Отчет

Пояснения регулярных выражений

1. i. Регулярное выражение для выбора слов с латинскими буквами и цифрами:
   \b[a-zA-Z0-9]+\b
   Находит слова только состоящие из латинских букв и цифр.
2. j. Регулярное выражение для выбора слов, имеющих хотя бы одну латинскую букву:
   \b\w*[a-zA-Z]\w*\b
   Находит слова, содержащие хотя бы одну латинскую букву. Остальные символы слова могут быть любыми буквами, цифрами или подчеркиваниями.
3. l. Регулярное выражение для выделения всех листовых тегов:
   <(\w+)>([^<]*)<\/\1>
   Находит теги, содержащие только текст без вложенных тегов. Используются группы для захвата имени тега и его содержимого.

Ответы на вопросы из пункта m

a) Что означает символ @ в начале строкового литерала в C#? Как будет выглядеть литерал @"(\W*)(\w+)(\W+)(\w+)(\W+)(\w+)(\W*)" без использования символа @?
@ означает, что строка является Verbatim строкой, содержимое которой не экранировано.
"(\\W*)(\\w+)(\\W+)(\\w+)(\\W+)(\\w+)(\\W*)"

b) \w* — ноль или более символов (буквы, цифры, подчеркивание).
\W* — ноль или более не-символов.
\w+ — один или более символов.
\W+ — один или более не-символов.

c) Что означает шаблон [^<>]*?
Шаблон [^\s<>]* соответствует последовательности символов, которая не содержит пробелов и угловых скобок.

d) Как сделать именованную группу?
Именованная группа создается с использованием синтаксиса (?<name>pattern)

e) Где использованных выше выражениях использовались:
   - Исчислители: \w+, \W+
   - Классы символов: \w, \W
   - Обычные символы: [a-zA-Z]
   - Обратные ссылки: (\w)\1
   - Подстановки: "$1$4$3$2$5$6$7"

Регулярные выражения для решения практической задачи:

1. Поиск пустых тегов <main-word>: **<main-word>\s*<\/main-word>**
2. Поиск пустых тегов с контекстом: **.{0,20}<main-word>\s*<\/main-word>.{0,50}**
3. Выделение слов за пустым тегом <main-word>: **<main-word>\s*<\/main-word>\s*(\w[\w\u0301-]*)**
4. Исправление дефектов, где слово за пустым тегом <main-word>:
   Regex re = new Regex(@"<main-word>\s*<\/main-word>\s*(\w[\w\u0301-]*)");
   string transformed = re.Replace(orig, @"<main-word><bold>$1</bold></main-word>");
5. Выделение слов перед пустым тегом <main-word>: **(\w[\w\u0301-]*)\s*<main-word>\s*<\/main-word>**
6. Исправление дефектов, где слово перед пустым тегом <main-word>:
   Regex re = new Regex(@"(\w[\w\u0301-]*)\s*<main-word>\s*<\/main-word>");
   string transformed = re.Replace(transformed, @"<main-word><bold>$1</bold></main-word>");

Выявление дефектов и их устранение

Было выявлено несколько дефектов пустых тегов <main-word>, как со словами после них, так и со словами перед ними. Все эти дефекты были успешно исправлены.