ISSN 2311-8814

«НАУКА | RASTUDENT.RU»

Электронный научно-практический журнал

График выхода: ежемесячно
Языки: русский, английский
ISSN: 2311-8814
Издатель: компания INFLASH
Учредитель: ИП Соколова А.С.
Место издания: г. Уфа, Российская Федерация
Прием статей по e-mail: rastudent@yandex.ru
Место издания: г. Уфа, Российская Федерация

Щеглов В.Ю. Выбор алгоритма выделения кода из документов со смешанным содержимым // Наука-RASTUDENT.RU. – 2014. – No. 4(04-2014) / [Электронный ресурс] – Режим доступа. – URL: http://nauka-rastudent.ru/4/1308/

УДК 004

Щеглов Владислав Юрьевич,
магистрант
Кафедра Вычислительной Техники, Факультет ЭСТ
Московский Государственный Университет Леса
Мытищи, Московская область, Россия

Выбор алгоритма выделения кода из документов со смешанным содержимым

Аннотация: В данной статье речь идёт о выборе алгоритма для извлечения кода из документов со смешанным содержимым. Рассматриваются метод регулярных выражений, поиск конструкций языка по их местоположению в тексте, построчный анализ текста на предмет содержания кода с строке по ключевым словам, построчный анализ текста с подсчётом в каждой строке количества ключевых и не ключевых слов.

Ключевые слова: выделение кода, анализ текста, выбор алгоритма.

The choice of algorithm extraction code of documents with mixed content

Shcheglov Vladislav Yurevich,
Master student
Department of Computer Engineering, Faculty of Electronics and Systems
Moscow State Forest University
Mytischi, Moscow region, Russia

Abstract: In this article we are talking about choosing an algorithm to extract the code from the documents with mixed content. We consider a method of regular expressions, search language constructs based on their location in the text, progressive analysis of the text for the content code line by keyword, progressive analysis of the text with the counting in each line the number of key and non-key words.

Keywords: extraction code, text analysis, choice of algorithm.

Исследуя проблему выделения кода из документов со смешанным содержимым и перейдя непосредственно к реализации решения в виде программы, было разобрано несколько алгоритмов, которые и будут рассмотрены в данной статье. Здесь также стоит добавить, что в целом было решено использовать подход шаблонов для каждого отдельного языка программирования на основе другого исследования.

В качестве первого варианта решения были взяты регулярные выражения, каждое из которых искало бы конкретные конструкции языка. Преимущество этого метода в том, что код будет выглядеть минималистично, однако существуют конструкции типа

“if(<множество выражений>)

{

<множество операций>

}

else

{

<множество операций>

}”,

где среди множества операций может быть вложена одна или несколько таких конструкций, в которых в свою очередь могут быть вложены такие конструкции и так далее. Для таких конструкций регулярные выражения уже отнюдь не тривиальны и написание их для всех видов конструкций может занять много времени, а эффективность находится под вопросом, ведь в тексте далеко не всегда встречаются законченные конструкции, там может отсутствовать какая-нибудь скобка и регулярное выражение даст сбой.

Другой вариант решения – нахождение строк по индексам ключевых слов. Например, возьмём конструкцию вида “using <не ключевые слова>;”. Ищем в тексте индекс (местоположение) слова using, далее начиная от него ищем первое вхождение символа ‘;’. Таким образом находим нужную нам строку. С начала может показаться, что у этого метода нет ни одного преимущества, но проблема в том, что в тестовых документах не всегда строка заканчивалась символом ‘\n’ и/или ‘\r’. Регулярные выражения не могли корректно распознать окончание строки. Этим же методом мы просто находим ближайший символ ‘;’ которым заканчивается рассматриваемый блок кода. Нахождение данным способом больших конструкций представляет большие сложности, но главным недостатком является то, что выделенный итоговый код будет перемешан, так как получение строк может происходить не в том порядке в котором они идут в тексте.

Третьим вариантом решения стало построчное чтение текста, проверяя каждую строку, является ли она частью кода на искомом языке. Определение, является ли строка частью кода, происходит поиском ключевых слов и символов в неё. Например, если мы ищем код на языке C#, рассматриваемая строка начинается со слова “using”, а заканчивается на символ ‘;’, то скорее всего это часть искомого кода. Однако, некоторые конструкции распознать бывает тяжело, и либо приходится жертвовать такими строками, либо алгоритм захватывает и часть обычного текста.

Взяв за основу третий вариант, был разработан четвёртый, окончательный вариант решения. Текст также просматривает построчно, но каждая строка делится на слова (признаком разделения строк является символ пробела ‘ ‘). Просматривая каждое слово, мы определяем, является ли оно ключевым для рассматриваемого языка. Если ответ положительный, то мы увеличиваем счётчик «ключевых» слов, если нет, то увеличиваем счётчик «не ключевых» слов. После того, как мы просмотрели все слова в строке и получили количество «ключевых» и «не ключевых» слов, мы можем

посчитать процент «ключевых» слов в строке. Опытным путём было установлено, что 15% «ключевых» слов практически всегда означает, что перед нами строка с кодом. Точность данного решения намного выше всех предыдущих, а код на выходе получается не перемешанным.

Дата публикации: 11.04.2014