Пермский филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский университет

«Высшая школа экономики»

Факультет экономики, менеджмента и бизнес-информатики

Образовательная программа бакалавриата «Программная инженерия»

**ОТЧЕТ**

**по производственной практике**

Выполнил студент группы ПИ-14-1

Щелкунов Андрей Александрович

(Фамилия, Имя, Отчество)

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

(подпись)

Проверили:

Руководитель практики от предприятия:

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

(должность)

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

(Фамилия, Имя, Отчество)

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

(оценка) (подпись)

МП

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

(дата)

Руководитель практики от НИУ ВШЭ – Пермь:

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

(должность, ученая степень, ученое звание)

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

(Фамилия, Имя, Отчество)

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

(оценка) (подпись)

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

(дата)

Аннотация

В работе представлено описание компании ООО «Новая платформа», а также этапов разработки утилиты удаления файлов, выполненной в рамках производственной практики. Приведено описание организационной структуры компании, рассмотрены сферы деятельности и используемые на предприятии методы, технологии и инструментальные средства разработки программных систем. Описаны этапы разработки утилиты, которые включают постановку задачи, проектирование модели тестовой предметной области, реализация алгоритмов удаления и подготовка модульных тестов.

Для создания тестовой предметной области было использовано case-средство Flexberry Designer. В качестве языка программирования был выбран C#. В процессе создания тестовой базы данных была использована СУБД MS SQL Server.

Работа состоит из 5 глав и 1 приложения, занимает 35 страницы и содержит 15 рисунков и 7 таблиц.

Оглавление

[Введение 4](#_Toc485082865)

[Глава 1. Описание компании «Новая платформа» 6](#_Toc485082866)

[1.1. Организационная структура компании 6](#_Toc485082867)

[1.1.1. Отдел маркетинга программных решений 7](#_Toc485082868)

[1.1.2. Ресурсно-технологический отдел 8](#_Toc485082869)

[1.1.3. Центры производства информационных систем и отраслевых технологий 10](#_Toc485082870)

[1.2. Сферы деятельности компании 12](#_Toc485082871)

[1.3. Используемые методы, технологии и инструментальные средства 13](#_Toc485082872)

[Глава 2. Разработка утилиты удаления файлов 14](#_Toc485082873)

[2.1. Постановка задачи и формулировка требований 14](#_Toc485082874)

[2.2. Проектирование 15](#_Toc485082875)

[2.2.1. Проектирование модели тестовой предметной области и базы данных 15](#_Toc485082876)

[2.2.2. Диаграмма классов 17](#_Toc485082877)

[2.3. Реализация 19](#_Toc485082878)

[2.3.1. Реализация алгоритма определения записей для удаления 19](#_Toc485082879)

[2.3.2. Реализация алгоритма удаления файлов 19](#_Toc485082880)

[2.3.3. Логирование действий программы 19](#_Toc485082881)

[2.4. Тестирование 19](#_Toc485082882)

[2.4.1. Подготовка модульных тестов 19](#_Toc485082883)

[2.4.2. Создание архива с файлами 19](#_Toc485082884)

[Заключение 20](#_Toc485082885)

[Список сокращений и условных обозначений 21](#_Toc485082886)

[Список литературы 22](#_Toc485082887)

[Приложение A. Техническое задание 23](#_Toc485082888)

[Приложение B. Организационная структура 34](#_Toc485082889)

Введение

Качественный процесс разработки компьютерных игр требует использования современных инструментальных технологий. Они позволяют не только в разы ускорить процесс разработки, но и значительно упростить его. Одним из таких инструментов является игровой движок **Unity3D**. Это современное средство разработки игр, позволяющее создавать 2D и 3D игры и импортировать их на различные платформы.

В данной работе будут описаны этапы разработки компьютерной игры **Worms**. Это пошаговая (turn-based) игра, которая сочетает в себе элементы жанра стратегии и артиллерии (ballistics calculator). Жанр является довольно востребованным, однако связанные с ним игровые механики не реализованы кроссплатформенными средствами.

**Проблема** работы заключается в реализации игровой модели с помощью современных технологий разработки компьютерных игр.

**Объектом** исследования является разработка игры Worms для компьютерной платформы.

**Предмет** исследования – использование игрового движка Unity3D для разработки игры Worms.

**Цель** работы – разработка игровых механик средствами Unity3D на примере игры Worms. Для достижения этой цели необходимо решить следующие **задачи**:

1. Анализ задачи и разработка требований к системе.
2. Формализация описания решения задачи. Выбор и/или разработка алгоритмов для реализации системы.
3. Проектирование приложения.
4. Реализация программной системы.
5. Тестирование и отладка приложения. Подготовка документации.

Последовательное описание выполнения этих задач задает структуру работы, которая разделяется на соответствующие главы.

На этапе анализа будет изучена информация о предметной области, сформулирован набор требований, описаны бизнес‑процессы, проанализированы современные технологии разработки компьютерных игр, рассмотрены аналоги с выявлением достоинств и недостатков, описан сценарий игры и выявлен набор механик, которые составляют геймплей игры.

На втором этапе работы на основе сценария будет построена математическая модель с выделением геометрической модели мира, сцен, игровых объектов и способами их взаимодействия. Также будет осуществлен выбор и приведено описание алгоритмов реализации, в данном случае игровых механик, которые управляют поведением игры.

На стадии проектирования будет описана архитектура системы с рассмотрением отдельных компонентов, классов и их поведения, построена диаграмма классов UML. Будут спроектированы рассмотренные на этапе анализа механики. Затем будет выполнено проектирование графического интерфейса с учетом решаемых системой задач.

На этапе реализации будет описан процесс перевода формальной модели в программную с точки зрения игровых объектов. С учетом описанных требований в систему будут интегрированы существующие алгоритмы реализации необходимых механик. Будут разработаны сцены и создан набор ресурсов (изображения, анимации, звуки, карты, шрифты, скрипты и т.д.), используемых при реализации прототипа.

Этап тестирования и отладки предназначен для проверки надежности разрабатываемой системы. Будет составлен набор тестов, проверяющих качество программы и разработана программная документация.

# Описание компании «Новая платформа»

* 1. Организационная структура компании

Компания «Новая платформа» состоит из восьми подразделений, которые включают в себя:

1. Совет директоров.
2. Департамент информационных систем.
3. Департамент инфраструктурных решений.
4. Департамент комплексных решений.
5. Коммерческий департамент.
6. Сервисный департамент.
7. Финансовый департамент.
8. Дирекция.

Основной задачей департамента информационных систем (ДИС) является осуществление деятельности по предоставлению услуг, связанных с разработкой решений в области информационных технологий, включая построение ИТ‑инфраструктуры, разработку бизнес-решений и заказную разработку программного обеспечения [1]. Основными компетенциями департамента являются создание и сопровождение бизнес-решений в сфере:

1. Организационного и технического документационного оборота.
2. Проектного управления.
3. Управления бизнес-процессами.
4. Управления коммуникациями и совместной работой.
5. Управления контентом.
6. Корпоративных порталов.
7. Управления взаимоотношения с клиентами.
8. Архивного хранения.

Производство программных информационных систем разделяется на этапы, которые включают:

1. Анализ бизнес-процессов заказчика.
2. Проектирование и разработка заказных и информационных систем (настольных и интернет-приложений).
3. Проектирование, реализация и обслуживание решений в области ИТ-инфраструктуры.
4. Внедрение информационных систем.
5. Оказание технической поддержки внедренных информационных систем.

Департамент информационных систем состоит из пяти отделов:

1. Отдел маркетинга программных решений (ОМПР).
2. Руководство департамента информационных систем (РДИС).
3. Ресурсно-технологический центр (РТЦ).
4. Центр производства информационных систем (ЦПИС).
5. Центр отраслевых технологий (ЦОТ).
   * 1. Отдел маркетинга программных решений

Основной целью ОМПР является организация работ и мероприятий для определения целевых рынков, привлечение, сохранение и наращивание количества потребителей программных решений ДИС [2]. Основными функциями отдела маркетинга в области маркетинговых исследований являются:

1. Поиск и обработка первичной информации для анализа рынка по продуктам (сведения о потенциальных потребителях, конкурентах и их характеристиках).
2. Анализ рынка по продуктам (оценка ёмкости рынка, прогнозирование динамики, сегментирование рынка, оценка насыщенности рынка и его сегментов).

Также отдел маркетинга в области разработки маркетинговой стратегии выполняет следующие функции:

1. Формирование продуктовой линейки ДИС, участие в планировании разработки новых продуктов.
2. Выбор целевых сегментов рынка для продуктов.
3. Определение ключевых конкурентных преимуществ продуктов.
4. Выработка предложений в области ценовой политики.
5. Выбор каналов продвижения и средств коммуникации.
6. Разработка планов маркетинговых мероприятий и бюджетов на их реализацию.
7. Участие в разработке планов продаж продуктов.

В области реализации маркетинговой стратегии ОМПР выполняет функции, которые включают следующие:

1. Организация продвижения продуктов по выбранным каналам: собственные сайты, ресурсы сети Интернет, отраслевые мероприятия, СМИ и т.д.
2. Обеспечение необходимых средств коммуникации для выбранных каналов продвижения: презентации, видеоролики, статьи, баннерная реклама, выставочные стенды, полиграфическая продукция и др.
3. Контроль исполнения планов маркетинговых мероприятий и расходов в рамках утвержденных бюджетов.
4. Анализ эффективности использования каналов продвижения, внесения изменений в планы маркетинговых мероприятий по результатам анализа.
5. Участие в развитии партнерской сети (поиск потенциальных партнеров – системных интеграторов в регионах, осуществление деловых контрактов с потенциальными партнерами, ведение переговоров, заключение партнерских соглашений, курирование работы с партнерами по текущим вопросам).
   * 1. Ресурсно-технологический отдел

Основной целью РТЦ является обеспечение производственных процессов по разработке программного обеспечения кадровыми ресурсами и развитие технологий производства программного обеспечения, подготовка и предоставление ресурсов, развития технологий производства программного обеспечения [3]. Основными задачами РТЦ являются:

1. Организация деятельности по поиску, привлечения и отбору потенциальных сотрудников, проведение собеседований с соискателями, их прием на работу, первоначальное обучение, проведение аттестаций.
2. Предоставление ресурсов программистов-разработчиков и аналитиков подразделениям департамента и внешним организациям.
3. Развитие поддержание и сопровождение технологии разработки программного обеспечения.
4. Создание перспективных технологий разработки программного обеспечения, выполнение НИОКР.
5. Построение и развитие экосистемы на основе технологической платформы.
6. Проведение учебных и презентационных мероприятий, а также формирование методик применения новых технологий разработки программного обеспечения.
7. Обучение сотрудников технологиям разработки программного обеспечения, используемым в ДИС.
8. Обеспечение поддержки и развития технологий разработки программного обеспечения.

Деятельность подразделения РТЦ организуется в нескольких областях, связанных с обеспечением как кадровых ресурсов, так и технологических. В области приема и первоначального обучения сотрудников отдел выполняет следующие функции:

1. Проведение мероприятий для поиска и привлечения новых сотрудников.
2. Отбор потенциальных сотрудников на основании разработанных методик оценки кандидатов.
3. Подготовка новы сотрудников по разработанным программа обучения.
4. Проведение аттестаций сотрудников.

В области предоставления ресурсов программистов-разработчиков и аналитиков выполняются задачи:

1. Предоставление подразделениям департамента ресурсов программистов-разработчиков и аналитиков различной квалификации.
2. Предоставление ресурсов программистов-разработчиков и аналитиков различной квалификации внешним организациям в качестве аутсорсинга.

Также выполняются задачи в области развития, поддержания и сопровождения технологии разработки программного обеспечения, которые включают в себя:

1. Поддержка технологической платформы. Обеспечение беспрерывной работы платформы. Выполнение доработок по требованиям пользователей.
2. Консультационные услуги, в том числе консультации по разработке с использованием технологических продуктов.
3. Расширение функциональных возможностей технологических продуктов.
4. Предоставление доступа к актуальной базе знаний по технологическим продуктам, технологиям их использования, а также по общим информационным технологиям.

Выполняются также задачи в области создания перспективных технологий разработки ПО и обеспечения НИОКР, среди которых:

1. Проведение анализа и исследования перспективных технологий разработки, реализация опытны образцов.
2. Применение новых базовых технологий в технологической платформе.
3. Подача заявок на гранты по тематике технологий разработки программного обеспечения.
4. Выполнение НИОКР, подготовка научно-технических отчетов.

Еще одна область, с которой работает РТЦ, связана с построением и развитием экосистемы на основе технологической платформы. Сюда входят следующие задачи:

1. Проведение анализа рынка технологий разработки ПО.
2. Создание законченных продуктов, составляющих технологическую платформу.
3. Проведение активностей по привлечению участников экосистемы.
4. Вовлечение участников экосистемы в работу над элементами технологической платформы.

И, наконец, отделом выполняется учебно-методическая деятельность, которая подразумевает выполнение следующих обязанностей:

1. Формирование учебных программ.
2. Проведение учебных курсов по разработанным программам для специалистов центра, других подразделений департамента и внешних организаций.
3. Проведение презентационных мероприятий (презентация новых технологий, новых версий используемых продуктов и т.д.).
4. Формирование учебно-справочных материалов, методик применения технологий и технологических продуктов в процессе промышленной разработки ПО.
   * 1. Центры производства информационных систем и отраслевых технологий

Основной задачей ЦПИС является разработка заказного программного обеспечения. ЦПИС обеспечивает полный цикл производства ПО, включая анализ бизнес-процессов заказчика, проектирование, разработку на основе собственной технологии Caseberry, внедрение, обучение пользователей, техническую поддержку разработанных программных продуктов и их модернизацию в случае необходимости [4]. Основной задачей ЦОТ является достижение целей ДИС в целом за счет производства и внедрения современных информационных систем [5]. Основными компетенциями ЦПИС и ЦОТ являются:

1. Производство прикладных программных решений широкого профиля.
2. Производство ГИС-решений – включение в решение интегрированного картографического компонента, представляющего собой полнофункциональную геоинформационную систему.
3. Встраивание в информационные системы поддержки многомерного анализа данных – производство аналитических систем, основанных на использовании технологии OLAP.
4. Встраивание в информационные системы поддержки интеллектуального анализа данных и прогнозирования – включение в решения интегрированного компонента поиска скрытых закономерностей в данных и прогнозирования ситуации на основе математических методов.
5. Производство интернет-решений: разработка Web‑приложений и Интернет‑порталов.
6. Интеграция корпоративных систем с использованием универсального средства – корпоративной сервисной шины.
7. Производство решений на основе интернет-сервисов.
8. Услуги в области информационной безопасности.

Основными задачами ЦПИС и ЦОТ являются:

1. Управление пакетом предоставляемых услуг в сфере компетенций подразделения (проектирование новых, совершенствование предоставляемых).
2. Исследование новых технологий, совершенствование технологии разработки информационных систем.
3. Продвижение и продажа услуг по разработке и внедрению информационных систем.
4. Предоставление услуг по разработке и внедрению информационных систем внутренним и внешним заказчикам.

В области управления пакетом предоставляемых услуг центры выполняют следующие задачи:

1. Обзор и анализ современных информационных технологий и платформ для отраслевых технологий, совершенствование собственной платформы для разработки ИС.
2. Обучение персонала центра применению современных технологий, инструментов, методик, и принципов в разработке и внедрении ИС.
3. Формирование и обновление набора услуг, основанных на применении современных технологий и инструментов в разработке ИС.

В области продвижения услуг по производству ИС выполняются следующие функции:

1. Мониторинг информации о проводимых конкурсах (аукционах, котировках) на оказание услуг в сфере компетенций центра.
2. Взаимодействие с прочими подразделениями и руководством компании по продвижению услуг центра.
3. Взаимодействие с партнерами, самостоятельная реализация маркетинговых мероприятий, направленных на продвижение услуг центра.
4. Формирование потребности в услугах по разработке и внедрению ИС у существующих клиентов.

В области предоставления услуг функциями департаментов являются:

1. Анализ бизнес-процессов заказчика.
2. Проектирование и разработка заказных ИС (настольных и интернет‑приложений).
3. Внедрение ИС.
4. Оказание технической поддержки внедренных ИС.
   1. Сферы деятельности компании

Основным направлением разработки компании являются информационные системы.

Заказчики

Методология разработки

Профессии

Источники финансирвоания

* 1. Используемые методы, технологии и инструментальные средства

Основными источниками финансирования компании являются средства, предоставляемые заказчиками при заключении контрактов.

1. Разработка утилиты удаления файлов

В рамках производственной практики была поставлена задача разработки утилиты удаления файлов, использование которой может помочь решить проблему целостности базы данных. Процесс разработки делится на этапы, которые включают постановку задачи и формулировку требований, проектирование модели тестовой предметной области, диаграммы классов и пользовательского интерфейса, реализацию алгоритмов определения и удаления записей, а также логирования действий программы, и тестирование для подготовки модульных тестов.

* 1. Постановка задачи и формулировка требований

Основным функциональным требованием утилиты является удаление записей о файлах из базы данных и удаление соответствующих файлов из файловой. Известно, что в базе данных есть таблица, предназначенная для хранения записей о файлах и произвольное количество других таблиц, ссылающихся на первую по внешним ключам. Также дано, что эта таблица имеет столбец, который хранит относительный путь к файлу в файловой системе (включая название и формат файла). Задача утилиты заключается в определении записей о файлах, на которые нет ссылок из других таблиц и удалении этих записей из базы данных и соответствующих файлов из файловой системы.

Функциональными требованиями утилиты являются:

1. Указание строки подключения к базе данных, в которой необходимо выполнить удаление.
2. Указание абсолютного пути к каталогу, в котором располагаются файлы.
3. Указание названия таблицы с файлами (значение по умолчанию «Файл»).
4. Указание названия поля первичного ключа (значение по умолчанию «PrimaryKey»).
5. Указание названия поля столбца, содержащего относительный путь к файлу (значение по умолчанию «Url»).
6. Запрос на подтверждение удаления.
7. Осуществление логирования действий программы.
8. Удаление файла формата .pdf (при его наличии), находящегося по заданному пути к основному файлу.

При реализации утилиты предлагается использование следующих инструментальных средств и технологий:

1. Flexberry Designer для моделирования тестовой предметной области.
2. MS SQL Server в качестве СУБД.
3. MS Visual Studio 2012 или выше в качестве среды разработки.
4. GitHub в качестве системы контроля версий.

Помимо функциональных требований к программе также предъявляются требования к качеству:

1. Алгоритмы определения отсутствия ссылок на запись о файле должны быть покрыты unit-тестами.
2. Код должен содержать достаточное для понимания сторонним разработчиком количество комментариев.
3. Все классы и методы должны иметь комментарии в формате [XML Documentation](https://msdn.microsoft.com/en-us/library/b2s063f7.aspx).
4. Все классы, методы, поля, свойства и переменные должны иметь понятные и осмысленные имена.

Утилита должна иметь консольный интерфейс и вызываться через командную строку посредством указания названия программы и набора аргументов, представляющих настройки, от которых зависит поведение программы во время работы.

Также предполагается, что утилита будет выполняться по расписанию, т.е. без участия человека. Это означает, что в программе должен быть заложен сценарий, при котором она будет отрабатывать полностью от начала до конца без каких бы то ни было действий со стороны.

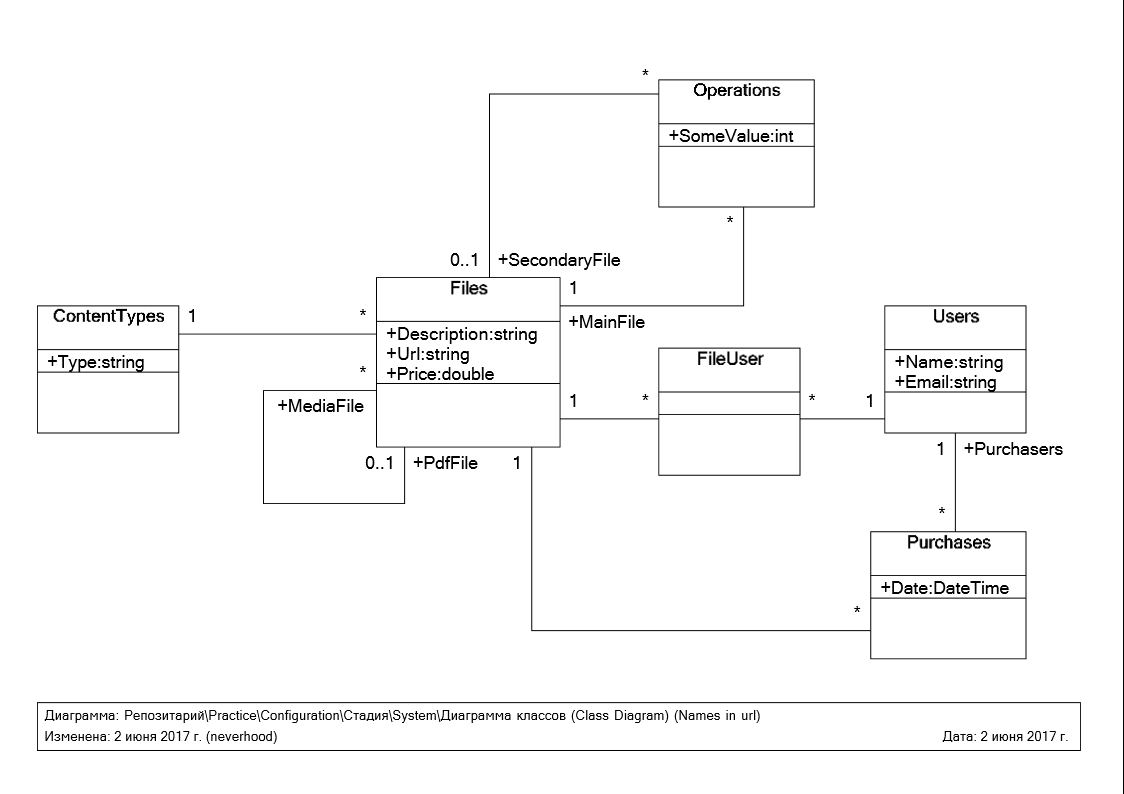
* 1. Проектирование

На этапе проектирования выполняется построение модели тестовой предметной области с учетом требований, выявленных на этапе постановки задачи. Выполняется построение диаграммы классов с учетом функциональных требований к программе.

* + 1. Проектирование модели тестовой предметной области и базы данных

Для осуществления тестового запуска утилиты необходимо построить модель тестовой предметной области, на основе которой создается схема базы данных. С этой базой данных будет взаимодействовать утилита для определения записей, которые необходимо удалить.

В качестве тестовой предметной области был выбран интернет-магазин мультимедиа контента. Магазин предоставляет на выбор покупателю набор файлов, которые представляют один из видов мультимедиа контента (музыка, звуки, изображения, анимации, модели, шрифты и фильмы). Также сервис позволяет пользователям добавлять свои работы для продажи. Таким образом, модель данной предметной области изображена на рис. рисунок 2.1.



**Рисунок 2.1. Модель предметной области «Интернет-магазин мультимедиа контента»**

На диаграмме видно, что таблица, отвечающая за хранение информации о файлах, называется Files и содержит три поля: Description (описание товара) типа string, Url (относительный путь к файлу) типа string, Price (цена товара) типа double. Также имеется таблица пользователей с полями Name (имя пользователя) типа string и Email (электронная почта) типа string. Т.к. пользователи могут добавлять свои файлы в систему необходима связь между таблицами Files и Users. Для этой цели служит таблица FileUser, предназначенная для обеспечения связи «многие-ко-многим».

Интернет‑магазин должен собирать информацию о покупках. Для этого создается таблица Purchases с единственным полем Date (дата покупки) типа DateTime. Она связана с таблицами Files и Users отношением «один-ко-многим», т.к. один и тот же файл может быть куплен несколько раз, и один и тот же пользователь может совершить несколько покупок.

Таблица ContentTypes играет роль типов контента, которые были описаны выше. Она содержит одно поле Type (вид контента) типа string и связана с таблицей файлов отношением «один-ко-многим».

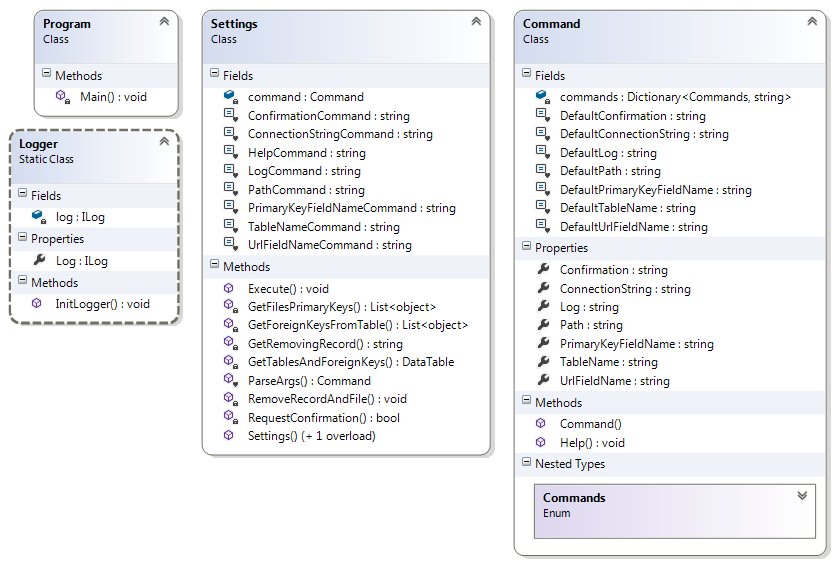
Для создания тестовых сценариев в модель были добавлены связь таблицы Files с самой собой и таблица Operations с двумя связями с таблицей файлов. Отношение таблицы файлов с самой собой предназначения для связи между медиа-файлом и файлом .pdf (который представляет более развернутое описание медиа контента). Т.к. файл с расширением .pdf может не существовать в файловой системе (по условию задачи), то данная связь имеет мощность «1-0..1» (обязательно один медиа-файл и ноль или один .pdf файл).

Таблица Operations добавлена исключительно для создания тестовой ситуации, в которой таблица файлов связана с другой таблицей сразу двумя отношениями. В таком случае, операция может представлять любое действие (подобное покупке) сразу над двумя файлами, причем в ней всегда участвует минимум один файл. Единственное поле SomeValue типа int представляет некоторое значение, характеризующее данную операцию.

Таким образом, в модели присутствуют три тестовые ситуации: ссылки на таблицу файлов из нескольких других таблиц, ссылка на саму себя, таблица с несколькими ссылками на таблицу файлов.

* + 1. Диаграмма классов

Как было сказано на этапе формулировки требований, утилита удаления файлов должна предоставлять возможность настройки работы программы перед запуском. Настройки будут представлять собой набор параметров (в виде строки), которые идут после названия утилиты в командной строке. Каждая настройка представляет собой команду, при задании которой меняется поведение утилиты во время работы. Также необходим класс, отвечающий за выполнение логирования. Таким образом, диаграмма классов имеет вид, представленный на рис. рисунок 2.2.



**Рисунок 2.2. Диаграмма классов**

Класс Program выполняет единственную функцию Main() при запуске программы и передает в нее массив строк, записанный при запуске программы, в качестве ее аргументов.

Класс Settings характеризует настройки программы. Все поля предназначены для хранения названий той или иной команды в сокращенной форме. Метод ParseArgs() распределяет переданный массив строк (из метода Main()) на команды command (в зависимости от конкретного названия в свойствах и переданного сразу после него аргумента). Метод Execute() выполняет команду с учетом настроек. Метод RequestConfirmation() требует подтверждения выполнения удаления (если данная команда установлена). Метод RemoveRecordAndFile() удаляет запись в базе данных (переданной как параметр команды ConnectionStringCommand) и соответствующие файлы. Остальные приватные методы предназначены для поиска записей-кандидатов на удаление.

Класс Command представляет команды утилиты, которые были описаны на этапе постановки задачи в функциональных требованиях (задание строки подключения, задание названия таблицы файлов и т.д.). Каждая команда имеет в качестве аргумента значение по умолчанию, которые представлены полями класса. Свойства предназначены для задания значения аргумента конкретной команды. Перечисление Commands содержит типы команд, которые может выполнить утилита в соответствии с функциональными требованиями. Оно включает Help, ConnectionString, TableName, PrimaryKeyFieldName, UrlFieldName, Path, Log, Confirmation. Метод Help() вызывает полную справку по программе.

Класс Logger предназначен для логирования действий программы. Метод InitLogger() инициализирует логгер. Свойство Log предоставляет доступ к интерфейсу ILog, с помощью которого выполняются методы, записывающие необходимую информацию в файл .log.

* 1. Реализация
     1. Реализация алгоритма определения записей для удаления
     2. Реализация алгоритма удаления файлов
     3. Логирование действий программы
  2. Тестирование
     1. Подготовка модульных тестов
     2. Создание архива с файлами

Заключение

В работе описаны этапы разработки игровых механик на примере игры Worms средствами платформы создания игр Unity.

На первом этапе проведен анализ предметной области и разработаны требования к системе. Проведен обзор и анализ аналога с выявлением достоинств и недостатков. На его основе написан сценарий игры и выявлены основные игровые механики. Также проведен анализ инструментария Unity.

На этапе формализации построена математическая модель игры, описаны алгоритмы работы основных механик, применяемых к системе, с оценкой сложности и ограничениями, указаны, достоинства и недостатки.

На этапе проектирования была описана архитектура системы, спроектирован пользовательский интерфейс, описано поведение игры при переходе из одного состояния в другое.

На этапе реализации программной системы был выбран инструмент разработки, описан процесс разработки с указанием принципа действия реализованных функций, построена диаграмма компонентов.

На последнем этапе разработан набор тестов, проведено тестирование приложения, подготовлена программная документация.

Таким образом были выполнены все задачи, обозначенные в начале работы. Выполнение всех задач подразумевает достижение цели, а именно разработка игровых механик на примере игры Worms средствами Unity. Предложен вариант решения проблемы, связанной с реализацией игровой модели с помощью современных инструментальных средств.

Список сокращений и условных обозначений

В работе использованы следующие сокращения:

1. БД – База данных.
2. ИС – Информационная система.
3. ГИС – Геоинформационная система.
4. OLAP – Online Analytical Processing.
5. НИОКР – Научно-исследовательские и опытно-конструкторские работы.
6. ДИС – Департамент информационных систем.
7. ОМПР – Отдел маркетинга программных решений.
8. РДИС – Руководство департамента информационных систем.
9. РТЦ – Ресурсно-технологический центр.
10. ЦПИС – Центр производства информационных систем.
11. ЦОТ – Центр отраслевых технологий.

Список литературы

x

|  |  |
| --- | --- |
| 1. | ДИС Положение о департаменте, ООО "Новая Платформа", Пермь, Положение 2012. |
| 2. | ДИС ОМПР Положение о подразделении, ООО "Новая платформа", Пермь, Положение 2013. |
| 3. | ДИС РТЦ Положение о подразделении, ООО "Новая платформа", Пермь, Положение 2015. |
| 4. | ДИС ЦПИС Положение о подразделении, ООО "Новая платформа", Пермь, Положение 2010. |
| 5. | ДИС ЦОТ Положение о подразделении, ООО "Новая платформа", Пермь, Положение 2015. |
| 6. | Сферы деятельности РТЦ, ООО "Новая платформа", Пермь, Положение 2014. |
| 7. | РДИС Положение о подразделении, ООО "Новая платформа", Пермь, Положение 2012. |

x

1. Техническое задание

**УТВЕРЖДЕНО**

**разработка утилиты удаления файлов**

**Техническое задание**

2017

Содержание

1. Введение 26

1.1. Наименование программы 26

1.2. Краткая характеристика области применения программы 26

1.3. Краткая характеристика объекта, в котором используют программу 26

2. Основание для разработки 26

2.1. Документы, на основании которых ведется разработка 26

2.2. Организации, утвердившие эти документы и дата их утверждения 26

2.3. Наименование и (или) условное обозначение темы разработки 26

3. Назначение разработки 27

3.1. Функциональное назначение программы 27

3.2. Эксплуатационное назначение разработки 27

4. Требования к программе 27

4.1. Требования к функциональным характеристикам программы 27

4.1.1. Требования к составу выполняемых функций 27

4.1.2. Требования к организации входных и выходных данных 28

4.1.3. Требования к временным характеристикам 28

4.2. Требования к надежности 28

4.2.1. Требования к обеспечению надежного функционирования программы 28

4.2.2. Требования к обеспечению устойчивого функционирования 28

4.3. Условия эксплуатации программы 28

4.3.1. Виды обслуживания программы 28

4.3.2. Необходимое количество и квалификация персонала 28

4.4. Требования к составу и параметрам технических средств 29

4.5. Требования к информационной и программной совместимости 29

4.5.1. Требования к информационным структурам на входе и выходе 29

4.5.2. Требования к методам решения 29

4.5.3. Требования к исходным кодам 29

4.5.4. Требования к языкам программирования 29

4.5.5. Требования к программным средствам, используемым программой 29

4.6. Требования к маркировке и упаковке 30

5. Требования к программной документации 30

5.1. Предварительный состав программной документации 30

5.2. Специальные требования к программной документации 30

6. Технико-экономические показатели 30

6.1. Ориентировочная экономическая эффективность программы 30

6.2. Предполагаемая годовая потребность в программе 30

7. Стадии и этапы разработки 31

7.1. Стадии разработки 31

7.2. Содержание работ по этапам 31

7.3. Сроки и исполнители разработки 32

8. Порядок контроля и приемки 32

8.1. Виды испытаний программы 32

8.2. Общие требования к приемке работы 32

1. Введение
   1. Наименование программы

Наименование программы – утилита удаления файлов.

* 1. Краткая характеристика области применения программы

Программа предназначена для удаления записей из базы данных и удаления соответствующих файлов из файловой системы.

* 1. Краткая характеристика объекта, в котором используют программу

Программа предназначена к применению в профильных подразделениях ООО «Новая платформа» в Перми.

1. Основание для разработки
   1. Документы, на основании которых ведется разработка

Основанием для проведения разработки является Договор №1 от 01.10.2016.

Договор согласован между старшим преподавателем кафедры информационных технологий в бизнесе Лебедевым В.В. (со стороны НИУ ВШЭ), ведущим инженером‑программистом Матвиенко О.А. (со стороны ООО «Новая платформа») и студентом 3-го курса факультета экономики, менеджмента и бизнес-информатики Щелкуновым А.А., 25.12.2016.

Договор утвержден академическим руководителем образовательной программы «Программная инженерия» НИУ ВШЭ Суховым А.О., 01.03.2017.

* 1. **Организации, утвердившие эти документы и дата их** утверждения

НИУ ВШЭ – Пермь, ООО «Новая платформа», 01.03.2017.

* 1. Наименование и (или) условное обозначение темы разработки

Наименование темы разработки – «Разработка утилиты удаления файлов».

1. Назначение разработки

Утилита предназначена для удаления файлов и записей о них в БД, на которые нет ссылок из других таблиц БД. В базе данных MS SQL Server есть таблица Файл. На эту таблицу есть внешние ключи из произвольного набора других таблиц этой же БД. В таблице Файл есть поле, содержащее относительный путь к файлу в файловой системе. Утилита должна осуществлять поиск записей о файлах, на которые нет ссылок из других таблиц, и удалять эти записи из БД и соответствующие файлы из файловой системы.

* 1. Функциональное назначение программы

Функциональным назначением программы является удаление файлов и записей о них из базы данных, на которые нет ссылок.

* 1. Эксплуатационное назначение разработки

Программа должна эксплуатироваться в профильных подразделениях ООО «Новая платформа». Пользователями программы должны стать сотрудники профильных подразделений ООО «Новая платформа».

1. Требования к программе
   1. Требования к функциональным характеристикам программы

Система должна обладать следующими характеристиками:

1. Надежность.
2. Сопровождаемость.
3. Удобство применения.
4. Эффективность.
5. Универсальность.
6. Корректность.
   * 1. Требования к составу выполняемых функций

Программа должна обладать следующими функциональными характеристиками:

1. Удаление записей таблицы файлов из базы данных.
2. Удаление соответствующих файлов из файловой системы.
3. Выполнение логирования результатов удаления.
   * 1. Требования к организации входных и выходных данных

Источниками входных данных являются пользовательские команды, передаваемые посредством устройств ввода (клавиатуры).

Входными данными является строка с названием программы и набором параметров, которая выполняется через командную строку.

Выходными данными является файл .log, предназначенный для хранения результатов работы программы.

* + 1. Требования к временным характеристикам

Время работы программы может увеличиваться пропорционально количеству записей в базе данных, но не должно превышать 5 секунд.

* 1. Требования к надежности
     1. Требования к обеспечению надежного функционирования программы

Алгоритмы определения отсутствия ссылок на запись о файле должны быть покрыты unit-тестами.

* + 1. Требования к обеспечению устойчивого функционирования

В случае возникновения ошибок и исключительных ситуаций программа должна выводить информационное сообщение об ошибке и способ ее исправления.

* 1. Условия эксплуатации программы

Программа должна эксплуатироваться на персональном компьютере под управлением операционной системы Windows 7.

* + 1. Виды обслуживания программы

Программа не нуждается в обслуживании.

* + 1. Необходимое количество и квалификация персонала

Необходимое количество персонала, требуемого для работы с программой, должно составлять 1 штатную единицу – пользователь программы (оператор).

Пользователь программы (оператор) должен обладать практическими навыками работы с командной строкой.

* 1. Требования к составу и параметрам технических средств

В состав технических средств должны входить:

1. IBM-совместимый компьютер.
2. Процессор Pentium 4 или выше с тактовой частотой не ниже 1.5 ГГц.
3. Оперативная память объемом не менее 1 Гб.
4. Свободное место на жестком диске в объеме 50 Мб.
5. Клавиатура.
   1. Требования к информационной и программной совместимости

Программа должна быть совместима с аппаратурой ООО «Новая платформа».

* + 1. Требования к информационным структурам на входе и выходе

Требования к информационным структурам не предъявляются.

* + 1. Требования к методам решения

Модель тестовой предметной области должна быть построена средствами Flexberry Designer.

* + 1. Требования к исходным кодам

Код должен содержать достаточное для понимания сторонним разработчиком количество комментариев. Все классы и методы должны иметь комментарии в формате [XML Documentation](https://msdn.microsoft.com/en-us/library/b2s063f7.aspx). Все классы, методы, поля, свойства и переменные должны иметь понятные и осмысленные имена.

* + 1. Требования к языкам программирования

В качестве среды разработки должна быть использована MS Visual Studio 2012 или выше. Целевым языком программирования должен быть C#.

В качестве системы управления базами данных должен быть использован MS SQL Server. Целевым языком запросов должен быть SQL.

* + 1. Требования к программным средствам, используемым программой

Для реализации логирования должна быть использована библиотека «log4net».

* 1. Требования к маркировке и упаковке

В качестве результатов производственной практики должны быть представлены:

1. Модель тестовой предметной области в Flexberry Designer.
2. Скрипт создания и заполнения БД для тестовой предметной области.
3. Архив с файлами для тестового запуска утилиты.
4. Скомпилированное приложение – утилита удаления файлов, соответствующее функциональным требованиям.
5. Исходный код утилиты в репозитории на GitHub или любом другом сервисе, предоставляющем систему контроля версий.
6. Требования к программной документации
   1. Предварительный состав программной документации

Состав программной документации должен включать:

1. Техническое задание.
2. Руководство пользователя.
3. Руководство программиста.
4. Набор тестов.
5. Отчет.
   1. Специальные требования к программной документации

К отчету по производственной практики должны прилагаться план прохождения практики с подписью руководителя от предприятия и печатью организации и отзыв с подписями руководителя от предприятия и руководителя от НИУ ВШЭ.

1. Технико-экономические показатели
   1. Ориентировочная экономическая эффективность программы

Ориентировочная экономическая эффективность не вычисляется.

* 1. Предполагаемая годовая потребность в программе

Предполагаемая годовая потребность в программе – один раз в неделю для 3‑5 информационных систем.

1. Стадии и этапы разработки
   1. Стадии разработки

Этапы разработки должны включать:

1. Ознакомление с организационной структурой компании ООО "Новая платформа", ее сферами деятельности, материально-технической базой.
2. Анализ используемых на предприятии методов, технологий, инструментальных средств индустриальной разработки программных систем.
3. Создание модели тестовой предметной области в Flexberry Designer.
4. Создание скрипта заполнения базы данных тестовыми данными из предметной области.
5. Реализация алгоритма определения записей для удаления.
6. Реализация алгоритма удаления записей из базы данных и соответствующих файлов из файловой системы.
7. Реализация логирования действий по удалению файлов.
8. Подготовка модульных тестов.
9. Создание архива с файлами для тестового запуска утилиты.
10. Оформление и представление отчета о прохождении практики руководителям.
    1. Содержание работ по этапам

На этапе анализа изучается информация о предметной области, формулируется набор требований, описываются бизнес‑процессы, анализируются современные технологии разработки компьютерных игр, рассматриваются аналоги с выявлением достоинств и недостатков, описывается сценарий игры и выявляется набор механик, которые составляют геймплей игры.

На втором этапе работы на основе сценария строится математическая модель с выделением геометрической модели мира, сцен, игровых объектов и способами их взаимодействия. Также осуществляется выбор и приводится описание алгоритмов реализации, в данном случае игровых механик, которые управляют поведением игры.

На стадии проектирования описывается архитектура системы с рассмотрением отдельных компонентов, классов и их поведения, строится диаграмма классов UML. Проектируются рассмотренные на этапе анализа механики. Затем выполняется проектирование графического интерфейса с учетом решаемых системой задач.

На этапе реализации описывается процесс перевода формальной модели в программную с точки зрения игровых объектов. С учетом описанных требований в систему интегрируются существующие алгоритмы реализации необходимых механик. Разрабатываются сцены и создается набор ресурсов (изображения, анимации, звуки, карты, шрифты, скрипты и т.д.), используемых при реализации прототипа.

Этап тестирования и отладки предназначен для проверки надежности разрабатываемой системы. Составляется набор тестов, проверяющих качество программы и разрабатывается программная документация.

* 1. Сроки и исполнители разработки

Началом срока разработки считается дата начала производственной практики (22.05.2017). Окончанием срока разработки считается дата окончания производственной практики (04.06.2017). Работа защищается во время публичного выступления и демонстрации возможностей программы (21.06.2017). Исполнителем разработки является студент 3-го курса направления «Программная инженерия» НИУ ВШЭ – Пермь, Щелкунов А.А.

1. Порядок контроля и приемки
   1. Виды испытаний программы

Испытания проводятся согласно разработанным методике оценки качества и плану испытаний. Тестирование проводится на основе разработанного набора тестов.

* 1. Общие требования к приемке работы

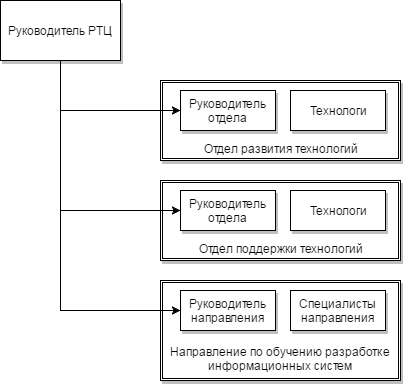
По завершении практики, не позднее пяти календарных дней после окончания  
практики (включая выходные и праздничные дни), руководителю от предприятия предоставляется отчет производственной практики для проверки.

К отчету о прохождении производственной практики прилагается отзыв руководителя практики от предприятия и календарно-тематический план прохождения производственной практики с подписью руководителя практики и печатью организации.

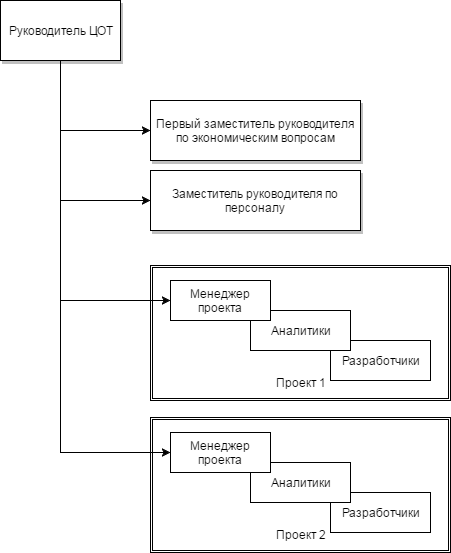
Отзыв руководителя практики от предприятия, на котором проходила практика (с печатью предприятия/организации), должен содержать полное наименование предприятия/организации и подразделения, в котором студент проходил практику, сроки проведения практики, выполненные студентом профессиональные задачи, оценку полноты и качества выполнения программы практики, оценку отношения студента к выполнению заданий, полученных в период практики, оценку сформированности планируемых компетенций (дескрипторов их сформированности), выводы о профессиональной пригодности студента; при необходимости – комментарии о проявленных им личных и профессиональных качествах.

Отчет по итогам практики оформляется в соответствии с Правилами подготовки курсовых работ и ВКР образовательной программы «Программная инженерия» НИУ ВШЭ – Пермь.

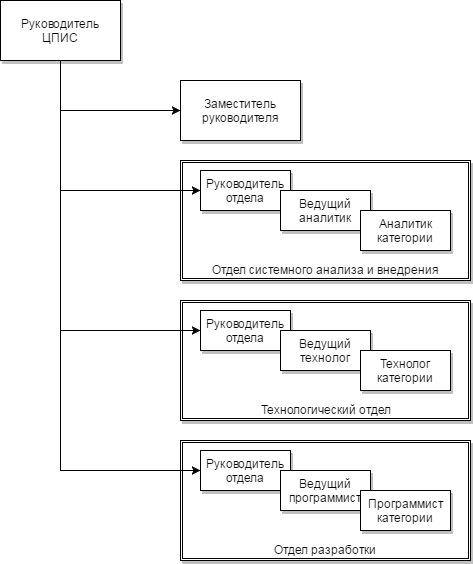
1. Организационная структура



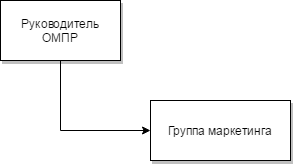
**Рисунок B.3. Организационная структура РТЦ**



**Рисунок B.4. Организационная структура ЦОТ**



**Рисунок B.5. Организационная структура ЦПИС**



**Рисунок B.6. Организационная структура ОМПР**