«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»

(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)

Факультет И (шифр) Информационные и управляющие системы (наименование)

Кафедра И9 (шифр) Систем управления и компьютерных технологий (наименование)

Дисциплина Технология программирования

КУРСОВАЯ РАБОТА

на тему

Программное приложение структурного анализа документов по заданному шаблону

Выполнил студент группы И9М33

Магомедов И.Н.

Фамилия И.О.

РУКОВОДИТЕЛЬ

Арсеньев Б.П. _____

Фамилия И.О. Подпись

Оценка _____

«_____» _____ 20____ г.

САНКТ-ПЕТЕРБУРГ

2017 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Ошибки, возникающие при написании документов, возникают очень часто, что иногда создаёт сложности при попытке правильно понять смысл, заключённый в документе. Чтении документов с ошибками может приводить к недопонимаю сторон и лишней трате времени.

Проверка документов на ошибки при сравнении с шаблоном очень рутинная задача для человека, что может приводить к пропуску ошибок. Она усугубляется в случаях, когда документы начинают разрастаться или они в первой своей итерации имели очень большой объём.

Для решения данной проблемы была поставлена задача создания приложения для сравнения документа с шаблоном.

Целью курсовой работы является проектирование программного приложения для сравнения входного документа с шаблоном.

1 ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Перед этапом проектирования следует составить техническое задание, которое описывает какие возможности должно быть реализовано в приложение и какие задачи оно призвано решать. Также описаны средствам с помощью, который должно быть реализовано приложение.

Наименования проводимых работ и результат работ приведены в Таблице 1.

Таблица 1 – Проводимые по техническому заданию работы.

№ п/п	Название проводимых работ	Результат проводимых работ
1	Разработка алгоритма сравнения документа с шаблоном	Алгоритм сравнения документа с шаблоном
2	Разработка способа выделения изменённого текста, оставшегося без изменений и удалённого	Способ выделения изменённого текста, оставшегося без изменений и удалённого
3	Разработка способа создания новых шаблонов	Способ создания новых шаблонов
4	Разработка способ быстрого поиска по шаблону и документу	Способ быстрого поиска по шаблону и документу
5	Рассмотрение двух текстовых формата документов docx и odt	Рассмотрены два текстовых формата документов docx и odt

К программному продукту предъявляются следующие вспомогательные требования:

- хранить найденные блоки в шаблоне и документе, до завершения работы с документом и шаблоном;
- указатель на перемещённый блок;
- иметь удобный интерфейс создания, редактирования и просмотра документа и шаблона;

К программному продукту предъявляются следующие нефункциональные требования:

- разработка должна производиться в Visual Studio, с использованием VSTO и языка программирования C#;
- наличие интуитивно понятного интерфейса с возможностью навигации;
- наличие кроссплатформенной совместимости.

2 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Анализ документов – это метод сбора первичных данных, при котором документы используются в качестве главного источника информации; это также совокупность методических приёмов и процедур, применяемых для извлечения информации из документальных источников при изучении процессов и явлений в целях решения определённых задач [1].

Методы анализа документов многообразны. Они постоянно совершенствуются. Так, методы анализа логической структуры текста позволяют при чтении черпать из текста документа больше, чем её содержится в самом документе, а при составлении на его основе сжатого текста – делать его логически стройным, легко понимаемым и убедительным.

Как правило, для этого применяется метод шаблонов: входной текст разделяется с помощью шаблонов, затем производится обработка полученных данных.

Метод анализа документов – метод сбора данных в ходе проведения исследований систем управления, основанный на применении информации, зафиксированной в письменной или печатной форме, на магнитной пленке, в электронном виде, в иконографической форме и др [2].

Анализ документов — это совокупность методических прие-мов и процедур, применяемых для извлечения из документальных источников социологической информации при изучении социальных процессов и явлений в целях решения определенных исследователь-ских задач [3].

Во всем многообразии исследовательских приемов, используемых при изучении документов, выделяются два их основных вида – качественный анализ (иногда его называют традиционным) и формализованный, носящий название контент анализа [4]. Эти два во многом различных подхода к изучению документальной информации тем не менее могут дополнять друг друга.

Качественный анализ зачастую служит предпосылкой последующего формализованного изучения документов. Как самостоятельный метод он

приобретает особое значение при изучении уникальных документов: их число всегда невелико, поэтому нет надобности в количественной обработке информации. На первый план в таких случаях выдвигаются углубленное логическое исследование содержания документа, обнаружение возможных «умолчаний», оценка своеобразия авторского языка и стиля изложения.

Стремление в максимальной степени избежать субъективизма, потребность в социологическом изучении и обобщении большого объема информации, ориентация на использование современной компьютерной техники при обработке содержания текстов привели к становлению метода формализованного, качественно–количественного изучения документов – контент–анализа.

Согласно этому методу, содержание текста определяется как совокупность имеющихся в нем сведений, оценок, объединенных в некую целостность единой концепцией, замыслом. Формализованный анализ документов имеет дело с текстом, но ориентирован прежде всего на изучение стоящей за ним реальности. Особо подчеркнем, что внетекстовой реальностью являются не только события, факты, человеческие отношения, отраженные в текстах, но и используемые при их подготовке принципы отбора материалов. Другими словами, для исследователя может быть в равной степени важно и то, что вошло в содержание текста, и то, что оказалось вне его рамок [5].

Процедура формализованного изучения документов начинается с выделения двух единиц анализа – смысловых (качественных) и единиц счета. В тексте она выражается по–разному – словом, сочетанием слов, описанием. Цель исследования – отыскать индикаторы, указывающие на наличие в документе темы, значимой для анализа, и раскрывающие содержание текстовой информации. Например, при изучении роли газеты в распространении технических знаний к публикациям на эту тему могут быть отнесены статьи, очерки, заметки, фотографии, где прямо или косвенно, с различной степенью достоверности говорится о новых достижениях в области техники и технологии.

3 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

Проектирование программного продукта производилось на основе выдвинутых в процессе анализа предметной области требований.

При проектировании были рассмотрены структуры текстовых документов docx и odt. Оба документа представляют из себя zip архивы и разделены на файл форматом xml [6]. Разметка документов схожа, имеются подобные разметки, отличающиеся только именем. Главная особенность разметки xml то, что текст делится на параграфы, представляющие из себя в тексте абзацы. Также таблицы, нумерованные списки и изображения выделяются в отдельную разметку. На рисунках 1, 2 и 3 представленные примеры разметок docx и odt, соответственно.

Рисунок 1 - Пример нумерованного списка в Word

```
- <w:p w:rsidP="00555D53" w:rsidRDefault="00555D53"
  w:rsidR="00555D53">
    - <w:pPr>
        <w:pStyle w:val="a5"/>
      - <w:numPr>
          <w:ilvl w:val="0"/>
          <w:numId w:val="1"/>
        </w:numPr>
      - <w:rPr>
          <w:lang w:val="en-US"/>
        </w:rPr>
      </w:pPr>
    - <w:r>
      - <w:rPr>
          <w:lang w:val="en-US"/>
        </w:rPr>
        <w:t>Text_num_1</w:t>
      </w:r>
  </w:p>
- <w:p w:rsidP="00555D53" w:rsidRDefault="00555D53"
  w:rsidR="00555D53">
```

Рисунок 2 - Пример разметки параграфов и таблицы в Word

```
+ <w:p w:rsidP="00555D53" w:rsidRDefault="00555D53"
  w:rsidR="00555D53">
+ <w:p w:rsidP="00555D53" w:rsidRDefault="00555D53"
  w:rsidR="00555D53">
+ <w:p w:rsidP="00555D53" w:rsidRDefault="00555D53"
  w:rsidR="00555D53">
+ <w:p w:rsidP="00555D53" w:rsidRDefault="00555D53"
  w:rsidR="00555D53">
+ <w:tbl>
+ <w:p w:rsidP="00555D53" w:rsidRDefault="00555D53"
  w:rsidR="00555D53">
```

Рисунок 3 - Пример разметки параграфов, текста, нумерованного спика и таблицы в odt

```
- <office:body>
  - <office:text text:use-soft-page-breaks="true">
      <text:p text:style-name="P1">Text_1</text:p>
    + <text:h text:style-name="Заголовок1" text:outline-level="1">
      <text:h text:style-name="P3" text:outline-level="2">Text_3</text:h>
      <text:p text:style-name="P4"/>
    - <text:list text:style-name="LFO1" text:continue-numbering="true">
      - <text:list-item>
          <text:p text:style-name="P5">Text_num_1</text:p>
        </text:list-item>
      - <text:list-item>
          <text:p text:style-name="P6">Text_num_2</text:p>
        </text:list-item>
      - <text:list-item>
          <text:p text:style-name="P7">Text_num_3</text:p>
        - <text:list text:continue-numbering="true">
          - <text:list-item>
              <text:p text:style-name="P8">Text_num_3_1</text:p>
            </text:list-item>
          - <text:list-item>
              <text:p text:style-name="P9">Text_num_3_2</text:p>
            </text:list-item>
          </text:list>
        </text:list-item>
      </text:list>
      <text:p text:style-name="P10"/>
    + <table:table table:style-name="Table11">
      <text:p text:style-name="P36"/>
    + <text:p text:style-name="Обычный">
      <text:p text:style-name="Обычный"/>
  </office:text>
</office:body>
```

Как видно элементы текстового документа, разделенные на отдельные блоки как в формате docx и odt. Такое разделение можно использовать, для разделение текстового документа на отдельные блоки (элементы), что упростит задачу анализа документа.

Далее были рассмотрены возможные алгоритмы, для модуля сравнения документов и сам процесс работы программы.

Первый вариант.

1. Программе подаётся шаблон на вход;
2. Шаблон анализируется, то есть выделяются основные части;
3. Программе подаётся файл для сравнения;
4. Программа выделяет отдельные блоки документа;

5. Программа поочерёдно заносит в буфер один из блоков исходного документа и сравнивает его с элементами шаблона;
6. Сравнение продолжается пока не будет найден элемент похожий на блок из исходного документа или не наступит конец шаблона;
7. Оценка схожести:
    a. если блок найден, то он выделяется зелёным цветом;
    b. если блок частично совпадает с элементом шаблона, то жёлтым выделяются схожие части блока;
    c. если блок не найден, выделяется красным.
8. В конце оба документа отображаются пользователю в виде: шаблон отображается слева без каких-либо пометок, а документ с пометками открывается справа;
9. Возможности после сравнения документов:
    a. возможность редактирования документа, при этом входной документ остаётся неизменным;
    b. возможность сохранения документа с пометками;
    c. возможность сохранения документа после внесения правок, как с пометками, так и без них;
    d. возможность быстрого поиска по средствам клика на элемент (блок) шаблона (исходного документа) или на участок элемента (блока) шаблона (исходного документа).

Второй вариант.

1. Шаблон загружается в программу;
2. Пользователь выделяет важные ему структурные элементы;
3. Пункты 4-9 первого варианта.

Третий вариант.

1. Пункты 1-2 первого варианта;
2. Отображается два поля в одном окне, слева шаблон, справа пока чистый лист, далее будет понятно;

3. Документ считывается поабзацно, то есть считываются как блоки: заголовок, основной текст, нумерованный список, таблица и так далее;
4. Каждый блок заносит в правую часть окна (там, где чистый лист) и выделяется по принципу оценки схожести пункта 7 первого варианта;
5. Пункты 8-9.

Анализ документа проводится в следующем виде:

- Заголовки и текст следующий до следующего структурного элемента выделяются как отдельный блоки;
- Отдельными блоками выделяются таблицы, нумерованные списки, изображения и остальные структурные элементы и включаются в блоки к заголовкам, в виде под блоков;
- Отдельные под блоки так же сравниваются с под боками шаблона, но при не нахождении их в шаблоне выделяются серым цветом, т.к. могут не являться структурным элементом документа и быть дополнительной информацией основного блока.

Это делается в следствие того, что в разметке текстовых документов, текст отделяется параграфами, а параграфы создаются для каждого нового элемента и если это сплошной текст, то параграфы создаются для каждого абзаца отдельно. Так после анализа документа мы получил документ, разделённый на блоки и при сравнении это упростит задачу, так как появится возможность точнее определять различия между документом и шаблоном.

Оценка схожести окрашивает блок в определённые цвета, тем самым в явном виде даёт знать пользователю какой блок был изменён, а какой остался без изменения. Зелёным будет выделятся весь блок, при условие их полного совпадения. При частичном совпадении блок и части, не совпадающие с оригиналом, будут выделятся жёлтым, а части идентичные оригиналу зелёным. Если найденные структурные элементы не совпадают с блокам шаблона, то они выделяются красным.

Так как проектируется надстройка для текстовых процессов, шапка текстовых процессов останется, а вот рабочее поле будет разделено на две части в виде раскрытой тетради. Слева будет отображаться шаблон с возможностью прокрутки, справа текст с пометками. Также будут иметься возможности: быстрого поиска, по средствам клика на блок или на элемент блока; редактирования документа с метками; возможность сохранения отредактированного документа как с метками, так и без них и сохранения документа с метками без редактирования.

Также будет сохранен (по возможности) функционал текстового процессора.

Блоки документов будут сохраняться в папке с шаблоном и удаляться по завершению работы. Это делается для безопасности и возможности повторного анализа уже исправленного документа, так как будут сохранятся выделенные блоки шаблона это позволит не проводить повторно анализ шаблона.

Сравнивать приложение будет только данные шаблона, то есть текст, идущий после структурного элемента рассматриваемого блока не будет рассматриваться при анализе, так как в данной работе рассматривается структурный анализ.

При сравнение структурных элементов будет рассматриваться не только содержание, а также стили и расположение объектов. Приложение будет показывать специальным знаком (стрелка вверх или вниз) позицию откуда был перемещён блок.

4 ВЫВОДЫ

Результаты проделанной работы продемонстрированы в таблице 3.

Таблица 3 - Результаты проведенных работ в соответствии с техническим заданием

№ п/п	Наименование работ	Ожидаемые результаты	Результаты работ	Соответствие техническому заданию
1	Разработка алгоритма сравнения документа с шаблоном	Разработан алгоритм сравнения документа с шаблоном	Алгоритм сравнения документа с шаблоном	Соответствует
2	Разработка способа выделения изменённого текста, оставшегося без изменений и удалённого	Способ выделения изменённого текста, оставшегося без изменений и удалённого	Для выделения используется цвета: Зелёный, для блока без изменений; Жёлтый, для блока с изменениями; Красный, для удалённого блока	Соответствует
3	Разработка способа создания новых шаблонов	Способ создания новых шаблонов	Разработан способ внесения изменений в шаблон из документа	Соответствует
4	Разработать способ быстрого поиска по шаблону и документу	Способ быстрого поиска по шаблону и документу элементы содержания	Был спроектирован поиск по клику на блок или на элементы содержания	Соответствует

Продолжение таблицы 3

№ п/п	Наименование работ	Ожидаемые результаты	Результаты работ	Соответствие техническому заданию
5	Рассмотрение двух текстовых формата документов docx и odt	Рассмотрены два текстовых формата документов docx и odt	После рассмотрения форматов, было принято решение использовать особенность форматов текстовых документов	Соответствует

Вывод: в ходе проделанной работы был выполнен весь объем работ в соответствии с техническим заданием.

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы было спроектировано приложение структурного сравнения документа с шаблоном.

Был произведен анализ предметной области, написано техническое задание на приложение, изучены форматы текстовых документов docx и odt и было спроектировано приложение.

Данное приложение имеет широкий круг применения, от учебных заведений до корпоративного сектора. Данное приложение может применяться при оформление договоров, технических заданий, рабочих и учебных планов.

Данное приложение позволит ускорить процесс создания документов за счёт быстрой проверки документа на ошибки и наличие нужных структурных элементов, а также даёт возможности редактирования и внесения исправлений без использования нескольких программ. Также позволяет создавать новые шаблоны на основании имеющегося шаблона и документа.

Данный программный продукт является классическим примером информационных технологий, т.к. осуществляет процессы сбора, хранения, обработки и предоставления информации конечному пользователю.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Формализованный метод разбора и анализа нормативных и методических документов, а также синтеза на их основе локальных актов [Электронный ресурс] URL: https://habrahabr.ru/post/285922/ (дата обращения: 19.12.2017).
2. Латентно-семантический анализ [Электронный ресурс] URL: https://habrahabr.ru/post/110078/ (дата обращения: 19.12.2017).
3. Gorshkov_Ispravlenia.indd [Электронный ресурс] URL: http://www.socioprognoz.ru/hta_sh/textbook/text/chapter8.pdf (дата обращения: 19.12.2017).
4. Иудин.rtf [Электронный ресурс] URL: http://window.edu.ru/resource/887/79887/files/unn2010_89.pdf (дата обращения: 19.12.2017).
5. Компьютерный анализ текста [Электронный ресурс] URL: http://computi.ru/kompeyuternij-analiz-teksta.html (дата обращения: 19.12.2017).
6. MSDN – сеть разработчиков Microsoft [Электронный ресурс] URL: https://msdn.microsoft.com/ru-ru (дата обращения: 19.12.2017).