

*Министерство образования и науки республики Казахстан
Петропавловский колледж железнодорожного транспорта*

***РАЗРАБОТКА СИСТЕМЫ КАТОЛОГИЗАЦИИ АНИМЕ
КОНТЕНТА***

ПКЖТ.1304.002.ДП.ПЗ.008.15

Выполнил:

Снежко Д.В.

_____ ПО-41

“ _____ ” _____

TO DO LIST

- а) Актуальность +
- б) Цель дипломного проекта +/-
- в) Задачи дипломного проекта
- г) Практическая значимость
- д) Объект исследования
- е) Характеристика вычислительной техники

СОДЕРЖАНИЕ

| | |
|---|-----------|
| Введение | 4 |
| 1 Аналитическая часть | 5 |
| 1.1 Анализ предметной области | 5 |
| 1.2 Мессия | 6 |
| 1.3 Структура и функционирование системы (как должно было бы быть) | 7 |
| 1.4 Функции разрабатываемой системы | 7 |
| 1.5 Математическое обоснование проекта | 7 |
| 1.6 Информационное и технологическое обоснование проекта | 7 |
| 1.7 Инструментальное обоснование проекта | 7 |
| 2 Проектная часть | 9 |
| 2.1 Система контроля версий | 9 |
| 2.2 Документирование | 11 |
| 2.3 Модульное тестирование | 12 |
| 2.4 Базы данных | 14 |
| 2.5 Язык программирования | 14 |
| 2.6 Qt | 14 |
| 2.7 Качество кода | 14 |
| Список литературы | 15 |

ВВЕДЕНИЕ

Задачи каталогизации различного контента возникают повсеместно, от школьных библиотек до государственных документов, однако в таких программах заинтересованы не только крупные учреждения, но и рядовые пользователи персональных компьютеров, одна из таких аудиторий - фанаты аниме - жанра японской мультипликации, имеющей очень большую популярность не только в Японии, откуда они родом, но и во всём мире, причём возрастной порог аудитории колеблется от дошкольного до, вполне, зрелого 35-40 лет.

Как правило на счету зрителей аниме, не один десяток просмотренных мультфильмов, в связи с чем - возникают определённые потребности, такие как: ведение списка, которым можно поделиться с друзьями, или записать в него те аниме которые хочешь посмотреть, отмечать на какой серии остановился, а так же некоторые другие и разумеется когда речь идёт не об одном десятке таких записей, возникает потребность иметь навигацию по этому списку, такие базовые вещи, как, поиск, фильтрация, добавление описания и картинки для большего удобства поиска и освежения воспоминаний при поиске, т.к. человеческий мозг наиболее эффективно работает с графической информацией, одного взгляда на картинку может хватить чтобы вспомнить аниме нежели просто прочитав название или вчитываясь в длинное описание.

Целью дипломного проекта является разработка программного продукта позволяющего каталогизировать аниме-контент, а так же связанные с ним, близкие по смыслу, мангу, AMV и Дорамы.

| | | | | | | | | |
|-----------|---------------|----------|-------|------|---|------|--------|----|
| | | | | | ПКЖТ.1304.002.ДП.ПЗ.008.15 | | | |
| Изм | Лист | № докум. | Подп. | Дата | Разработка системы каталогизации аниме контента | | | |
| Разраб. | Снежко Д.В. | | | | | | | |
| Пров. | Черников П.П. | | | | | | | |
| Н. контр. | | | | | | | | |
| Утв. | | | | | | | | |
| | | | | | Лит. | Лист | Листов | |
| | | | | | Д | П | 4 | 16 |
| | | | | | ПО-41 | | | |

1 АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1 Анализ предметной области

1.1.1 Общая характеристика объекта исследования

Каталогизация — это совокупность методов и процессов направленных на хранение, упорядочивание и классификацию какой-либо информации. Термин «каталогизация» употреблялся уже XVI веке. В XIX веке каталогизация становится научной дисциплиной, имеющей несколько самостоятельных направлений.

1.1.2 Организация предметной области

Аниме — японская анимация. В отличие от мультфильмов других стран, предназначенных в основном для просмотра детьми, большая часть выпускаемого аниме рассчитана на подростковую и взрослую аудитории, и во многом за счёт этого имеет высокую популярность в мире.

Аниме отличается характерной манерой отрисовки персонажей и фонов. Издаётся в форме телевизионных сериалов, а также фильмов, распространяемых на видеоносителях или предназначенных для кинопоказа. Сюжеты могут описывать множество персонажей, отличаться разнообразием мест и эпох, жанров и стилей. Источниками для сюжета аниме-сериалов чаще всего являются: манга (японские комиксы), ранобэ (лайт-новел), или компьютерные игры (как правило, в жанре «визуальный роман»).

При экранизации обычно сохраняется графический стиль и другие особенности оригинала. Реже используются другие источники, например, произведения классической литературы. Есть также аниме, имеющие полностью оригинальный сюжет (в этом случае уже само аниме может послужить источником для создания по нему книжных и манга-версий). Значение термина «аниме» может варьироваться в зависимости от контекста. В западных странах аниме является объектом исследования учёных-культурологов, социологов и антропологов — Эри Идзавы, Скотта Маклауда, Сьюзан Напьер, Шерон Кинселлы и других.

| | | | | | | |
|-----|------|----------|-------|------|----------------------------|------|
| | | | | | ПКЖТ.1304.002.ДП.ПЗ.008.15 | Лист |
| Изм | Лист | № докум. | Подп. | Дата | | 5 |

1.2 Мессия

1.2.1 Непроглядная тьма

Рассмотрим процесс составления списка просмотренных аниме на примере обычного, среднестатистического пользователя ПК. Как уже было упомянуто, мысль о ведении списка, приходит на ум не сразу, а как минимум после первого десятка просмотренных аниме. Допустим что наш пользователь посмотрел N -ное количество аниме, где $N > 10$ и ему на ум приходит мысль, что было бы полезным записывать просмотренные аниме, на случай если кто-то спросит совета какое аниме посмотреть, или самому посмотреть/посчитать сколько аниме он посмотрел, его поглощает эта мысль, он открывает блокнот и начинает выписывать всё что вспоминает, посидев минут 20 и вспомнив K количество просмотренных аниме, он так же вспоминает что некоторые он не досмотрел, в виду того что на тот момент они только начинали выпускаться, и новых серий ждать было очень долга, а в последствии некоторые просто забылись и он уже не в состоянии вспомнить какое из них до какой серии он посмотрел. Тут к нему приходит осознание того факта, что блокнот не самое удобное ПО для данной задачи и он думает над тем какую программу будет использовать удобней чем блокнот, он вспоминает про табличные процессоры, такие как MS Excel, LO Calc и другие, конечно он начинает переписывать свои записи из блокнота в выбранную им программу. Спустя какое-то время, год-два, он начинает понимать, что искать по таблице которая уже составляет более сотни аниме не самая простая затея, мало того что данные сильно разбросаны, не все помещаются в одну строчку, некоторые названия просто обрезаются, и в конце концов уже становится сложно что-то найти.

1.2.2 Луч надежды

Тут нашему пользователю на ум, приходит озарение и он вспоминает что в интернете можно не только аниме смотреть но и информацию искать, в том числе другие программы и он начинает искать, спрашивать у знакомых и наконец на-

| | | | | | | |
|-----|------|----------|-------|------|----------------------------|------|
| | | | | | ПКЖТ.1304.002.ДП.ПЗ.008.15 | Лист |
| Изм | Лист | № докум. | Подп. | Дата | | 6 |

ходит нашу программу, вот тут он конечно будет плакать узнав что программа не умеет импортировать из MS Excel и LO Calc ... но не будем о грустном, он начинает переносить свой список в нашу программу и с удовольствием обнаруживает новые возможности, такие как добавление обложки, указание студии, добавления описания и прочего, что он не записывал и ещё более приятной новостью становится когда он узнаёт что эти данные не нужно забивать вручную, а достаточно вписать название и нажать на кнопку поиска, программа сама найдёт аниме или мангу с данным названием и заполнит все поля, в том числе и скачает обложку.

Теперь наш счастливый пользователь сможет добавлять понравившееся ему аниме, мангу, AMV и дораму, удобно искать, сортировать и фильтровать их, помечать те которые он хочет посмотреть и многое другое.

1.3 Структура и функционирование системы (как должно было бы быть)

1.4 Функции разрабатываемой системы

1.5 Математическое обоснование проекта

1.6 Информационное и технологическое обоснование проекта

1.7 Инструментальное обоснование проекта

1.7.1 Эргономика программного обеспечения

(Эргономика - Научная дисциплина, изучающая взаимодействие человека и других элементов системы, а также сфера деятельности по применению теории, принципов, данных и методов этой науки для обеспечения благополучия человека и оптимизации общей производительности системы.)

Программный продукт должен быть разработан с учётом привычек пользователей и накопленного ими опыта, интерфейс программы не должен выбиваться из общего вида системных приложений но в то же время должен быть инновационным и лаконичным.

| | | | | | | |
|-----|------|----------|-------|------|-----------------------------------|------|
| | | | | | <i>ПКЖТ.1304.002.ДП.ПЗ.008.15</i> | Лист |
| Изм | Лист | № докум. | Подп. | Дата | | 7 |

Пользователи должны иметь возможность настраивать отображение элементов управления программы и их размеры в соответствии с их нуждами предпочтениями, помимо этого пользователи должны иметь возможность смены стиля интерфейса.

(В связи с тем что аниме-культура имеет широкое распространение по всему миру)Интерфейс программы должен быть локализуемым на другие языки мира для охвата большей аудитории.

Помимо прочего в различных операционных системах, присутствуют свои устоявшиеся нормы, такие как место на файловой сестеме, где программы хранят свои данные, например, для операционной системы семейства Microsoft Windows таким местом является «C:\Users\<Имя_Пользователя>\AppData\», в то время как для дистрибутивов UNIX-подобных систем, такием местом является «/home/<Имя_Пользователя>/.local/share/». Данное обстоятельство, несомненно, так же относится к эргономичности развёртывания программ под различными операционными системами.

| | | | | | | |
|-----|------|----------|-------|------|----------------------------|------|
| | | | | | ПКЖТ.1304.002.ДП.ПЗ.008.15 | Лист |
| Изм | Лист | № докум. | Подп. | Дата | | 8 |

2 ПРОЕКТНАЯ ЧАСТЬ

2.1 Система контроля версий

Любая информационная система процесс разработки которой не заканчивается на достижении кокого-либо конкретного результата, предполагает постоянную доработку и внесение изменений, иногда такие изменения могут повлечь за собой серьёзные изменения в кодовой базе проекта, выводящие его из функционального состояния на время разработки новых функциональных возможностей.

В связи с тем что, разработка новых функциональных возможностей программного продукта может продолжаться весьма продолжительное время, теряется возможность внесения критически важных обновлений в текущую версию продукта, исходя из этого становится очевидной необходимость каким-либо образом контролировать процесс разработки таким образом чтобы разработка новой версии продукта не вносила осложнений в процесс доработки текущей его версии.

Системы контроля версий предназначены, помимо всего прочего и для решения таких задач, они позволяют контролировать процесс разработки программного продукта, имея возможность создавать контрольные точки, такие как выход новой версии и в последствии возможность возврата к любой из контрольных точек, помимо этого системы контроля версий позволяют создавать различные ветви разработки программы, например, основная ветвь может постоянно отражать рабочее состояние продукта, а отдельная ветвь для критически важных обновлений позволит разрабатывать их не выводя из строя основную версию, причём в последствии эти ветви могут синхронизироваться друг с другом таким образом объединения изменения с различных ветвей.

Так же, системы контроля версий так же позволяют вести разработку программного продукта нескольким разработчикам, значительно упрощая процесс разработки благодаря возможности синхронизации(слияния) ветвей. Таким образом 2 разработчика могут работать над различными частями программного продукта, а по завершении (объединить/синхронизировать/слить) изменения из своих ветвей в главную ветвь получая новую работоспособную версию продукта.

Существуют централизованные и децентрализованные системы контроля версий, основополагающим различием которых является расположение основно-

| | | | | | | |
|-----|------|----------|-------|------|----------------------------|------|
| | | | | | ПКЖТ.1304.002.ДП.ПЗ.008.15 | Лист |
| Изм | Лист | № докум. | Подп. | Дата | | 9 |

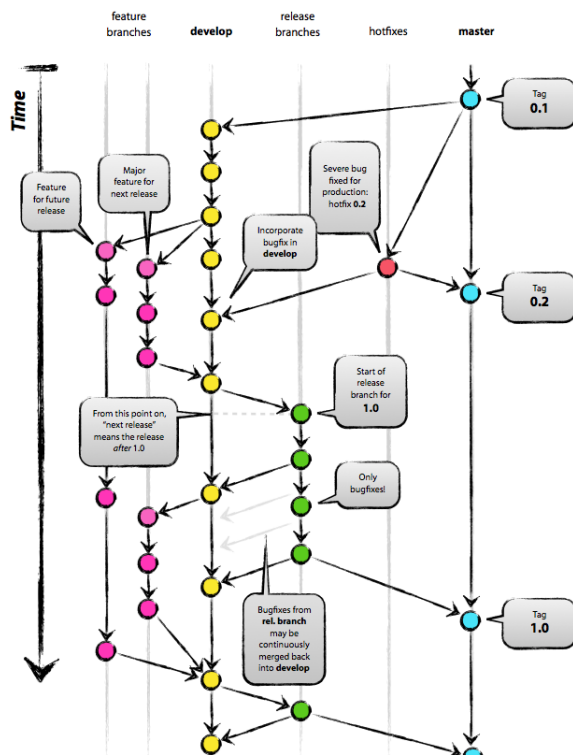


Рисунок 2.1 – Модель ветвления в Git (Оригинальная идея)

го репозитория. В централизованных системах контроля версий существует один единственный репозиторий с которым работают все разработчики и все изменения отправляются туда, минусом такого подхода является зависимость разработчика от доступа к основному репозиторию, а так же проблемы при слиянии(синхронизации) ветвей, в централизованных системах контроля версий этот процесс абсолютно всегда является болезненным, напротив же в децентрализованных системах контроля версий, ярким представителем которых является Git, процесс слияния(синхронизации) различных ветвей является обычным делом, в таких системах каждый разработчик имеет свою копию репозитория, с которой может распоряжаться как угодно и даже если он внесёт какие-либо изменения приносящие вред всему проекту или выходу из строя самой системы контроля версий - эти изменения окажут влияние только на его копию проекта, таким образом разработчик может протестировать внесённые изменения на своём компьютере и убедиться в их работоспособности, помимо того имея на руках копию репозитория он может работать из любого места, будь то самолёт, дача или метро не завися от наличия доступа к интернет.

Таким образом используя в своей работе систему контроля версий я имею возможность, гибко работать с проектом, не боясь потерять какие-либо измене-

ния или вывести из строя проект целиком, в любой момент я имею возможность вернуться к любой из контрольных точек и внести изменения, которые затем с лёгкостью могу перенести/слить/синхронизировать с любой другой ветвью разработки.

2.2 Документирование

Важной частью любого серьёзного проекта является документация, в документации разработчик может и обязан зафиксировать все аспекты работы системы и отдельных её частей. Документация позволяет другим разработчикам узнать необходимую им информацию по интересующему их модулю системы и работе системы в целом, что облегчает другим разработчикам процесс вхождения в разработку проекта.

Многие хорошие проекты разработчики которых забросили их разработку и поддержку, из-за отсутствия документации приходится переписывать с нуля, потому как никто не знает как устроена система и о том как взаимодействуют между собой её отдельные модули, благо что в последнее время о важности документации знают все и новые проекты разрабатываются уже с документацией, но некоторые даже крупные проекты не пишут толковую документацию, а делают это только для вида, например фреймворк FireMonkey, по которому очень трудно найти внятную документацию, но в некоторых проектах, например фреймворк Qt - документация настолько хороша что в ней можно найти ответ на любой возникающий вопрос, на любую мелочь, что является очень весомым плюсом данного фреймворка и сыграло немаловажную роль в моём пристрастии к нему.

2.2.1 Doxygen

Doxygen - это кроссплатформенная система документирования исходных текстов, которая на сегодняшний день, по имеющемуся основанию заявлению разработчиков, стала фактически стандартом для документирования программного

| | | | | | | |
|-----|------|----------|-------|------|----------------------------|------|
| | | | | | ПКЖТ.1304.002.ДП.ПЗ.008.15 | Лист |
| Изм | Лист | № докум. | Подп. | Дата | | 11 |

обеспечения, написанного на языке C++, а также получила пусть и менее широкое распространение и среди ряда других языков.

Doxxygen генерирует документацию на основе набора специальным образом комментированных исходных текстов и также может быть настроен для извлечения структуры программы из недокументированных исходных кодов. Возможно составление графов зависимостей программных объектов, диаграмм классов и исходных кодов с гиперссылками. Результатом работы программы Doxygen является готовая документация для распространения и использования. Doxygen позволяет генерировать на основе исходного кода, содержащего комментарии специального вида, красивую и удобную документацию, содержащую в себе ссылки, диаграммы классов, вызовов и т.п. в различных форматах: HTML, L^AT_EX, CHM, RTF, PostScript, PDF, man-страницы.

Doxxygen используется многими проектами, в том числе KDE, Pidgin, Torque Game Engine, AbiWord, Mozilla, FOX toolkit, Crystal Space, Drupal. Есть встроенная поддержка в KDevelop.

2.3 Модульное тестирование

Любой современный программный продукт представляет собой сложную систему разбитую на некоторые отдельные модули, такое разложение системы на модули заложена в саму суть Объектно Ориентированного Программирования, где каждый класс является отдельным модулем, который так же может зависеть от других модулей которые выполняют более мелкие его функции.

Модульное тестирование, или юнит-тестирование (англ. unit testing) — процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы.

Основная идея модульного тестирования состоит в том, чтобы писать тесты для каждой нетривиальной функции или метода. Это позволяет достаточно быстро проверить, не привело ли очередное изменение кода к регрессии, то есть к появлению ошибок в уже оттестированных местах программы, а также облегчает обнаружение и устранение таких ошибок.

Цель модульного тестирования — изолировать отдельные части программы и показать, что по отдельности эти части работоспособны. Модульное тестирование позже позволяет программистам проводить рефакторинг, будучи уверенными, что модуль по-прежнему работает корректно (регрессионное тестирование). Это поощряет программистов к изменениям кода, поскольку достаточно легко проверить, что код работает и после изменений.

Модульное тестирование помогает устранить сомнения по поводу отдельных модулей и может быть использовано для подхода к тестированию «снизу вверх»: сначала тестируя отдельные части программы, а затем программу в целом.

Модульные тесты можно рассматривать как «живой документ» для тестируемого класса. Клиенты, которые не знают, как использовать данный класс, могут использовать юнит-тест в качестве примера.

Поскольку некоторые классы могут использовать другие классы, тестирование отдельного класса часто распространяется на связанные с ним. Например, класс пользуется базой данных; в ходе написания теста программист обнаруживает, что тесту приходится взаимодействовать с базой. Это ошибка, поскольку тест не должен выходить за границу класса. В результате разработчик абстрагируется от соединения с базой данных и реализует этот интерфейс, используя свой собственный mock-объект. Это приводит к менее связанному коду, минимизируя зависимости в системе.

2.4 Базы данных

2.4.1 SQLite

2.5 Язык программирования

2.5.1 C++

2.6 Qt

2.6.1 Кроссплатформенность

2.6.2 Интернационализация

2.7 Качество кода

2.7.1 Паттерны проектирования

2.7.2 UML

2.7.3 Рефакторинг

2.7.4 Статический анализатор кода CppCheck

СПИСОК ЛИТЕРАТУРЫ

1. Кнут Д.Э. Искусство программирования / перевод с англ. Козаченко Ю.В. в 3-х т. М.: Просвещение 1997.
2. Ульман Дж. Основы систем баз данных / Дж.Ульман, Д.Уидом. М.: Лори, 2000;
3. Шлее М. Qt 4.8 - Профессиональное программирование на C++, «БХВ-Петербург», 2012
4. Саммерфилд М. Qt Профессиональное программирование, разработка кроссплатформенных приложений на C++, «Символ-Плюс», 2011
5. Фаулер М. Рефакторинг - Улучшение существующего кода, «Символ-Плюс», 2003
6. Банда Четырёх Приёмы объектно-ориентированного проектирования, «Питер», 2001
7. Скотт Чакон и Бен Страуб, Pro-Git, «Apress», 2009

Отличия программы от аналогов (Anime List):

- Стабильность, программа тщательно тестируется перед выпуском версий.
- Open Source и свободная лицензия, вы можете свободно делиться программой с друзьями, принять участие в разработке программы и даже написать собственную программу основываясь на нашей, с одним лишь обязательством - она должна быть опубликована под той же лицензией.
- Кроссплатформенность, программа работает на 3х основных платформах - Windows, Linux, MacOS. Хотя для последнего пока ещё нет официальной сборки.
- Эргономичный дизайн и возможность его мргльмодификации при помощи QSS(почти тот же CSS)
- Не требует нарушения политики безопасности (Запуск от имени администратора)
- Интернационализация. Официально мы предоставляем Русскую и английскую локализации, но ничто не мешает вам сделать свою.
- Тех поддержка, мы ответим на все ваши вопросы и постараемся решить любые проблемы в нашей группе.
- Имеется конвертер из экспортного формата программы AnimeList
- Автономность. Для работы программы не требуется постоянное соединение с интернетом, все данные хранятся на вашем локальном компьютере, что так же означает что вы можете легко перенести свою БД на любой другой компьютер.