# Тестовое задание C++ Задача C++|Python – Mini readability

Описание варианта реализации

Бардушко Н.М.

Казань 2015

# Содержание

# Назначение программы

Программа предназначена для получения содержимого статьи, размещенной на интернет сайте. Программа запускается в качестве утилиты командной строки, принимая в качестве параметре URI статьи и сохраняя переработанное содержимое в виде текстового файла.

Содержимое страницы сохраняется не в полном объеме. Выделяется лишь значимое, с точки зрения алгоритма, определенного программой, содержимое.

# Структура программы

Программа состоит из следующих структурных элементов

1. Блок http-клиента.
2. Блок алгоритма обработки статьи.
3. Блок работы с файловой системой.
4. Блок настроек.

## Http-клиент

Данный блок предназначен для получения в виде байтового массива ответа от http сервера. В качестве входного параметра принимает строку – URI запрашиваемой статьи. На выходе – байтовый массив, содержащий ответ сервера.

## Блок алгоритма обработки статьи

Назначение – выделить из всего содержимого html страницы только значимый текст. Использованный в данной программе алгоритм выделяет заголовок и содержимое статьи.

Адреса ссылок, расположенных в тексте записываются в текст по мере встречи в квадратных скобках.

### Заголовок статьи

В качестве заголовка принимается весь текст расположенный между тегами <title>и </title>. Данный тег является уникальным в пределах страницы, что обеспечивает однозначность определения заголовка, в отличие от тегов <h1></h1> количество которых не регламентировано.

Кроме того, из соображений SEO оптимизации содержимое тега <title> наиболее релевантное по отношению к контента статьи.

### Текст статьи

Выделение текста статьи менее очевидно. Предлагается в качестве контента статьи брать содержимое тегов . Учитываются все теги , встреченные на странице в порядке их следования.

Параметры тегов игнорируются.

### Ссылки

Учитываются только те ссылки, которые расположены внутри тегов . Текст ссылки оставляется как есть, а адрес ссылки добавляется в квадратных скобках сразу после текста ссылки без пробела.

Порядок определения текста ссылки такой:

1. Ищется в тексте вхождение строки **</a** .

2. Начиная, с найденной на шаге 1 позиции ищется ближайшее вхождение символа >. Таким образом, игнорируются все возможные параметры ссылки.
3. В качестве текста ссылки принимается весь текст, расположенный от позиции, найденной на шаге 2 до ближайшего вхождения строки **</a>.**

Возможные вхождения внутри найденного текста ссылки других тегов (например **<img/>**) в данной версии алгоритма никак не учитывается и переносятся в итоговый текст как есть.

Определение адреса ссылки

1. Ищется в тексте вхождение строки **<a/**.
2. Начиная с позиции, найденной на шаге 1, ищется строка **href**.
3. Ищется ближайшее вхождение символа **“**.
4. В качестве адресы ссылки принимается весь текст начиная с позиции, найденной на шаге 3 до ближайшего вхождение символа **“**.

# Системные требования

Программа рассчитана на работу в операционной системе Windows.

# Порядок работы с программой

## Установка программы

Установка данного программного обеспечения выполняется путем копирования папки, файлов articleGrabber.exe и settings.txt на компьютер пользователя.

## Запуск

Запуск программы осуществляется в режиме командной строки. В качестве параметра необходимо указать URI обрабатываемой статьи.

Допускается вариант запуска программы без параметров. В этом случае адрес ресурса будет определен из файла settings.txt.

## Настройка

Настройки системы хранятся в файле settings.txt. Настройки содержат параметры, записанный в формате ключ значение, разделитель пробел. В данной версии используются следующие параметры:

1. uri – адрес ресурса для обработки
2. lineLength – длина строки в обработанном тексте.

# Варианты возможного дальнейшего развития программы

## Кроссплатформенность исходных кодов

Текущая версия исходных кодов программы в части блоков http-клиента и блока работы с файловой системой ориентированы на работу в операционной системе Windows.

## Возможность использования различных алгоритмов

В текущей версии используется единственный алгоритм разбора статьи.

## Дополнительное логирование системы

С целью упрощения обслуживания программы предусмотреть вывод в журнал системных диагностических сообщений с фактами запуска, предупреждениями и ошибками.

# Приложение

## Состав исходных кодов

Таб. 1 Исходные коды в составе

| № | Имя файла | Описание |
|---|---|---|
| 1 | main.cpp | Содержит точку входа программы |
| 2 | HttpClient.h | Определение класса HttpClien, предназначенного для получения кода html страницы по указанному URI |
| 3 | httpClient.cpp | Реализация методов класса HttpClient |
| 4 | FileHelper.h | Определение класса для сохранения данных в файловую систему FileHelper |
| 5 | fileHelper.cpp | Реализация методов класса FileHelper |
| 6 | CreateDir.h | Определение класса CreateDir, предназначенного для создания структуры каталогов хранения полученных статей. |
| 7 | createDir.cpp | Реализация методов класса CreateDir |
| 8 | HtmlParser.h | Определение класса HtmlParser, содержащего алгоритм разбора кода страницы и выделение из него текста статьи с заголовком. |
| 9 | htmlParser.cpp | Методы класса HtmlParser. |
| 10 | Settings.h | Определение класса для работы с настройками программы. |
| 11 | settings.cpp | Реализация методов класса Settings. |

## Список протестированных URI

Таб. 2. Протестированные URI

| № | URI | Имя файла |
|---|---|---|
| 1 | http://lenta.ru/lenta.ru/news/2015/04/04/lavrov_crimea | \lenta.ru\news\2015\04\04\lavrov_crimea\2015_04_05__11_39_21.txt |
| 2 | http://lenta.ru/news/2015/04/04/germanwings | \lenta.ru\news\2015\04\04\germanwings\2015_04_04__05_52_14.txt |
| | | |