящей из программиста и нескольких художников, могли обеспечивать продажи в количестве сотен тысяч копий. Многие из этих игр были разработаны всего за несколько месяцев, что давало возможность разработчикам выпускать по несколько игр в год. Это давало возможность издателям предлагать весьма щедрые отчисления разработчикам, включая различные роялти с проданных копий. В течение этого экономически благоприятного периода было создано много известных компаний-издателей, например, Origin Systems, Sierra Entertainment, Capcom, Activision и Electronic Arts.

На данный момент игровая библиотека только в одном онлайн-сервисе цифрового распространения игр Steam опубликованы десятки тысяч игр от тысяч разработчиков со всего мира.

В настоящее время компьютерные игр вносят значительный вклад в мировую экономику ввиду большого успеха продаж основных игровых систем и игр типа Call of Duty: Black Ops, заработавшая в течение первых 5 дней продаж более $600 млн., что стало мировым рекордом пятидневных продаж среди фильмов, книг и компьютерных игр.

В последние годы набирает популярность модель «пожертвований» (с помощью сервисов аналогичных Kickstarter) для разработки компьютерных игр. Часто известные в прошлом разработчики игр предлагают создать идейное «продолжение» или «аналог» известных игр, если пользователи соберут запрашиваемую сумму. Примерами проектов, наиболее успешно собравших средства с помощью этой схемы, являются Star Citizen (запросили – $2 млн., собрали – $27 млн.), Torment: Tides of Numenera (запросили – $1 млн., собрали – более $4,2 млн.) и Elite: Dangerous (запросили – $1,25 млн., собрали – более $1,7 млн.).

Каждая игра имеет уникальный игровой процесс(геймплей) и относится к одному из игровых жанров: platformer, CCG, shooter, RPG, simulator, MOBA, quest, arcade, racing, puzzle, sandbox, horror и так далее. Они помогают игрокам примерно понимать, чего стоит ожидать от данной конкретной игры. Ведь даже, казалось бы, схожие «аркадные гонки» и «гоночные симуляторы» на самом деле обеспечивают разный игровой опыт, в то время как первые своей целью ставят непосредственно весёлое времяпрепровождение с друзьями и нередко внедряют в игровой процесс всяческие бонусы дабы разнообразить игровой процесс, вторые делают упор на максимально правдоподобную симуляцию опыта от настоящего вождения.

За свой, небольшой, относительно других сфер искусства, срок жизни в игровой индустрии сменилось 8 поколений игровых консолей, на данный момент актуальными являются 5 основных игровых платформ: PC, PlayStation, Nintendo Switch, Xbox и mobile, однако до сих пор можно приобрести и сыграть на устаревших платформах вроде NES или 3DS. Игры могут выпускаться как эксклюзивно для определённой платформы, так и на нескольких сразу.

Немалую долю рынка занимают проекты с упором на онлайн составляющую (мультиплеер). Однако не стоит забывать, что разработка как крупнейших проектов ААА класса, так и мелких indie игр зачастую стоит немалых денег и каждому разработчику приходится самостоятельно определять каким образом заработать денег с игры. Исторически самым первым методом распространения игр является непосредственная их покупка игроками, однако стоит отметить, что игрок, покупая игру, получает её в бессрочную аренду, но не получает полного владения ней, то есть не может создавать цифровые копии купленной игры и продавать их другим людям. Но с развитием интернета стали появляться и игры, которые распространяются по модели «Free to Play», их можно официально скачать и начать играть абсолютно бесплатно, однако в самой игре может присутствовать контент доступный исключительно по внутри игровой покупке. Данная модель распространения пользуется особой популярностью на мобильном рынке. [6]

## Сравнительный анализ аналогов и прототипов

Для формирования требований к разрабатываемой программе необходимо изучить аналоги и прототипы.

Аналогами приложения послужили следующие прототипы:

### MS Excel (<https://www.microsoft.com/ru-ru/microsoft-365/excel>).

Microsoft Excel – табличный процессор, разработанный Microsoft для Windows, macOS, Android и iOS. Он включает в себя возможности расчета или вычислений, графические инструменты, сводные таблицы и язык программирования макросов – VBA. Является частью пакета программного обеспечения Microsoft Office (рисунок 1.1). [7]

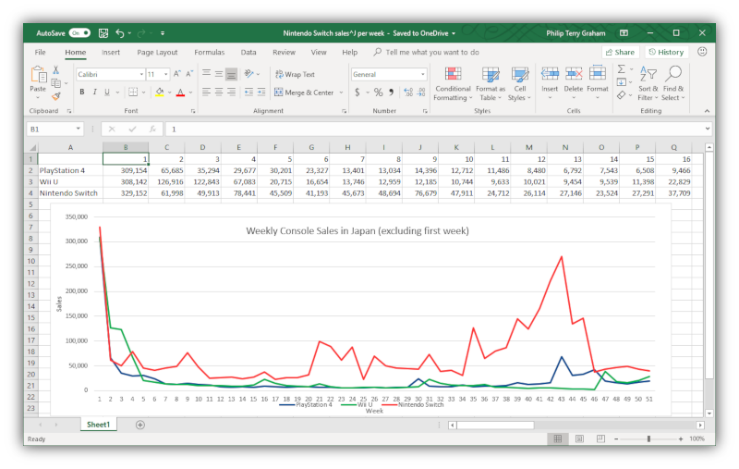


Рисунок 2.1 – Окно программы Microsoft Excel

Достоинства программы:

* мультиплатформенность;
* наличие веб-версии;
* включён в стандартный пакет Microsoft Office;
* синхронизация с облаком.

Недостатки приложения:

* труден в освоении для новых пользователей;
* веб-версия урезана по функционалу;
* высокая стоимость лицензии.

### Google Sheets (<https://www.google.com/sheets/about/>).

Google Sheets – табличный процессор, входящий в состав бесплатного набора веб-редакторов Google Docs Editors, предлагаемого Google. Google Sheets доступна в виде веб-приложения, мобильного приложения для: Android, iOS и настольного приложения в Google ChromeOS (рисунок 1.2).

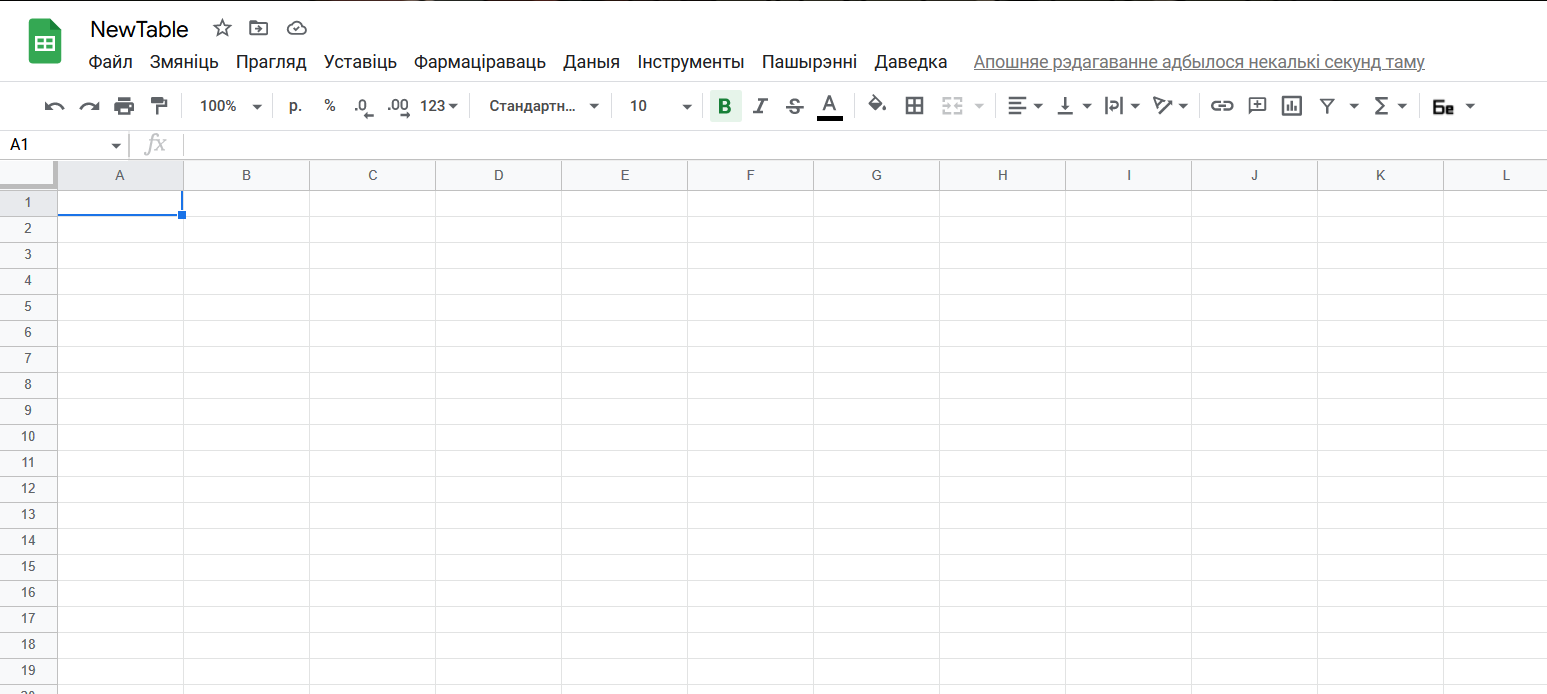


Рисунок 2.2 – Окно программы Google Sheets

Достоинства приложения:

* удобная веб-версия;
* мобильные версии;
* широкие возможности для совместной разработки;
* высокая популярность в управлении игровыми данными;
* бесплатный доступ для пользователя;
* синхронизация с облаком.

Недостатки приложения:

* необходимо думать над структурой данных;
* отсутствие настольной версии для Windows, Linux, Mac.

Проанализировав все аналоги, можно сделать выводы: каждое программное средство имеет свои достоинства и недостатки. Похожих аналогов данной программы не существует, в связи с этим разрабатываться система будет на основе приведённых выше систем. Учитывая их особенности и недостатки, программа «Эко-ферма. Модуль управления данными» будет иметь преимущество перед конкурента в силу своей индивидуальности и предлагаемым требованиями:

* бесплатное использование программы;
* высокая скорость работы;
* простой и понятный интерфейс.

## Разработка технического задания

### Спецификация функций программного средства

Диаграмы вариантов использования описывают взаимоотношения и зависимости между группами вариантов использования и действующих лиц, участвующими в процессе**. [8]**

**Для данной** программы**, в соответствии с диаграммой вариантов использования (ПРИЛОЖЕНИЕ А), необходима реализация следующих функций**:

Для роли Администратор БД:

* **формирование списка игровых сущностей – добавление, редактирование, удаление;**
* **экспорт в Excel –** экспортирование информации о игровых сущностях в виде MS Excel таблицы.

Для роли Система:

– валидация данных – проверка данных на корректность;

– перенос данных в игру – подгрузка информации об игровых сущностях в игровой модуль.

### Обоснование архитектуры программного средства и способа организации данных

Для создания информационной системы можно использовать различные архитектуры. Рассмотрим архитектуру СУБД и её виды.

Системы управления базами данных (СУБД) – это программные средства, пред­назначенные для создания, наполнения, обновления и удаления баз данных.

По своей архитектуре СУБД делятся на одно-, двух- и трехзвенные. В однозвенной архитектуре используется единственное звено (клиент), обеспечиваю­щее необходимую логику управления данными и их визуализацию. В двухзвенной архитектуре значительную часть логики управления данными берет на себя сервер БД, в то время как клиент в основном занят отображением данных в удобном для пользователя виде. В трехзвенных СУБД используется промежуточное звено – сервер приложений, являющееся посредником между клиентом и серве­ром БД. Сервер приложений призван полностью избавить клиента, от каких бы то ни было забот по управлению данными и обеспечению связи с сервером БД.

В архитектуре «клиент-сервер» сервер базы данных не только обеспечивает доступ к общим данным, но и берет на себя всю обработку этих данных. Клиент посылает на сервер запросы на чтение или изменение данных, которые формулируются на языке SQL. Сервер сам выполняет все необходимые изменения или выборки, контролируя при этом целостность и согласованность данных, и результаты в виде набора записей или кода возврата посылает на компьютер клиента.

**Архитектура данной программы файл-серверная, так как все данные хранятся в определённом разделе в реестре компьютера пользователя.**

### Характеристика инструментов реализации

#### Обоснование выбора СУБД

Для хранения данных и их синхронизации для всех пользователей в системе должны использоваться системы управления базами данных.

Были расмотрены следующие варианты СУБД: Microsoft SQL Server, SQLite.

Microsoft SQL Server – СУБД, разработанная корпорацией Microsoft. Основной используемый язык запросов – Transact-SQL, создан совместно Microsoft и Sybase.

Достоинство SQL Server:

* рутинные административные задачи автоматизированы: это управление блокировками, памятью, редактура размеров файлов;
* реализован поиск по фразам, тексту, словам, можно создавать ключевые индексы;
* в систему интегрирован сервер интерактивного анализа для принятия решений, создания корпоративных отчетов;
* запросы можно формулировать на английском языке, без программирования;
* СУБД поддерживает работу с другими продуктами Microsoft: Access, MS Excel.

SQLite – это механизм базы данных, написанный на языке программирования C. Это не отдельное приложение; скорее, библиотека, которую разработчики программного обеспечения встраивают в свои приложения. Таким образом, он принадлежит к семейству встроенных баз данных. Это наиболее широко распространенный движок базы данных, так как он используется несколькими ведущими веб-браузерами, операционными системами, мобильными телефонами и другими встроенными системами.

Достоинства SQLite:

* Высокая скорость работы;
* Хранение данных в одном файле;
* Надежность;
* Доступность;
* Кроссплатформенность;
* Автономность;
* Малый размер.

Для разработки программы «Эко-ферма. Модуль управления данным» использовалось СУБД SQLite, поскольку данная СУБД легко устанавливается для создания файл-серверной архитектуры и имеет интеграцию с Unity. [9]

#### Обосновани**е выбора языка программирования**

Для разработки программы «Эко-ферма. Модуль управления данным» использовался язык программирования C#. [10]

Язык программирования C# является объектно-ориентированным. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML. Разработан в 1998 – 2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270. Позволяет в автоматическом режиме очистить память от объектов, которые не используются, или уничтоженных приложений. В языке принята общая система работы с типами, начиная от примитивов и заканчивая сложными, в том числе, пользовательскими наборами. Применяется единый набор операций для обработки и хранения значений типизации. Также можно использовать ссылочные типы пользователя, что позволит динамически выделить память под объект или хранить упрощенную структуру в сети.

В процессе выбора технологии работы с базой данных были рассмотрены:

* Entity Framework;
* Dapper;
* ADO.NET.

Entity Framework – объектно-ориентированная технология доступа к данным, является object-relational mapping решением для .NET Framework от Microsoft. Предоставляет возможность взаимодействия с объектами как посредством LINQ в виде LINQ to Entities, так и с использованием Entity SQL.

Dapper – технология сопоставления (маппинга) результатов sql-запросов с классами c#. В этом плане Dapper немного похож на Entity Framework. В то же время за счет своей легковесности Dapper обеспечивает большую производительность и быстрее позволяет выполнять запросы, нежели Entity Framework.

ADO.NET – это технология доступа к данным из Microsoft .NET Framework, которая обеспечивает связь между реляционными и нереляционными системами через общий набор компонентов.

Основной средой разработки была выбрана JetBrains Rider ввиду ее функциональности, надежности и предпочтения разработчика.

Rider – кроссплатформенная интегрированная среда разработки программного обеспечения для платформы .NET, разрабатываемая компанией JetBrains. Поддерживаются языки программирования C#, VB.NET и F#.

### Обоснование эксплуатационных требований

Разрабатываемое программное средство предполагает эксплуатацию под операционной системной не меньше Windows 10, программная платформа NET. Framework 4.8.

Минимальные аппаратные требования к компьютеру пользователя следующие:

– процессор: «Intel Core 2 Duo» и новее;

– ОЗУ: 4 ГБ для 32-разрядной системы и 4 ГБ для 64-разрядной системы;

– место на жестком диске: не менее 500 МБ;

– видеоадаптер: DirectX 9 или более поздняя версия.

## Проектирование программного средства

## Проектирование структуры данных

ER-модель – [модель данных](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85" \o "Модель данных), позволяющая описывать [концептуальные схемы](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%86%D0%B5%D0%BF%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D1%85%D0%B5%D0%BC%D0%B0" \o "Концептуальная схема) [предметной области](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%BC%D0%B5%D1%82%D0%BD%D0%B0%D1%8F_%D0%BE%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C). [11]

ER-модель используется при концептуальном [проектировании баз данных](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%B1%D0%B0%D0%B7_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85" \o "Проектирование баз данных). С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

Во время [проектирования баз данных](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%B1%D0%B0%D0%B7_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85" \o "Проектирование баз данных) происходит преобразование ER-модели в конкретную [схему 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" \o "Схема базы данных) на основе выбранной модели данных ([реляционной](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%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85" \o "Реляционная модель данных), [объектной](https://ru.wikipedia.org/w/index.php?title=%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85&action=edit&redlink=1), [сетевой](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) или др.).

ER-модель представляет собой формальную конструкцию, которая сама по себе не предписывает никаких графических средств её визуализации. В качестве стандартной графической нотации, с помощью которой можно визуализировать ER-модель, была предложена диаграмма «сущность-связь».

Для программы «Эко-ферма. Модуль управления данными» была разработана диаграмма «Сущность-связь», которая представлена в ПРИЛОЖЕНИИ В.

На диаграмме, разработанной для данной программы, выделены следующие сущности: Объект развития (ОР), ОР на старте уровня, уровень, Цель, Прогресс игрока, Игрок, Цель по ОР, Здание, Выходные продукты, Продукт, Входные продукты, Дерево, Производственное здание, Генератор, Параметр окружающей среды, Ресурс, Ресурс здания.

Связи в данной диаграмме являются:

* + ОР – уровень: один уровень может иметь несколько ОР и один ОР может быть на нескольких уровнях;
  + Игрок – Уровень: Игрок может пройти несколько уровней и один уровень может быть пройден несколькими игроками;
  + Цель – Уровень: один уровень может содержать одну и более целей;
  + Цель – ОР: одна цель может быть по нескольким ОР, один ОР может содержаться в нескольких целях;
  + Дерево – Продукт: каждое дерево производит один продукт, один продукт может быть произведён разными деревьями;
  + Здание – Производство: каждое производство является зданием, но не каждое здание является производством;
  + Здание – Генератор: каждый генератор является зданием, но не каждое здание является генератором;
  + Производство – Продукты: Одно производство принимает и выдаёт несколько продуктов;
  + Производство – Ресурс: каждое производство может использовать несколько ресурсов, один ресурс может использоваться на нескольких производствах;
  + Генератор – Ресурс: каждый генератор генерирует один ресурс, каждый ресурс может быть сгенерирован несколькими генераторами.

Таблица 2.6.1 – Структура сущности «Уровень»

|  |  |  |
| --- | --- | --- |
| Название поля | Заполнение | Описание атрибута |
| ID\_Уровня | Обязательное (ключевое поле) | Первичный ключ, идентификатор |
| Название уровня | Обязательное | Название уровня |
| Описание уровня | Обязательное | Описание уровня |
| Состояние уровня | Обязательное | Состояние уровня |

Таблица 2.6.2 – Структура сущности «Цель»

|  |  |  |
| --- | --- | --- |
| Название поля | Заполнение | Описание атрибута |
| ID\_Цели | Обязательное (ключевое поле) | Первичный ключ, идентификатор |
| ID\_Уровня | Обязательное | Внешний ключ, идентификатор |
| Название уровня | Обязательное | Название уровня |
| Описание уровня | Обязательное | Описание уровня |
| Состояние уровня | Обязательное | Состояние уровня |

Таблица 2.6.3 – Структура сущности «Ресурс»

|  |  |  |
| --- | --- | --- |
| Название поля | Заполнение | Описание атрибута |
| ID\_Ресурса | Обязательное (ключевое поле) | Первичный ключ, идентификатор |
| Наименование | Обязательное | Наименование |

Таблица 2.6.4 – Структура сущности «Статистика игры»

|  |  |  |
| --- | --- | --- |
| Название поля | Заполнение | Описание атрибута |
| ID\_Уровня | Обязательное | Внешний ключ, идентификатор |
| ID\_Цели | Обязательное | Внешний ключ, идентификатор |
| Значение | Обязательное | Значение |

## Проект интерфейса программного средства

В данном пункте пояснительной записки представлен интерфейс компьютерной игры в виде: главного меню, меню игрового интерфейса.

Таблица 2.3 – Перечень экранных форм

|  |  |
| --- | --- |
| № формы | Название |
| N1 | Главная форма |
| N2 | Окно импорта/экспорта БД |
| N3 | Окно сериализации |

### Граф переходов состояний интерфейса

В данном пункте представлены графы переходов по экранным формам для Игрока (рисунок 2.3).

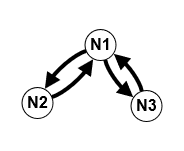


Рисунок 2.3 – Граф переходов администратора БД

### Проектирование экранных форм

Для данной информационной системы был спроектирован интерфейс программы (рисунки 2.4 – 2.6).



Рисунок 2.4 – Главная форма



Рисунок 2.5 – Окно Импорта/Экспорта БД



Рисунок 2.6 – Окно сериализации

## Реализация и тестирование программного средства

Полный исходный код программы доступен в репозитории GitHub. [12]

Создана главная форма с элементами управления (Рисунок 2.7):

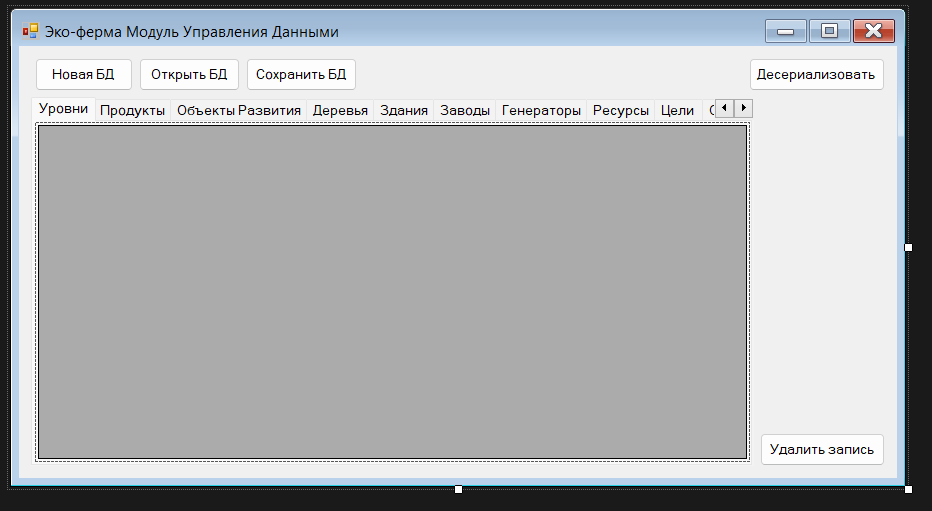


Рисунок 2.7 – Главная форма

Реализовано открытие файла базы данных (Рисунок 2.8):

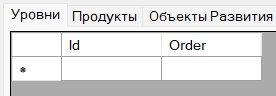


Рисунок 2.8 – Открытая БД

Реализован импорт данных из БД (Рисунок 2.9):

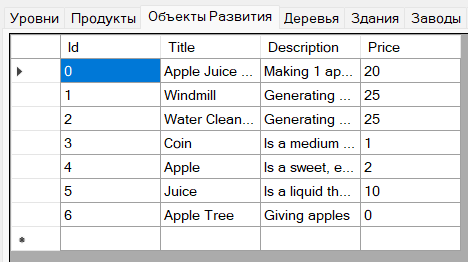


Рисунок 2.9 – Импортированые данные из БД

Реализована сериализация (Рисунок 2.10)

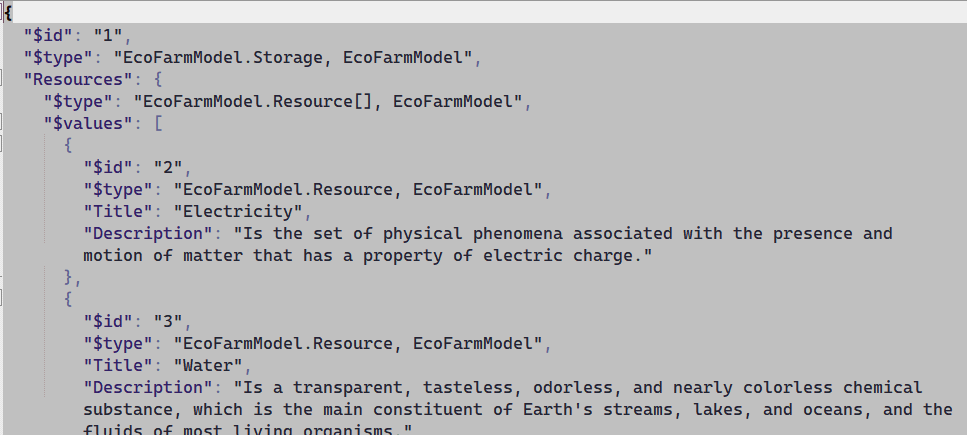


Рисунок 2.10 – Сериализованный файл в формате json

## Описание реализации программного средства

## Тестирование программного средства

Тестирование программного обеспечения – процесс исследования, испытания программного обеспечения (ПО) с целью получения информации о качестве продукта.

С целью исключения возможных ошибок в работе, программа должна пройти функциональное тестирование.

Функциональное тестирование является одним из ключевых видов тестирования, задача которого – установить соответствие разработанного программного обеспечения исходным функциональным требованиям заказчика. Тестирование приложения выполнено согласно документу «Программа и методика испытаний» (ПРИЛОЖЕНИЕ Е).

Результаты функционального тестирования представлены в ПРИЛОЖЕНИИ Д.

## Установка и эксплуатация ПО

Эксплуатационное назначение системы – использование администратором базы данных игрового приложения «Эко-ферма».

Для установки программного средства необходимо распаковать файлы программы в любое удобное место на компьютере.

На самом компьютере должны быть установлены библиотеки .NET Framework 4.5 или выше.

Минимальные требования изложены в Техническом задании в ПРИЛОЖЕНИИ Б.

Процесс эксплуатации программного продукта описан в документе «Руководство оператора» в ПРИЛОЖЕНИИ Ж.

## Выводы по второму разделу

В данном разделе была проанализирована предметная область, проанализированы аналоги и прототипы. Был обоснован выбор инструментальных средств разрабатываемого приложения: в качестве среды разработки были выбраны Unity и Rider, определена локальная архитектура программы, спроектирована логическая и физическая модель данных. Разработана диаграмма «Сущность-связь», описана структура разрабатываемого программного средства и структура хранения данных, проведено тестирование программного средства согласно плану тестирования (ПРИЛОЖЕНИЕ Д).

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению: ГОСТ 19.201–78 ЕСПД. Введ. 01.01.1980 – М.: Издательство стандартов, 1987. – 8 с.
2. Единая система программной документации. Руководство системного программиста. Требования к содержанию и оформлению: ГОСТ 19.503–79 ЕСПД. Введ. 01.01.1980 – М.: Издательство стандартов, 1982. – 8 с.
3. Единая система программной документации. Руководство оператора. Требования к содержанию и оформлению: ГОСТ 19.505–79 ЕСПД. Введ. 01.01.1980 – М.: Издательство стандартов, 1982. – 8 с.
4. Единая система программной документации. Программа и методика испытаний. Требования к содержанию, оформлению и контролю качества: ГОСТ 19.301-2000 ЕСПД. Введ. 01.09.2001 – Минск: Межгосударственный совет по стандартизации, метрологии и сертификации, 2000. – 9 с.
5. Джейсон, Прайс; Майк, Гандэрлой Visual C# .NET. Полное руководство; КОРОНА принт, 2004. - 960 c.
6. Калбертсон Р, Браун К., Кобб Г. Быстрое тестирование: Пер. с англ.. – М.: Издательский дом «Вильямс», 2002.– 384 с.: ил.
7. Культин Н. Б. Самоучитель C# Builder. – СПб.: БХВ-Петербург, 2004.
8. Леоненков А.В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2006, 319 с
9. Орлов С. Технологии разработки программного обеспечения. Разработка сложных программных систем. Учебное пособие. СПб: Питер, 2003. 480 с, ил.
10. Фаронов В.В. Программирование на языке С# – ООО "Питер-Пресс", 2007 г. Учебное пособие, 241 стр.
11. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. – СПб.: Питер, 2002. – 496 с.: ил.
12. Проект на GitHub [Электронный ресурс] – режим доступа: <https://github.com/rapushka/BdKursach> – Дата доступа: 27.11.2022

# ПРИЛОЖЕНИЕ А (обязательное) Диаграмма вариантов использования



Рисунок А.1 – Диаграмма вариантов использования

# ПРИЛОЖЕНИЕ Б (обязательное) Техническое задание

## Б Введение

Настоящее техническое задание распространяется на разработку программы «Эко-Ферма. Модуль управления данными».

## Б1 Основания для разработки

Основанием для разработки является индивидуальное задание на выполнение на технологической практике, выданное учащемуся группы П41 Сятковскому Николаю Васильевичу.

Тема работы: «Эко-Ферма. Модуль управления данными».

## Б2 Назначение разработки

## Б2.1 Функциональное назначение

Функциональное назначение системы – автоматизация формирования каталога игровых сущностей и статистики игрового процесса.

**Б2.2 Эксплуатационное назначение**

Эксплуатационное назначение системы – использование администратором базы данных игрового приложения «Эко-ферма».

## Б3 Требования к программе

### Б3.1 Требования к функциональным характеристикам

В системе должны быть реализованы уровни доступа к информации под следующими ролями: Администратор, Система.

Количество потенциально-возможных пользователей информационной системы в один момент времени – 1.

### Б3.2 Требования к надежности

* предусмотреть контроль вводимой информации;
* предусмотреть защиту от некорректных действий пользователя;
* обеспечить целостность информации в локальной системе хранения данных.

### Б3.3 Условия эксплуатации

Система должна обеспечивать следующие условия эксплуатации:

– условие эксплуатации в соответствии с СанПин 2.2.2.542 – 96;

– предусмотреть защиту от некорректных действий пользователя;

– обеспечить целостность информации в базе данных.

### Б3.4 Требования к составу и параметрам технических средств

– процессор: «Intel Core 2 Duo» и новее;

– ОЗУ: 4 ГБ для 32-разрядной системы и 4 ГБ для 64-разрядной системы;

– место на жестком диске: не менее 500 МБ;

– видеоадаптер: DirectX 9 или более поздняя версия.

### Б3.5 Требования к информационной и программной совместимости

Приложение должно быть установлено на компьютер, содержащий:

* операционная система: Windows 10, 11;
* .NET Framework 4.5 или выше.

### Б3.6 Требования к маркировке и упаковке

Специальные требования к маркировке и упаковке не предъявляются.

## Б4 Требования к программной документации

Программная документация должна быть представлена в следующем составе:

* «Руководство оператора» — ГОСТ 19.505-79;
* «Программа и методика испытаний» — ГОСТ 19.301-2000

## Б5 Технико-экономические показатели

Технико–экономическое обоснование разработки не производится.

## Б6 Стадии и этапы разработки

Разработка программы включает в себя следующие стадии, представленные в таблице:

Таблица – Стадии разработки программы

|  |  |
| --- | --- |
| Наименование этапов курсового проектирования | Срок выполнения |
| 1 | 2 |
| 1. Описание предметной области | до 25.09.2022 |
| 1. Разработка диаграммы «Сущность-связь» | до 25.09.2022 |
| 1. Разработка диаграммы вариантов использования | до 30.09.2022 |
| 1. Разработка технического задания | до 15.10.2022 |
| 1. Разработка 1-го раздела ПЗ КП | до 20.10.2022 |
| 1. Проектирование интерфейса программного средства | до 23.10.2022 |
| 1. Обоснование выбора инструментальных средств | до 23.10.2022 |
| 1. Разработка физической модели данных | до 30.10.2022 |
| 1. Разработка 2-го раздела ПЗ КП | до 10.11.2022 |
| Продолжение табл. | |
| 1 | 2 |
| 1. Реализация программного средства | до 25.11.2022 |
| 1. Разработка программного документа «Программа и методика испытаний», тестирование ПО | до 30.11.2022 |
| 1. Разработка эксплуатационной документации | до 30.11.2022 |
| 1. Разработка 3-го раздела ПЗ КП | до 10.12.2022 |
| 1. Оформление курсового проекта (ПЗ, чертежи, электронный носитель) | до 14.12.2022 |

## Б7 Порядок контроля и приёмки

Б7.1 Порядок контроля

Контроль выполнения осуществляется руководителем практики от предприятия.

Б7.2 Порядок защиты

Защита осуществляется перед руководителем практики.

Б7.3 Срок защиты

Срок защиты определяется графиком прохождения технологической практики.

Примечание

В процессе выполнения работы возможно уточнение отдельных требований технического задания по взаимному согласованию преподавателя и исполнителя.

# ПРИЛОЖЕНИЕ В (обязательное) Диаграмма Сущность-связь

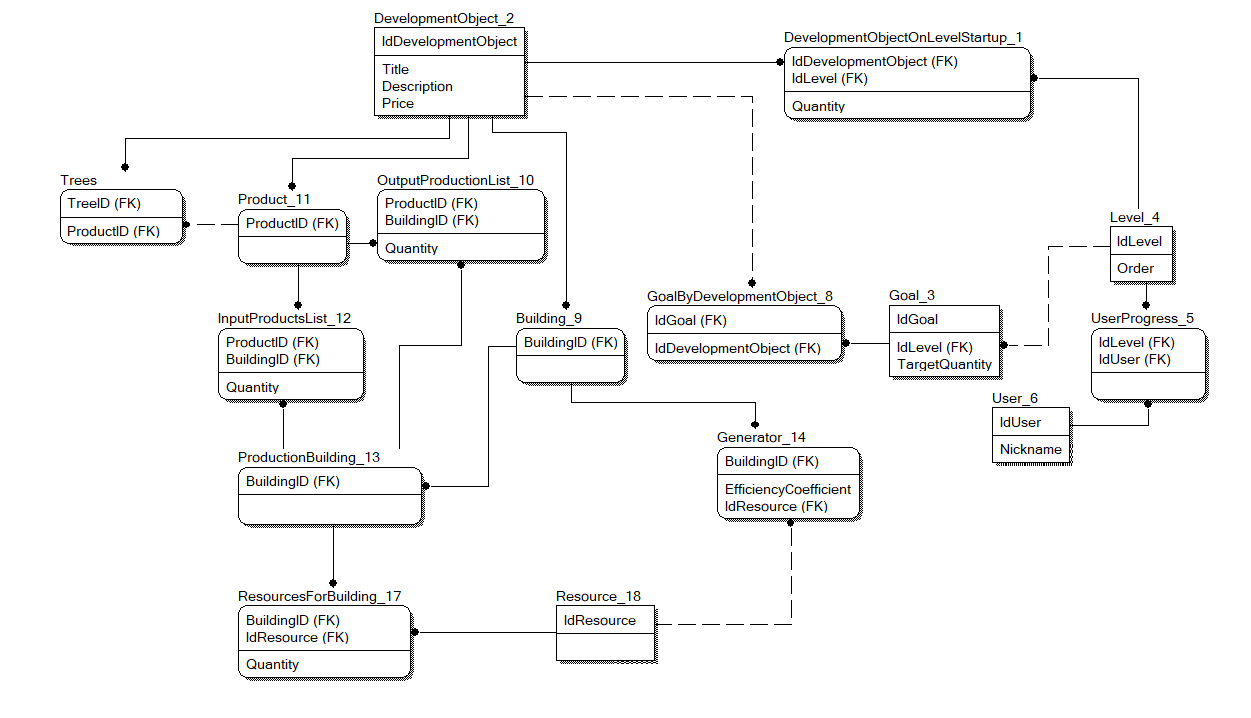


Рисунок В.1 – Диаграмма сущность связь

# ПРИЛОЖЕНИЕ Г (обязательное) Диаграмма Развёртывания



Рисунок Г.1 – Диаграмма развёртывания

# ПРИЛОЖЕНИЕ Д (обязательное) Программа и методика испытаний

**Д1 Объект испытаний**

Объектом испытания является программа «Эко-ферма. Модуль управления данными».

Пользователями системы могут быть администратор БД.

**Д2 Цель испытаний**

Целью проведения испытаний является установление работоспособности разработанной информационной системы в соответствии техническому заданию и другим программным документам. При выявлении несоответствий работы приложения техническому заданию, либо ошибок в программной части, требуется доработка. Тестирование программы должно:

* подтвердить работоспособность системы в соответствии с техническим заданием на разработку;
* выявить все ошибки, исключить недоработки и несоответствия техническому заданию.

Таким образом, главной целью испытаний является выявление как логических, так и функциональных ошибок программы.

**Д3 Требования к программе**

**Д3.1 Требования к функциональным характеристикам**

В системе должны быть реализованы уровни доступа к информации под следующими ролями: Администратор, Система.

Количество потенциально-возможных пользователей информационной системы в один момент времени – 1.

**Д4 Требования к надежности**

* предусмотреть контроль вводимой информации;
* предусмотреть защиту от некорректных действий пользователя;
* обеспечить целостность информации в локальной системе хранения данных.

**Д5 Требования к программной документации**

Разработанная конфигурация должна быть укомплектована следующими программными документами:

* «Программа и методика испытаний»;
* «Руководство оператора».

Программная документация должны быть оформлена в соответствии со следующими нормативными документами:

* «Руководство оператора» – ГОСТ 19.505-79;
* «Программа и методика испытаний» – ГОСТ 19.301-2000.

**Д6 Средства и порядок испытаний**

**Д6.1 Средства испытаний**

Программа должна функционировать на персональных компьютерах с минимальной конфигурацией технических средств для клиентского приложения:

Требования к составу и параметрам технических средств:

– процессор: «Intel Core 2 Duo» и новее;

– ОЗУ: 4 ГБ для 32-разрядной системы и 4 ГБ для 64-разрядной системы;

– место на жестком диске: не менее 500 МБ;

– видеоадаптер: DirectX 9 или более поздняя версия.

**Д6.2 Порядок испытаний**

1. Проверка формирования каталога Объектов развития:

* импорт;
* добавление;
* удаление;
* редактирование.

1. Формирование каталога уровней:

* импорт;
* добавление;
* удаление;
* редактирование.

1. Формирование каталога целей:

* импорт;
* добавление;
* удаление;
* редактирование.

1. Формирование каталога зданий:

* импорт;
* добавление;
* удаление;
* редактирование.

1. Импорт базы данных.
2. Экспорт базы данных.

9) Сериализация файлов.

**Д6.3 Методы испытаний**

Тестирование программного продукта должно проводится на исправном оборудовании, с набором программных средств, заявленными в настоящем документе и техническом задании. При тестировании сверяется ожидаемый и фактический результат, если результаты совпадают, то испытание считается успешным, если возникают ошибки, то испытание считается неуспешным. Для данного программного модуля проводится функциональное тестирование методом «черного» ящика для которого используется тест-план, представленный в таблице Д.1. Сценарий тестирования ссылается на пункты в «Руководство оператора» в ПРИЛОЖЕНИИ Ж.

Таблица Д.1 – Тест-план функционального тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № теста | № тест требования | Описание теста | Входные данные | Ожидаемый результат | Сценарий тестирования |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 1.1 | Авторизация Администратора |  | Открытая главная форма | Пункт Ж1 |
| 2 | 2.1 | Добавление записи | Порядковый номер – 1 | Запись добавлена | Пункт Ж4 |
| 3 | 2.1 | Добавление некорректной записи | Порядковый номер – АБВ | Сообщение об ошибке | Пункт Ж5 |
| 4 | 2.2 | Редактирование записи | Order – 10 | Запись изменена | Пункт Ж6 |
| 5 | 2.3 | Удаление записи | Запись с номером – 1 | Запись удалена | Пункт Ж7 |
| 6 | 2.3 | Попытка удаления записей без выбора |  | Кнопка удаления не доступна | Пункт Ж8 |
| 7 | 3.1 | Сохранение БД |  | Данные занесены в файл БД | Пункт Ж9 |
| 8 | 3.1 | Попытка сохранить неоткрытую БД |  | Сообщение об ошибке | Пункт Ж10 |
| 9 | 4.2 | Сериализация | Раположение: рабочий стол | Сериализованный файл | Пункт Ж12 |

# ПРИЛОЖЕНИЕ Е (обязательное) Результаты тестирования

Таблица – Результаты тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тестовый случай | Входные данные | Ожидаемый результат | Фактический результат | Результат тестирования |
| 1 | 2 | 3 | 4 | 5 |
| Авторизация Администратора |  | Открытая главная форма | Открытая главная форма | Тест пройден |
| Создание Нового файла БД | Расположение: рабочий стол | Открытый файл БД | Открытый файл БД | Тест пройден |
| Импорт файла БД | Расположение: рабочий стол | Открытый файл БД | Открытый файл БД | Тест пройден |
| Добавление записи | Id – 0;  Order – 1 | Запись добавлена | Запись добавлена | Тест пройден |
| Добавление некорректной записи | Id – АБВ | Сообщение об ошибке | Сообщение об ошибке | Тест пройден |
| Редактирование записи | Order – 10 | Запись изменена | Запись изменена | Тест пройден |
| Удаление записи | Запись с Id – 1 | Запись удалена | Запись удалена | Тест пройден |
| Попытка удаления записей без выбора |  | Сообщение об ошибке | Сообщение об ошибке | Тест пройден |
| Сохранение БД |  | Данные занесены в файл БД | Данные занесены в файл БД | Тест пройден |
| Попытка сохранить неоткрытую БД |  | Сообщение об ошибке | Сообщение об ошибке | Тест пройден |
| Десериализация файла | Раположение: рабочий стол | Данные подгрузились | Данные подгрузились | Тест пройден |

# ПРИЛОЖЕНИЕ Ж (обязательное) Руководство оператора

**Ж1 Назначение программы**

Программа «Эко-Ферма. Модуль управления данными» разработана в соответствии с индивидуальным заданием.

В программе предусмотрены такие категории пользователей, как администратор БД.

**Ж2 Условия выполнения программы**

Программа должна функционировать на персональных компьютерах с минимальной конфигурацией технических средств для клиентского приложения:

– процессор: «Intel Core 2 Duo» и новее;

– ОЗУ: 4 ГБ для 32-разрядной системы и 4 ГБ для 64-разрядной системы;

– место на жестком диске: не менее 500 МБ;

– видеоадаптер: DirectX 9 или более поздняя версия.

**Ж3 Выполнение программы**

**Ж3.1 Запуск программы**

Для запуска программы требуется выполнить двойной клик левой кнопкой мыши по исполняемому файлу «EcoFarmAdmin.exe».

**Ж3.2 Роль «Администратор»**

После запуска программы под ролью «Администратор», пользователь попадает на главную форму, где может просматривать каталог игровых сущностей. Форма показа на рисунке Ж1.

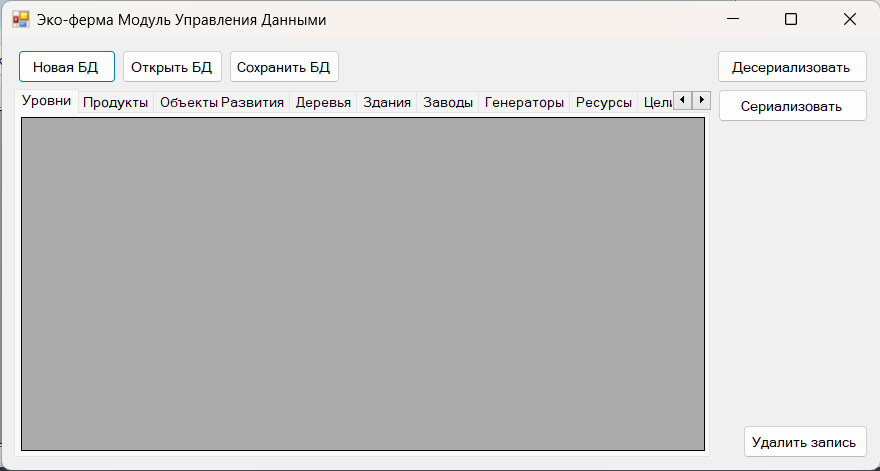
****

Рисунок Ж1 – Главное окно администратора

Для работы с программой необходим файл Базы данных. Если его нет — администратор может создать новый, нажав кнопку «Новая БД» и выбрав подходящую дирректорию (Рисунок Ж2):

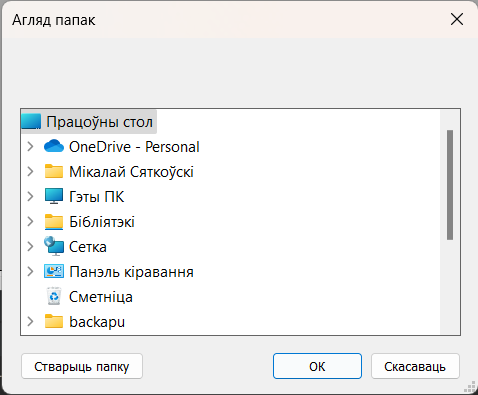


Рисунок Ж2 – Выбор дирректории для БД

После открытия БД, пользователь может переключаться между таблицами и заполнять их (Рисунок Ж3)

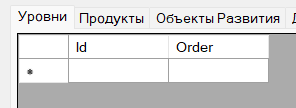


Рисунок Ж3 – Редактирование таблицы

Если файл БД уже имеется – пользователь может нажать на «Открыть БД» и выбрать необходимый файл (Рисунок Ж4):

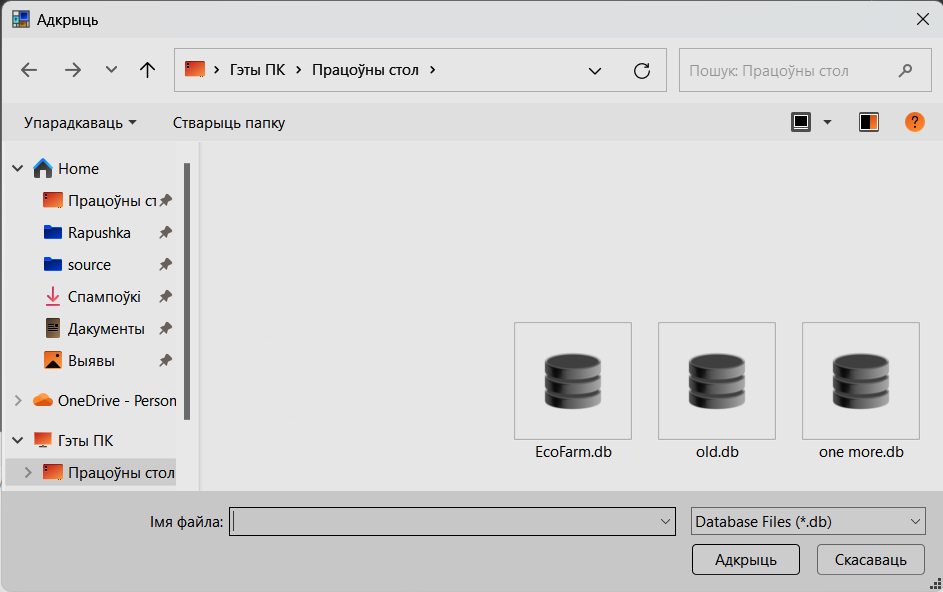


Рисунок Ж4 – Выбор файла БД для открытия

После завершения работы пользователь может сохратнить   
БД (Рисунок Ж5):

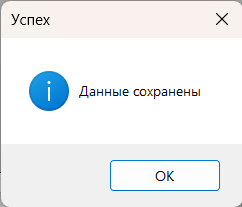


Рисунок Ж5 – Окно подтверждающее сохранение

Для добавления записи в таблицу пользователь может нажать на последнюю пустую строку. Для редактирования – дважды кликнуть по нужной ячейке. А для удаления – необходимо выделить нужные строки и нажать кнопку «Удалить записи» (Рисунок Ж6):

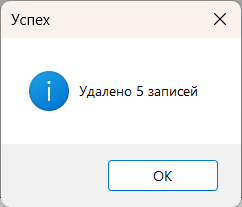


Рисунок Ж6 – Подтверждение удаления

В программе также доступна работа с сериализованными json файлами. Пользователь может нажать кнопку «Десериализовать» и выбрать нужный файл (Рисунок Ж7):

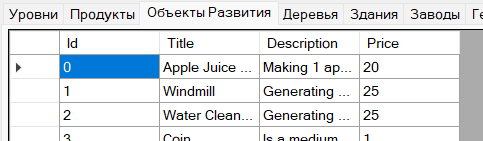


Рисунок Ж7 – Десериализованные данные

Также доступна возможность сериализации, для дальнейшего использования файлов в игровом приложении «Эко-ферма» (Рисунок Ж8):

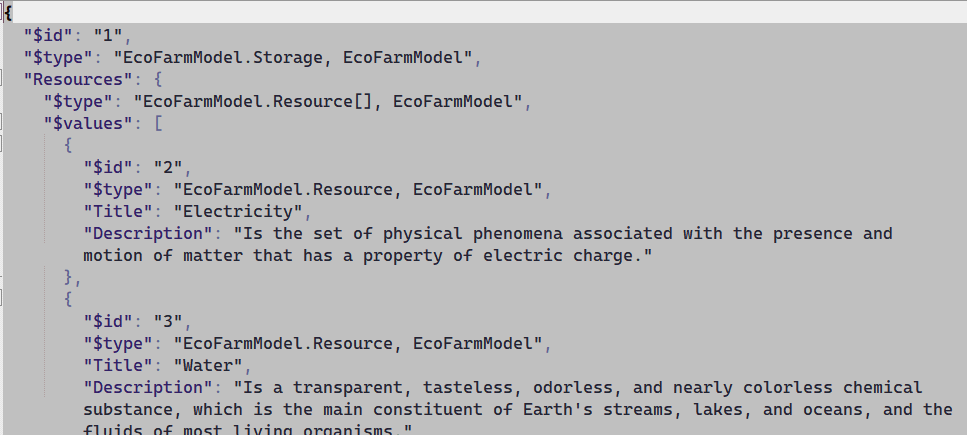


Рисунок Ж8 – Сериализованные данные

# ПРИЛОЖЕНИЕ И (обязательное) Содержание электронного носителя

На прилагаемом электронном носителе информации расположены следующие директории и файлы:

1. Директория «DOC» содержит файл «П41\_Сятковский\_Н\_В\_ТП.docx» и представляет собой документ с пояснительной запиской.
2. Директория «PRG» содержит исходные файлы программы.
3. Директория «EXE» содержит финальную версию исполняемого файла проекта «EcoFarmAdmin.exe».
4. Файл «Readme.txt» содержит описание содержания электронного носителя.