

## ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на проект

### «Разработка программного продукта по конвертированию входных данных согласно отображению \*.docx (\*.odt) – \*.xml»

**Цель проекта:** требуется разработать Программу, позволяющую по входным данным формата \*.docx (при возможности – также с \*.odt) сгенерировать конечный продукт в виде файла формата \*.xml.

**Исходная информация.** Имеется файл формата \*.docx или \*.odt. Требуется, чтобы Программа позволяла, согласно описанию ниже, получить на своём выходе файл формата \*.xml требуемой внутренней топологии.

**Описание.** *Текущий технологический процесс.* В настоящее время имеется следующая, представленная ниже, схема достижения текущей цели проекта (рис. 1).

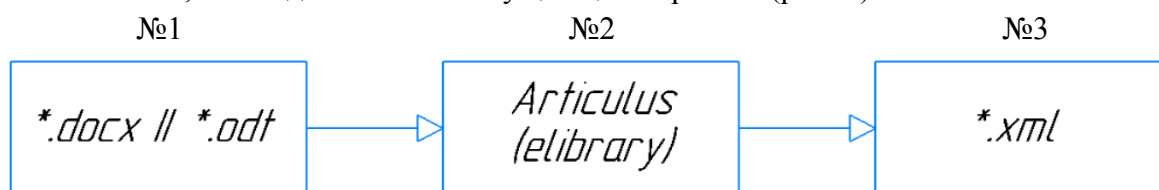


Рис. 1 – Схема достижения текущей цели проекта

В качестве исходных данных выступает файл формата \*.docx или \*.odt (блок №1). Файл представляет собой сборник научных статей в количестве порядка 60 штук и объемом 300-400 страниц. Количество статей может изменяться. Каждая статья имеет одинаковый паттерн, и его структура не меняется от статьи к статье. Работа происходит следующим образом в хронологическом порядке:

1. Запускается веб-сервис «Articulus» (блок №2).
2. Запускается исходный файл.
3. В веб-сервисе создается проект, целевая задача которого заключается в формировании конечного xml-файла.
4. Производится импорт определенных полей из файла в соответствующие поля веб-сервиса <!>.
5. По команде веб-сервиса формируется конечный xml-файл с наперед заданной веб-сервисом топологией (блок №3).

*Ограничения текущего технологического процесса.* Текущий процесс весьма затратный по времени и объёму работы, в связи с чем возникла необходимость его оптимизации средствами автоматической генерации требуемого xml-файла. Львиную долю времени занимает настройка требуемого выходного xml-паттерна в блоке №2. Настройка, подразумевающая импорт данных по одноименному паттерну, выполняется ручным образом.

*Желаемый технологический процесс.* Выходная работа разрабатываемой в рамках текущего проекта Программы должна выглядеть следующим образом. Программа должна совмещать в себе работу вышеприведённых блоков рис. 1 таким образом, чтобы пользователь действовал согласно схеме (рис. 2).

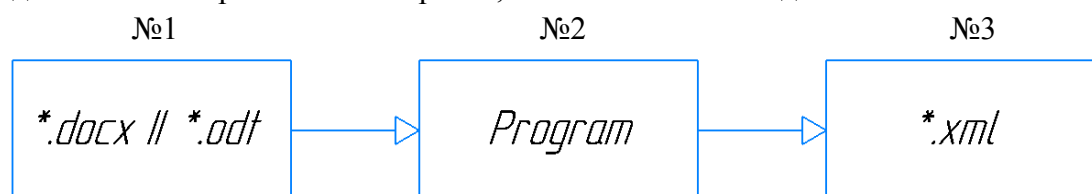


Рис. 2 – Желаемая схема достижения текущей цели проекта

Формат Программы должен представлять собой графический интерфейс, пример шаблона которого представлен ниже (рис. 3).

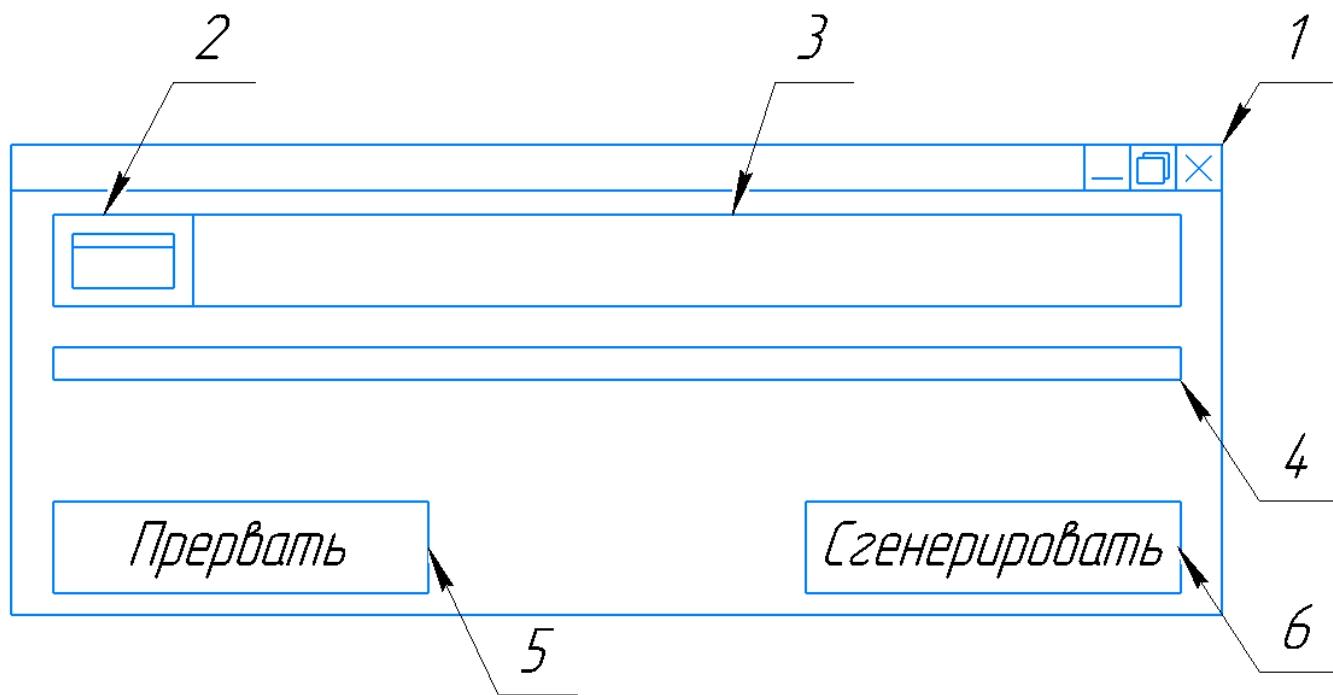


Рис. 3 – Шаблон графического интерфейса разрабатываемой Программы

Цифрами обозначены следующие структурные элементы:

1. графическое окно;
2. окно поиска «Обзор», при нажатии на которое операционная система даёт возможность выбрать файл для последующей его трансляции в xml-формат;
3. окно строки пути, в котором прописывается полный путь транслируемого файла;
4. строка состояния (status bar) процесса, оценивающая общий прогресс трансляции в режиме реального времени;
5. кнопка «Прервать», которая прерывает процесс трансляции при необходимости. Кнопка должна быть активной только в процессе трансляции и требовать пользовательское подтверждение действия подобно правилу «Вы уверены? – Да/Нет»;
6. кнопка «Сгенерировать», которая начинает процесс трансляции. Кнопка должна быть активной только после загрузки в Программу транслируемого файла (после появления пути файла). При её нажатии должен последовать запрос пользовательского подтверждения действия подобно правилу «Вы уверены? – Да/Нет».

<!>: Ниже представлен форматный xml-шаблон одной статьи, по которому заполняются данные, соответствующие импортируемые из статьи сборника и который генерируется самим веб-сервисом «Articulus». Знак «...» отражает окаймляемыми его тегами соответствующую информацию. Пример: <pages> ... </pages> – здесь знак «...» означает номера страниц, соответствующие текущей статье из сборника. Комментарии в xml-тексте выделены как «(<комментарии>)».

```
<article>
  <pages>...</pages>
  <artType>PRC</artType>
  <authors>
    <author num="001" id="...">
      <individInfo lang="RUS">
        <surname>...</surname>
```

```

    <initials>...</initials>
    <orgName>...</orgName>
    <address>...</address>
  </individInfo>
  <individInfo lang="ENG">
    <surname>...</surname>
    <initials>...</initials>
    <orgName>...</orgName>
    <address>...</address>
  </individInfo>
</author>
<author num="002" id="...">
  <individInfo lang="RUS">
    <surname>...</surname>
    <initials>...</initials>
    <orgName>...</orgName>
    <address>...</address>
  </individInfo>
  <individInfo lang="ENG">
    <surname>...</surname>
    <initials>...</initials>
    <orgName>...</orgName>
    <address>...</address>
  </individInfo>
</author>
</authors>

```

**(количество author-ов может изменяться)**

```

<artTitles>
  <artTitle lang="RUS">...</artTitle>
  <artTitle lang="ENG">...</artTitle>
</artTitles>
<abstracts>
  <abstract lang="RUS">...</abstract>
  <abstract lang="ENG">...</abstract>
</abstracts>
<text lang="RUS">...</text>
<codes>
  <udk>...</udk>
</codes>
<keywords>
  <kwdGroup lang="ANY">
    <keyword>...</keyword>
    <keyword>...</keyword>
  </kwdGroup>
</keywords>

```

**(количество keyword-ов может изменяться)**

```

<references>
  <reference>...</reference>
  <reference>...</reference>
</references>

```

**(количество reference-ов может изменяться)**

```

<files>
  <file desc="fullText">...</file>
</files>
</article>

```

К техническому заданию, в качестве примера, прилагаются следующие файлы:

1. сборник статей в формате \*.docx;
2. xml-файл требуемой внутренней топологии, получаемого после работы Программы.

**Используемый язык программирования** – Python.

**Сроки выполнения:** по договоренности.

**Цена реализации проекта:** по договоренности.