**Отчет по заданию от Компани Тензор.**

Оглавление

[Формулировка задания 1](#_Toc57988671)

[Описание Алгоритма 1](#_Toc57988672)

[Идея работы алгоритма: 1](#_Toc57988673)

[Алгоритм работы функции sample\_file (name): 2](#_Toc57988674)

[Алгоритм работы функции beautiful\_string (url – адрес директории для записи файла, text- строка которую нужно вывести по образцу форматирования, job – массив данных где первая переменная длина строки, вторая – кол-во пустых строк после абзаца, третья – кол-во отступов в красной строке): 2](#_Toc57988675)

[Исходный код 2](#_Toc57988676)

[Исполняемая утилита 2](#_Toc57988677)

[Примеры работы алгоритма 2](#_Toc57988678)

[Направления дальнейшего развития программы 2](#_Toc57988679)

# Формулировка задания

Большинство веб-страниц сейчас перегружено всевозможной рекламой... Наша задача «вытащить» из веб-страницы только полезную информацию, отбросив весь «мусор» (навигацию, рекламу и тд).

Полученный текст нужно отформатировать для максимально комфортного чтения в любом текстовом редакторе. Правила форматирования: ширина строки не больше 80 символов (если больше, переносим по словам), абзацы и заголовки отбиваются пустой строкой. Если в тексте встречаются ссылки, то URL вставить в текст в квадратных скобках. Остальные правила на ваше усмотрение.

Программа оформляется в виде утилиты командной строки, которой в качестве параметра указывается произвольный URL. Она извлекает поэтому URL страницу, обрабатывает ее и формирует текстовый файл с текстом статьи, представленной на данной странице.

В качестве примера можно взять любую статью на lenta.ru, gazeta.ru и тд

Алгоритм должен быть максимально универсальным, то есть работать на большинстве сайтов.

Усложнение задачи 1: Имя выходного файла должно формироваться автоматически по URL.

Примерно так:

http://lenta.ru/news/2013/03/dtp/index.html => [CUR\_DIR] /lenta.ru/news/2013/03/dtp/index.txt

Усложнение задачи 2: Программа должна поддаваться настройке – в отдельном файле/файлах

задаются шаблоны обработки страниц.

## Описание Алгоритма

## Идея работы алгоритма:

1. Скачиваем html код по url ссылки и записываем его в временный файл
2. Открываем временный файл на чтение и считываем html код
3. Выбираем из файла строки кода в которых чаще всего храниться текст статьи с помощью соответствующих тегов
4. Вызов функции get\_hyper\_link ддя получение массива ссылок и индекса их вождения в текст. Создание массива и поиск по html коду ключевых слов для ссылки, затем поиск места окончания ссылки и вставка ее и место начало ее вхождения в массив. Возвращает двумерный массив из пар ссылка и начало ее вхождения. Где идет сначала начало вхождения затем ссылка в квадратных скобках.
5. Вызов функции mesto для получения массива слов перед которыми надо вставить url ссылку. Поиск места, перед которым необходимо вставить url ссылку с помощью анализа кода. Передает массив, полученный из предыдущей функции
6. Обработка ответа шага 3 и извлечение текста из него
7. Усложнение 1: в функции Create\_url\_name создание директории и файла как в указанной ссылке url ссылки, разбиение на директории относительно слешей и название файла относительно название строки. Функция возражает адрес директории и файла для записи
8. Усложнение 2: Выбор пользователем необходимости воспользоваться форматированием по образцу другого документа или воспользоваться стандартным форматированием из основного условия
9. Проход по массиву текста, полученного на шаге 6
10. Проход по массиву слов перед которыми необходимо вставить ссылку и вставка при нахождении подходящего текста
11. Вызов функции beautiful\_string с передачей адреса файла, текста и массива из 3 элементов для редактирования (или стандартный или полученный из образца)

## Алгоритм работы функции sample\_file (name):

Name – адрес директории в которой находиться файл

1. Открытие файла по адресу name на чтение
2. Чтение файла по строкам и запись из в массив
3. Проход по массиву с анализом каждой строки
4. Если строка не пустая то просмотр всех строк с поиском максимальной длины строки
5. Если строка пустая то подсчет кол-во пустых строк . Тем самым мы анализируем кол-во пустых строк после абзаца
6. Анализ кол-во отступов в красной строке
7. Возвращает массив из максимальной длины, кол0во отступов и кол-во отступов в красной строке

## Алгоритм работы функции beautiful\_string (url – адрес директории для записи файла, text- строка которую нужно вывести по образцу форматирования, job – массив данных где первая переменная длина строки, вторая – кол-во пустых строк после абзаца, третья – кол-во отступов в красной строке):

1. Попытка открытия файла на добавления теста в конец его, если файл не создан создание и открытие его на запись
2. Пока строка не пустая и не противоречит правила форматирования запись строки по образцу
3. Если длина оставшейся строки больше максимальной строки, то поиск последнего слова и перенос по нему, иначе если длина строки больше, то нахождение конца ссылки которая не позволяет перенести текст и перенос после ссылки
4. Вывод оставшейся части строки

# Исходный код

Исходный код находиться в директории в файле tensor\_task.py

# Исполняемая утилита

Утилита находиться в директории в файле tensor\_task.exe

# Примеры работы алгоритма

Полученные ответы находиться в папке с именем статьи:

1. <https://meduza.io/news/2020/12/03/v-merii-moskvy-poobeschali-dobavit-troyku-v-apple-pay-i-koshelki-platezhnyh-sistem-samsung-i-google>
2. <https://meduza.io/news/2020/12/04/ilya-azar-podal-isk-k-rossii-v-espch>
3. <https://lenta.ru/news/2020/12/04/ukos/>
4. <https://lenta.ru/news/2020/12/04/getout/>
5. <https://lenta.ru/news/2020/12/04/usd/>
6. <https://panorama.pub/52937-gitler-osudil-politiku-holokosta.html>
7. <https://panorama.pub/52913-germaniya-vydelit-navalnomu.html>
8. <https://panorama.pub/49556-o-vosstanovlenii-dvoryanskih-titulov.html>
9. <https://www.championat.com/football/article-4207233-zherebevka-chm-2022-7-dekabrja-soperniki-rossii-otkuda-proizoshlo-vyrazhenie-gruppa-smerti.html>
10. <https://www.championat.com/football/article-4207305-zenit-cska-lokomotiv-i-krasnodar--v-tablice-koefficientov-uefa-lch-i-le.html>
11. <https://life.ru/p/1357274>
12. <https://life.ru/p/1357266>

# Направления дальнейшего развития программы

1. Увеличения процента сайтов работы скрипта – так как данный скрипт работает на 50-60 % сайтов, в идеале использования машинного обучения для анализа
2. Добыча картинок и видео и вставка их в файл или в директорию нахождения статьи
3. Создания облачного директория для хранения файлов
4. Скачивание и форматирование всех статьей за определенный срок с содержанием файлов
5. Использования алгоритма на агрегатах сайтов по типу Яндекса для формирование и анализа мнения о проблеме