# Содержание

Список обозначений 3

Введение 4

Глава 1. Разработка формата данных 0

1. 1. Обзор существующих технологий. 0

2. 1. 1. Система TEX 0

2. 1. 1. Язык MathML 0

1. 2. Определение нового формата данных 0

Глава 2. Проектирование приложения 0

2. 1. Архитектура приложения 0

2. 1. 1. Варианты использования 0

2. 1. 2. Классы 0

2. 1. 1. Последовательсть 0

2. 1. 2. Кооперация 0

2. 1. 1. Компоненты 0

2. 1. 2. Развертывание 0

2. 2. ? 0

Глава 3. Внедрение решения 0

3. 1 Конвертер 0

3. 2. Версия поддерживающая шифрование 0

Заключение 0

Список использованных источников 0

# Список обозначений

WWW (World Wide Web) — Всемирная паутина

W3C (World Wide Web Consortium) — Консорциум Всемирной паутины

DOM (Document Object Model) — объектная модель документа

XML (eXtensible Markup Language) — расширяемый язык разметки

MathML (Mathematical Markup Language) — язык математической разметки

TEX — система компьютерной вёрстки

ePub (Electronic Publication) — открытый формат электронных версий книг

ПО — программное обеспечение

Java — объектно-ориентированный, структурный, императивный, кроссплатформенный язык программирования

КПК — Карманный персональный компьютер

# Введение

С появлением технологии отображения информации имитирующей обычную печать на бумаге, неуклонно растет популярность узкоспециализированных компактных планшетных компьютерных устройств (электронных книг).

Несмотря на высокие темпы роста, в настоящее время, все еще сохраняются проблемы с поддержкой технической и научной литературы современными электронными устройствами для чтения. Преобладающим методом распространения научной информации является графическое представление, обладающее рядом негативных факторов: отсутствием возможности для поиска и индексации, сложным созданием документов, зачастую плохим качеством изображений.

Одной из причин слабой поддержки технической и научной литературы является отсутствие универсального механизма, удобного для создания, редактирования и отображения информации такого рода.

Целями данной работы являются:

1. Разработка формата компьютерных данных, способного хранить любые математические выражения и простую текстовую информацию.
2. Проектирование наиболее важных и объемных компонентов приложения для визуализации содержимого нового формата на экранах узкоспециализированных компактных планшетных компьютерных устройств.
3. Создание прототипа части компонентов приложения.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Проанализировать существующие наработки по данной проблеме.
2. Спроектировать приложение для электронных устройств поддерживающее новый формат данных.
3. Исследовать необходимые меры для успешного внедрения и популяризации нового формата данных и приложения.

# Глава 1. Формат данных

## 1. 1. Обзор существующих технологий

### 1. 1. 1. Система TEX

TEX – система компьютерной вёрстки, разработанная американским профессором информатики Дональдом Кнутом в целях создания компьютерной типографии.

Cтиль работы при подготовке текста в системе TEX отличается от стиля работы при работе с редактором Microsoft Word и ближе к программированию, чем к редактированию текста в обычном смысле. Система TEX разделяет более абстрактное представление текста в исходном файле и его типографское расположение на странице, так что в принципе вся работа может совершаться с абстрактным представлением текста и завершаться компиляцией лишь в самом конце. Тем не менее, если мощность компьютера позволяет, рекомендуется часто компилировать текст, чтобы вовремя замечать и устранять неизбежные при его наборе погрешности и ошибки.

Разделение абстрактного представления текста в исходном файле и его типографского исполнения имеет как достоинства, так и недостатки. Несомненным достоинством является то, что основное внимание автора сосредотачивается на содержании текста, а аспекты его форматирования (выбор шрифта, детали расположения текста и набора формул и т.п.) частично передоверяются тщательно разработанным алгоритмам самой системы, а частично могут быть выбраны автором уже после того, как содержание текста создано. (Подобный подход применяется и в редакторе Microsoft Word при последовательном использовании стилей форматирования документа.) Недостатком же системы TEX является необходимость работать со значительно более абстрактным, лишенным наглядности представлением текста, что осложняет первоначальное освоение системы и на этапе подготовки текста нередко приводит к ошибкам, выявляющимся лишь при компиляции исходного файла.

TEX можно использовать для всех видов текста любой сложности. Многие издательства используют его для книгопечатания или книжного набора.

Пример записи формулы на языке TEX для нахождения действительных корней квадратного уравнения:

x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}

### 1. 1. 2. Язык MathML

MathML – язык математической разметки, основанный на технологии XML.

Текущая спецификация MathML 3.0, утвержденная в октябре 2010 года, отвечает современным идеям математической системы представления, предоставляя возможность использовать высокоразвитую систему математической нотации в различных документах.

MathML рассматривает не только представление, но и смысл элементов формулы. Разрабатывается система разметки математической семантики OpenMath, призванная дополнить MathML.

XML-структура MathML обеспечивает широкую область использования и позволяет быстро отображать формулы в WWW, а также легко интерпретировать их значения в математических программных продуктах.

Для соответствия различным требованиям научного сообщества MathML разрабатывается с учетом следующих критериев [http://www.w3.org/Math/]:

1. Предоставление информации, подходящей как для обучения, так и для научной коммуникации любого типа.
2. Возможность преобразования данных с другими математическими форматами, как презентационными, так и семантическими.
3. Поддержка корректного просмотра длинных выражений.
4. Обеспечение расширяемости.
5. Поддержка шаблонов и других средств редактирования математической информации.

Пример записи формулы на языке разметки MathML для нахождения действительных корней квадратного уравнения:

<?xml version="1.0" encoding="UTF-8"?>

<math xmlns="http://www.w3.org/1998/Math/MathML">

<mrow>

<mi>x</mi>

<mo>=</mo>

<mfrac>

<mrow>

<mrow>

<mo>-</mo>

<mi>b</mi>

<mo>±</mo>

</mrow>

<msqrt>

<mrow>

<msup>

<mi>b</mi>

<mn>2</mn>

</msup>

<mo>-</mo>

<mrow>

<mn>4</mn>

<mi>a</mi>

<mi>c</mi>

</mrow>

</mrow>

</msqrt>

</mrow>

<mrow>

<mn>2</mn>

<mi>a</mi>

</mrow>

</mfrac>

</mrow>

</math>

1. 2. Определение нового формата данных

Решением поставленной задачи по разработке необходимого формата данных может быть слияние формата ePub с языком математической разметки MathML.

ePub — открытый формат электронных версий книг, разработанный Международным форумом по цифровым публикациям.

Формат данных ePub позволяет производить и распространять цифровую публикацию в одном файле, обеспечивая совместимость между программным и аппаратным обеспечением, необходимым для воспроизведения цифровых книг и других публикаций с плавающей вёрсткой.

Новый формат данных будет представлять собой контейнер включающий в себя:

1) XML-файл, содержащий математические выражения описанные на языке MathML.

2) XHTML-файл с текстовой информацией.

3) XML-файл с описанием данных о документе, автора и условий распространения документа.

4) Папка с векторной и растровой графикой.

5) Папка со шрифтами, что позволит дополнять документ необходимыми дополнительными шрифтами.

6) Файл таблицы стилей.

# Глава 2. Проектирование приложения

## 1. Унифицированный язык моделирования UML

Язык моделирования – это графическая нотация, которая используется методом для описания проекта. Нотация представляет собой совокупность графических объектов, которые используются в моделях. Нотация является синтаксисом языка моделирования.

UML – современный язык моделирования, принятый на вооружение многими крупными компаниями – производителями ПО.

Главными в разработке UML были следующие цели [Вендров]:

* предоставить пользователям готовый к использованию выразительный язык визуального моделирования, позволяющий разрабатывать осмысленные модели и обмениваться ими;
* предусмотреть механизмы расширяемости и специализации для расширения базовых концепций;
* обеспечить независимость от конкретных языков программирования и процессов разработки;
* обеспечить формальную основу для понимания этого языка моделирования (язык должен быть одновременно точным и доступным для понимания, без лишнего формализма);
* стимулировать рост рынка объектно-ориентированных инструментальных средств;
* интегрировать лучший практический опыт.

Создатели UML представляют его как язык определения, представления, проектирования и документирования программных систем.

При проектировании приложения используем следующий набор диаграмм для моделирования:

* Диаграмма вариантов использования (use case diagram)
* Диаграмма классов (class diagram)
* Диаграмма состояний (statechart diagram)
* Диаграмма последовательности (sequence diagram)
* Диаграмма кооперации (collaboration diagram)
* Диаграмма компонентов (component diagram)
* Диаграмма развертывания (deployment diagram)

## 2. 2. Проектирование структуры программы

### 2. 2. 1. Диаграмма вариантов использования

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

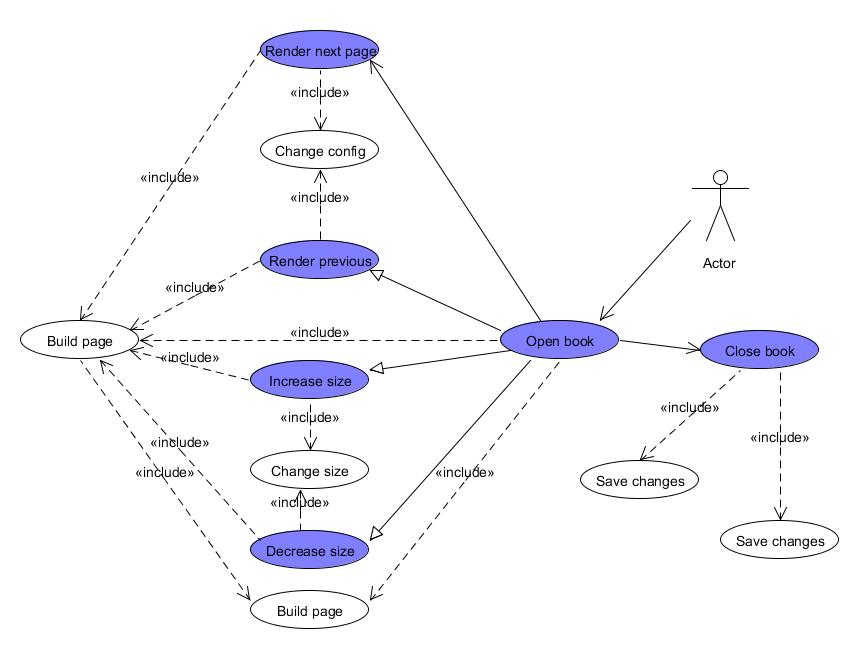
Действующим лицом определим читателя электронной книги, а вариантами использования будем считать действия, которые может совершать данное лицо.

Так как в общем случае действующее лицо всегда находится вне системы, его внутренняя структура никак не определяется. Для действующего лица имеет значение только то, как оно воспринимается со стороны системы.

Цель варианта использования заключается в том, чтобы определить законченный аспект или фрагмент поведения сущности без раскрытия её внутренней структуры.

Для успешного построения диаграммы вариантов использования необходимо также определить отношения элементов диаграммы.

// ассоциации, обобщения, расширения, влючения



### 3. 2. 2. Диаграмма классов

Диаграмма классов является центральным звеном объектно-ориентированных методов. Диаграмма классов определяет типы объектов системы и различного рода статические связи, которые существуют между ними.

На диаграммах классов изображаются атрибуты классов, операции классов и ограничения, которые накладываются на связи между объектами.

Построение диаграммы классов обычно рассматривают в разных аспектах. При построении диаграммы зачастую выбирают единственный аспект и не советуют пренебрегать их смешением. При чтении диаграммы следует выяснить, в соответствии с каким аспектом она строилась. Знание аспекта помогает правильно интерпретировать диаграмму.

Для построения диаграммы классов будем использовать аспект (концептуальный, спецификации, реализации) - ?

### 3. 2. 3. Диаграмма состояний

Диаграммы состояний чаще всего используются для описания поведения отдельных объектов, но также могут быть применены для спецификации функциональности других компонентов моделей, таких как варианты использования, актеры, подсистемы, операции и методы.

Диаграмма состояний является графом специального вида, который представляет некоторый автомат. Вершинами графа являются возможные состояния автомата, изображаемые соответствующими графическими символами, а дуги обозначают его переходы из состояния в состояние. Диаграммы состояний могут быть вложены друг в друга для более детального представления отдельных элементов модели.

### 3. 2. 4. Диаграмма последовательности

На диаграмме последовательности изображаются только те объекты, которые непосредственно участвуют во взаимодействии. Ключевым моментом для диаграмм последовательности является динамика взаимодействия объектов во времени.

Графически каждый объект изображается прямоугольником и располагается в верхней части своей линии жизни. Внутри прямоугольника записываются имя объекта и имя класса разделенные двоеточием. При этом вся запись подчеркивается, что является признаком объекта.

Линия жизни объекта изображается пунктирной вертикальной линией, ассоциированной с единственным объектом на диаграмме последовательности. Линия жизни служит для обозначения периода времени, в течение которого объект существует в системе и, следовательно, может потенциально участвовать во всех ее взаимодействиях.

### 3. 2. 5. Диаграмма кооперации

Подобно диаграммам последовательности, диаграммы кооперации отображают поток событий в конкретном сценарии варианта использования. Главная особенность диаграммы кооперации заключается в возможности графически представить не только последовательность взаимодействия, но и все структурные отношения между объектами, участвующими в этом взаимодействии.

Прежде всего, на диаграмме кооперации в виде прямоугольников изображаются участвующие во взаимодействии объекты, содержащие имя объекта, его класс и, возможно, значения атрибутов. Далее, как и на диаграмме классов, указываются ассоциации между объектами в виде различных соединительных линий. При этом можно явно указать имена ассоциации и ролей, которые играют объекты в данной ассоциации. Дополнительно могут быть изображены динамические связи – потоки сообщений. Они представляются также в виде соединительных линий между объектами, над которыми располагается стрелка с указанием направления, имени сообщения и порядкового номера в общей последовательности инициализации сообщений.

В отличие от диаграммы последовательности, на диаграмме кооперации изображаются только отношения между объектами, играющими определенные роли во взаимодействии, а последовательность взаимодействий и параллельных потоков определяется с помощью порядковых номеров.

### 3. 2. 6. Диаграмма компонентов

Диаграммы компонентов показывают, как выглядит модель системы на физическом уровне. На диаграмме изображаются компоненты программного обеспечения и связи между ними.

Пунктирные стрелки, соединяющие модули, показывают отношения взаимозависимости, аналогичные тем, которые имеют место при компиляции исходных текстов программ. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними.

В первую очередь, необходимо решить, из каких физических частей или файлов будет состоять система. На этом этапе следует обратить внимание на такую реализацию системы, которая обеспечивала бы возможность повторного использования кода за счет рациональной декомпозиции компонентов, а также создание объектов только при их необходимости.

### 3. 2. 7. Диаграмма развертывания - ?

Диаграмма размещения / развертывания отражает физические взаимосвязи между программными и аппаратными компонентами системы. Она является хорошим средством для того, чтобы показать маршруты перемещения объектов и компонентов в распределенной системе.

Каждый узел на диаграмме размещения представляет собой некоторый тип вычислительного устройства, в большинстве случаев — часть аппаратуры. Эта аппаратура может быть простым устройством или датчиком, а может быть и мэйнфреймом.

Компоненты на диаграмме размещения представляют собой физические модули программного кода. Как правило, они в точности соответствуют компонентам на диаграмме компонентов. Таким образом, диаграмма размещения отражает выполнение каждого компонента в системе.

Компонент может иметь более одного интерфейса, в этом случае видно, какие компоненты взаимодействуют с каждым интерфейсом.

С другой стороны, каждая система имеет свои собственные физические характеристики, которые желательно явно выделить, и в дальнейшем потребуется большая степень формализма по мере достижения лучшего понимания того, какие проблемы следует решать в первую очередь с помощью диаграмм размещения.

# Глава 3. Внедрение решения

## 2. 1. Конвертер

Наиболее важным условием в процессе распространения нового формата данных среди читателей электронных книг является наличие удобного инструмента конвертирования данных.

# Заключение

В ходе исследования текущей темы проанализированы основы технологий предназначенных для описания математических выражений, некоторые из популярных форматов электронных версий книг.

Качественная визуализация математических выражений сложна по причине их сложной и высокоразвитой двумерной символьной системы обозначений.

Рассмотрен и построен набор некоторых важных диаграмм моделирования, используемых при проектировании современного программного обеспечения.

Построение диаграмм моделирования позволяет существенно облегчить разработку программного обеспечения, обеспечить возможность расширения системы в будущем и независимость от конкретных языков программирования, а также интегрировать в систему лучший практический опыт.

# Список использованных источников

1. Википедия – свободная энциклопедия [Электронный ресурс]. - http://ru.wikipedia.org/wiki/MathML . - (дата обращения: 13.05.2013).

2. Википедия – свободная энциклопедия [Электронный ресурс]. - http://ru.wikipedia.org/wiki/TeX . - (дата обращения: 14.05.2013).

3. Википедия – свободная энциклопедия [Электронный ресурс]. - http://ru.wikipedia.org/wiki/Document\_Object\_Model . - (дата обращения: 10.05.2013).

4. Википедия – свободная энциклопедия [Электронный ресурс]. - http://ru.wikipedia.org/wiki/XML . - (дата обращения: 15.05.2013).

5. W3C – Консорциум Всемирной паутины [Электронный ресурс]. - http://www.w3.org/Math/. - (дата обращения: 15.05.2013).

6. Вендров, А. М. Проектирование программного обеспечения экономических информационных систем / А. М. Вендров.– Москва: Финансы и статистика, 2002. – 349 с.