**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 Лайн Девелопмент»

за период с 14 сентября 2020 г. по 9 ноября 2020 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](#_Toc83665719)

[1.1 Функции предприятия, его организационная структура 4](#_Toc83665720)

[1.2 Функции, цели и задачи подразделения (если проходил практику в конкретном подразделении) 4](#_Toc83665721)

[1.3 Должностные обязанности персонала рабочих мест 4](#_Toc83665722)

[1.4 1Компьютерная сеть предприятия 4](#_Toc83665723)

[1.5 Организация взаимодействия с глобальными сетями 4](#_Toc83665724)

[1.6 Эксплуатируемое прикладное ПО 4](#_Toc83665725)

[1.7 Методы и средства разработки ПО (если предприятие разрабатывает ПО) 4](#_Toc83665726)

[1.8 Организация электронного документооборота (если используется на предприятии) 4](#_Toc83665727)

[1.9 Методы защиты информации 4](#_Toc83665728)

[1.10 Организация регистрации и устранения сбоев и ошибок ПО и аппаратуры 4](#_Toc83665729)

[2 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ ТЕХНОЛОГИЧЕСКОЙ ПРАКТИКИ 5](#_Toc83665730)

[2.1 Анализ исходных данных и постановка задачи 5](#_Toc83665731)

[2.2 Описание предметной области 5](#_Toc83665732)

[2.3 Сравнительный анализ аналогов и прототипов 5](#_Toc83665733)

[2.4 Разработка технического задания 5](#_Toc83665734)

[2.5 Проектирование программного средства 5](#_Toc83665735)

[2.6 Проектирование структуры данных 5](#_Toc83665736)

[2.7 Проектирование классов (если используются) 5](#_Toc83665737)

[2.8 Проект интерфейса программного средства 5](#_Toc83665738)

[2.9 Реализация и тестирование программного средства 5](#_Toc83665739)

[2.10 Описание реализации программного средства 5](#_Toc83665740)

[2.11 Тестирование программного средства 5](#_Toc83665741)

[2.12 Установка и эксплуатация ПО 5](#_Toc83665742)

[2.13 Выводы по второму разделу 5](#_Toc83665743)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 6](#_Toc83665744)

[ПРИЛОЖЕНИЕ А (обязательное) \*\*\*\*\*\* 7](#_Toc83665745)

[ПРИЛОЖЕНИЕ Б (обязательное) Содержание электронного носителя 8](#_Toc83665746)

# РЕЗУЛЬТАТЫ ПРОХОЖДЕНИЯ ОБЩЕЙ ОЗНАКОМИТЕЛЬНОЙ ЧАСТИ ПРАКТИКИ

## Функции предприятия, его организационная структура

## Функции, цели и задачи подразделения

*(если проходил практику в конкретном подразделении, то описываем его цели и задачи)*

## Должностные обязанности персонала рабочих мест

*(тут пишем выдержки из должностных инструкций техников-программистов и т.п.)*

## Компьютерная сеть предприятия

*(описываем сеть: как устроена, сколько ПК в сети…)*

## Организация взаимодействия с глобальными сетями

*(как реализован доступ в Интернет. какие есть ограничения и т.п.…)*

## Эксплуатируемое прикладное ПО

*(какой софт есть на ПК пользователей)*

## Методы и средства разработки ПО

*(если предприятие разрабатывает ПО)*

## Организация электронного документооборота

*(если используется на предприятии)*

## Методы защиты информации

*(описываем существующие на предприятии методы защиты информации от несанкционированного доступа и т.д…)*

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

# РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ ТЕХНОЛОГИЧЕСКОЙ ПРАКТИКИ

## Анализ исходных данных и постановка задачи

Сюда индивидуальное задание?

## 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.

## Сравнительный анализ аналогов и прототипов

Для формирования требований к разрабатываемому приложению необходимо изучить аналоги и прототипы.

Аналогами приложения послужили следующие прототипы:

1. Microsoft 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).

Достоинства приложения:

* мультиплатформенность;
* наличие веб-версии;
* включён в стандартный пакет Microsoft Office;
* широкий функционал;
* возможности расчета;
* графические инструменты;
* сводные таблицы;
* язык программирования макросов – VBA;
* синхронизация с облаком.

Недостатки приложения:

* труден в освоении для новых пользователей;
* большинство функций не используются в управелении игровыми данными;
* веб-версия урезана по функционалу;
* высокая цена.

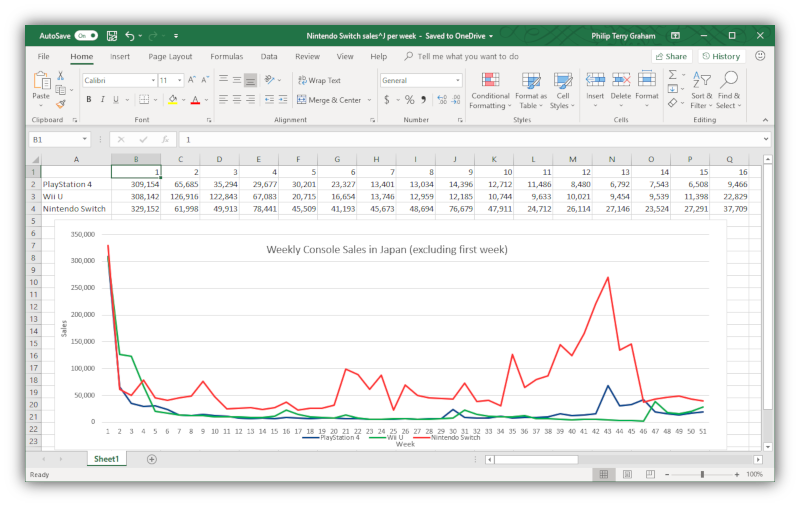


Рисунок 1.1 – Microsoft Excel

1. 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.

Достоинства приложения:

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

Недостатки приложения:

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

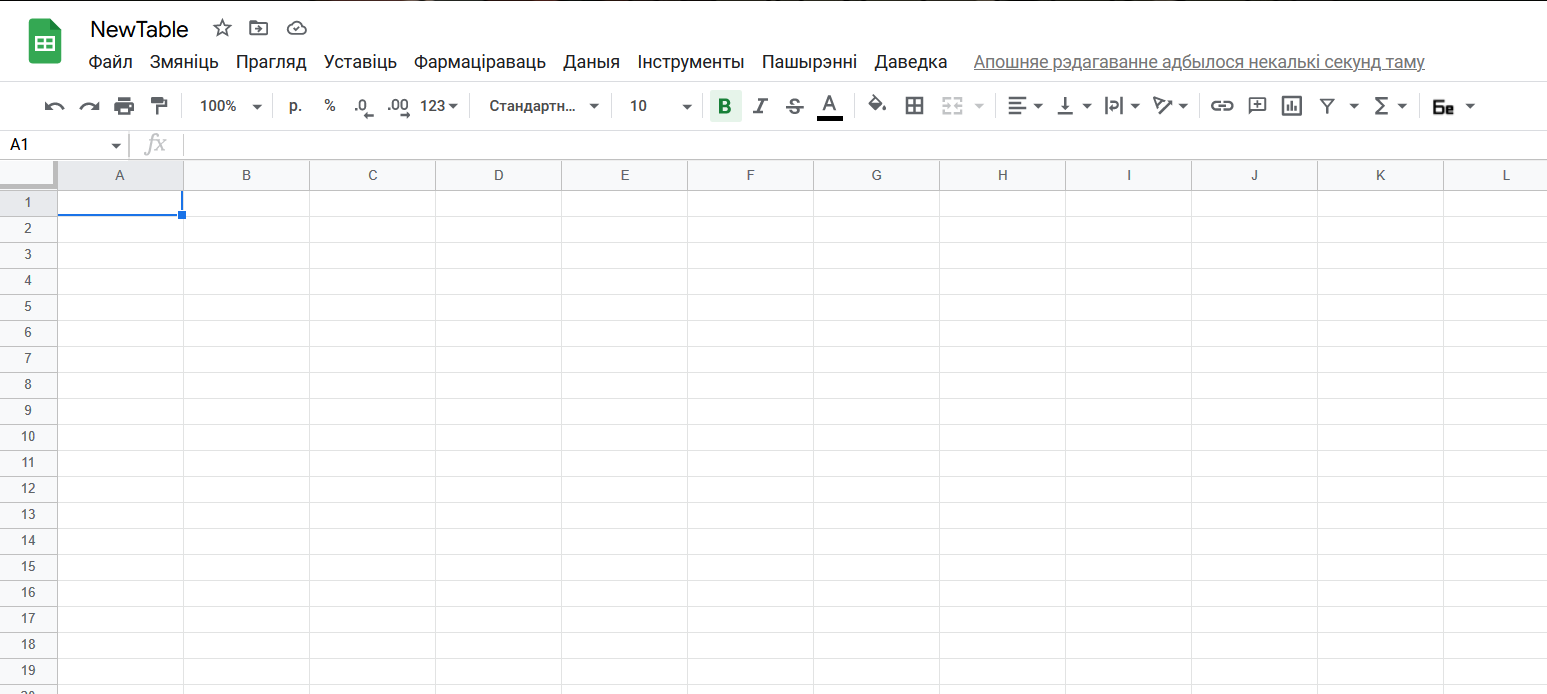


Рисунок 1.2 – Google Sheets

## Разработка технического задания

**2.4.1 Спецификация функций программного средства**

Диаграммы вариантов использования описывают взаимоотношения и зависимости между группами вариантов использования и действующих лиц, участвующими в процессе**. [7]**

**Для данного** компьютерной игры**, в соответствии с диаграммой вариантов использования (ПРИЛОЖЕНИЕ А), необходима реализация следующих функций для игрока:**

Основные требования для пользователей системы:

Для роли Игрок:

* **функции передвижения фигур – передвижение фигур по горизонтали, поворот фигур и ускорение падения;**
* **очистка линий –** линии должны удаляться при заполнении;
* **проигрыш –** завершение игры при переполнении игрового поля.

Функции для системы:

* **базовые функции физики – просчёт взаимодействия фигур на основе их материалов;**
* **комбо –** расчёт очков в зависимости от числа очищенных одновременно линий;

### 2.4.2 Обоснование архитектуры программного средства и способа организации данных

**Среди архитектурных визуализаций существуют 2 основных типа:**

**2D – в 2D – играх используется плоская графика, называемая спрайтами, которая не имеет трехмерной геометрии. Спрайты отображаются на экране как плоские изображения, а камера не имеет перспективы (ортогональная проекция). Из примеров 2D-игр, созданных в Unity, можно вспомнить [Hollow Knight](https://unity.com/madewith/hollow-knight) от Team Cherry, [Children of Morta](https://unity3d.com/2d-game-development-case-stories/children-of-morta). Не все 2D игры одинаковы. Существует два основных стиля: «классические» 2D игры: с видом сверху вниз (шахматы, шашки) или с видом сбоку (Sonic) и «изометрические», которые пытаются имитировать 3D с помощью изометрической проекции под заданным углом. Прежде чем перейти к полноценному 3D, многие жанры двигались от «классического» 2D в сторону «изометрического».**

**3D – в 3D-играх обычно используется трехмерное пространство, где материалы и текстуры отрисовываются на поверхности игровых объектов, формируя целостное окружение, персонажей и объекты игрового мира.**

**Сцену 3D-игры обычно отрисовывают в перспективе, поэтому с приближением объектов к камере они становятся больше. Из примеров 3D-игр, недавно созданных в Unity, можно назвать [Praey for the Gods](https://unity.com/madewith/praey-for-the-gods) от No Matter Studios, [Osiris: New Dawn](https://unity.com/madewith/osiris-new-dawn) от Fenix Fire и [Eastshade](https://www.eastshade.com/) от Eastshade Studios.**

**В** компьютерной игре **«TetraCraft» предусмотрена 3D визуализация. Сделан такой выбор по нескольким причинам:**

**– стиль игры подразумевает 3-х мерное пространство;**

**– простота разработки.**

**Данная игра будет локальной, которая не будет требовать подключение к интернету для игрового процесса. В игре будет участвовать 1 главный игрок.**

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

### 2.4.3 Характеристика инструментов реализации

Выбор игрового движка представляет собой сложную многопараметрическую задачу и является одним из важных этапов при разработке игры. Выбранный программный продукт должен удовлетворять как текущим, так и будущим потребностям при разработке игры, при этом следует учитывать финансовые затраты на приобретение необходимого аппаратного обеспечения, самой системы, разработку необходимого программного обеспечения на ее основе.

Вариаций игровых движков в современном мире очень много, из которых был выбран Unity 3D.

Игровой движок Unity 3D В первую очередь, дает возможность разрабатывать игры, не требуя для этого каких-то особых знаний. Здесь используется компонентно-ориентированный подход, в рамках которого разработчик создает объекты (например, главного героя) и к ним добавляет различные компоненты (например, визуальное отображение персонажа и способы управления им). Благодаря удобному Drag & Drop интерфейсу и функциональному графическому редактору движок позволяет рисовать карты и расставлять объекты в реальном времени и сразу же тестировать получившийся результат.

Второе преимущество движка – наличие огромной библиотеки ассетов и плагинов, с помощью которых можно значительно ускорить процесс разработки игры. Их можно импортировать и экспортировать, добавлять в игру целые заготовки – уровни, врагов, паттерны поведения ИИ и так далее.

Unity доступен бесплатно, что открывает перед независимыми разработчиками дверь в игровую индустрию.

Для графической составляющей игры были использованы следующие программы:

– Paint Tool SAI 2;

это легкий редактор растровой графики и программное обеспечение для рисования для Microsoft Windows, разработанное и опубликованное Systemax Software.

– Aseprite;

это графический редактор с открытым исходным кодом, предназначенный для рисования и анимации pixel art.

– Zenject;

Фреймворк позволяющий реализовать паттерн «Внедрение зависимостей» (Dependency Injection, DI) в Unity.

Внедрение зависимостей – Процесс предоставления внешней зависимости программному компоненту. Является специфичной формой «инверсии управления», когда она применяется к управлению зависимостями.

### 2.4.4 Обоснование эксплуатационных требований

В ходе сравнительного анализа аналогов и прототипов разрабатываемого программного средства были приняты решения использовать операционную систему (ОС) Windows 10 и выше, а также минимальные требования к ней:

* 64-разрядный (x64) процессор с тактовой частотой 1.5ГГц или выше;
* 4ГБ ОЗУ;
* 16 ГБ (для 32-разрядной системы) или 20 ГБ (для 64-разрядной системы) свободного места на жестком диске;
* графическое устройство DirectX 9 с драйвером WDD 1.0 и более поздней версии.

## Проектирование программного средства

## Проектирование структуры данных

## Проектирование классов

*(если используются)*

## Проект интерфейса программного средства

## Реализация и тестирование программного средства

## Описание реализации программного средства

## Тестирование программного средства

## Установка и эксплуатация ПО

## Выводы по второму разделу

# 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.

*первые пункты не правьте. они будут у вас такие же. остальное оформляем по СТП. В тексте отчёта на источники, которые вы тут перечислили обязательно должны быть ссылки. Будем проверять!! на защите*

# ПРИЛОЖЕНИЕ А (обязательное) \*\*\*\*\*\*

*в приложения обязательно отсылаем громоздкие рисунки. таблицы, которые загромождают повествование (например описание физической модели БД, проект экранных форм интерфейса..), а также программную документацию*

# ПРИЛОЖЕНИЕ Б (обязательное) Содержание электронного носителя