

ОБРАБОТКА ТЕКСТА С ИСПОЛЬЗОВАНИЕМ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ

В статье рассматриваются основные технологии работы с регулярными выражениями. Приведено описание разработанного авторами на языке C# программного обеспечения, позволяющего выполнять различные манипуляции с текстом в формате .txt и .doc.

Ключевые слова: регулярное выражение, шаблон, метасимвол, модификатор, класс, метод, алгоритм, интерфейс.

Регулярные выражения (regular expression) – это мощный инструмент для обработки больших текстов [1, 2]. Они позволяют описать шаблон (pattern), на основании которого производятся такие операции с текстом, как поиск, замена или удаление определенных текстовых фрагментов, проверка соответствия строки определенным требованиям и так далее.

Популяцией регулярных выражений для обработки текстов способствовал набор утилит, поставляемых в дистрибутивах ОС Unix, в среде которой они появились. Современные языки программирования, такие, как Perl, ActionScript, JavaScript, языки платформы .Net Framework, Python, C#, Delphi и другие, содержат поддержку регулярных выражений, которая позволяет выполнять различные операции и манипуляции с текстом с использованием языка регулярных выражений, в основе которого лежат шаблоны, формируемые на основе метасимволов и модификаторов, определяющих режимы поиска соответствий; квантификаторы, задающие количество повторений; перечисления, используемые для указания альтернативных вариантов поиска;

В языке C# для работы с регулярными выражениями используется класс Regex пространства имени System.Text.RegularExpressions, который содержит такие методы, как:

- IsMatch, проверяющий принадлежность строке текста подстроки, соответствующей шаблону регулярного выражения;
- Match, возвращающий первую подстроку, соответствующую шаблону, в виде объекта класса Match, предоставляющего информацию о подстроке: длину, индекс, значение и так далее;
- Matches, возвращающий все подстроки соответствующие шаблону в виде коллекции типа MatchCollection;
- Replace, возвращающий строку, в которой все подстроки, соответствующие шаблону, заменены новой строкой;
- Split, возвращающий массив строк, полученный в результате деления входящей строки в местах соответствия шаблону регулярного выражения.

Regex в качестве второго аргумента может принимать следующие значения RegexOptions: IgnoreCase – игнорирование регистра при поиске; RightToLeft – выполнение поиска справа налево; Multiline – многострочный режим поиска;

Singleline – однострочный режим поиска; ExplicitCapture – поиск только буквенных соответствий; Compiled – компилирование регулярного выражения в сборку для более быстрого его исполнения; IgnorePatternWhitespace – игнорирование в шаблоне всех незранированных пробелов; None – использование поиска по умолчанию.

В настоящей работе рассматривается разработанное на языке C# программное приложение для работы с регулярными выражениями, позволяющее по входному шаблону найти вхождения в файле форматов .txt или .doc и вывести результаты в форму richTextBox с отображением всего текста с выделенными фрагментами, соответствующими шаблону [3].

Алгоритм работы разработанного приложения содержит следующие этапы:

1. Выбор файла или ввод строки для поиска;
2. Ввод шаблона для манипуляций с подстроками текста в поле «Регулярное выражение»;
3. Выбор действия, которое необходимо выполнить над подстрокой текста;
4. Если выбрано действие «Заменить», появляется поле «Заменить на...», куда требуется вписать подстроку замены;
5. Выполнение выбранного действия в результате нажатия кнопки «Проверить»;
6. Нажатие кнопки «Пуск» запускает на выполнение выбранный режим работы приложения.

Для закрытия приложения предусмотрена кнопка «Выход».

Интерфейс задачи построен на основе форм UserForm, содержащей элементы управления Label, TextBox, ComboBox, CommandButton, рисунок 1.

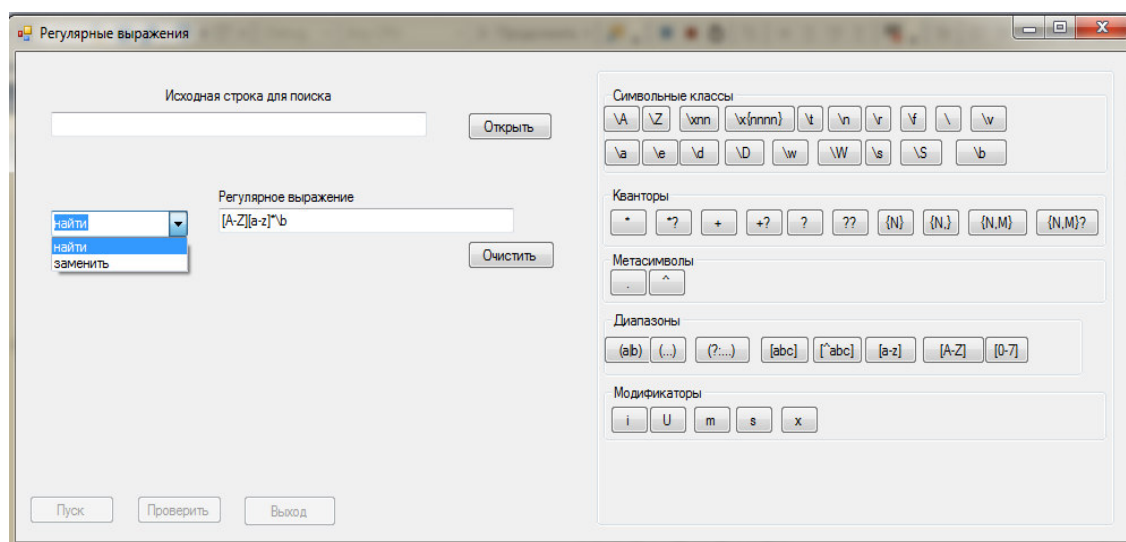


Рисунок 1 – Главное окно приложения

На основе названных элементов управления в окне интерфейса программы построены такие элементы формы «Регулярные выражения», как «Исходная

строка для поиска», «Регулярное выражение», «Символьные классы», «Кванторы», «Метасимволы», «Диапазоны», «модификаторы», а также кнопки: «Открыть», «Очистить», «Пуск», «Проверить» и «Выход», рисунок 1.

Приложение разработано на платформе .NET Framework в Windows Forms на языке C#. Главный компонент обработки текста представлен объектом System.Text.RegularExpressions.Regex.

Для работы с текстовыми файлами подключалось пространство имен System.IO, с .doc-файлами – библиотека Microsoft.Office.Interop.Word.

Приложение предусматривает два режима работы: с файлом и с текстовым фрагментом.

Если работа выполняется с текстовым фрагментом, то в текстовое поле «Исходная строка для поиска» необходимо ввести или вставить упомянутый фрагмент. При работе с файлом в это же текстовое поле следует ввести полное имя файла и кликнуть кнопку «Открыть». Затем в текстовое поле «Регулярное выражение» требуется ввести соответствующее режиму работы регулярное выражение и для проверки его на корректность можно нажать кнопку «Проверить». Кнопкой «Пуск» запускается процесс обработки текстовых фрагментов или файла.

В случае отсутствия совпадений в проверяемом файле или фрагменте на экран выводится окно MessageBox с сообщением «Совпадений не найдено», рисунок 2.

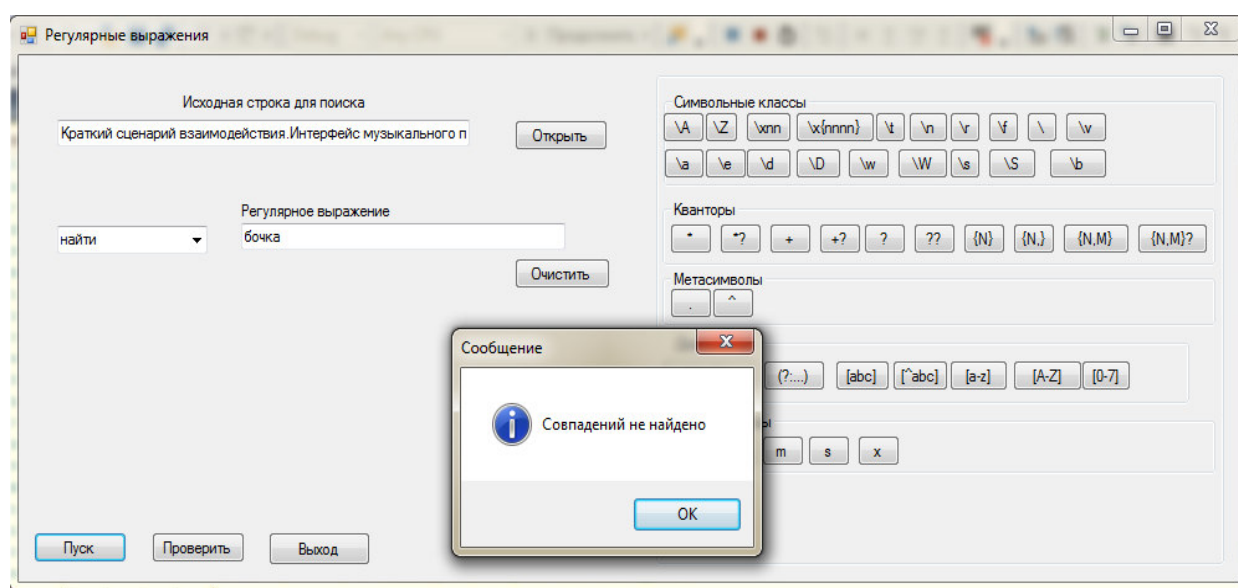


Рисунок 2 – Результат работы программы

Если требуется заменить в файле подстроку, то модифицированный файл сохраняется по новому пути, указанному пользователем в открывающемся при этом диалоговом окне «Сохранить как...».

В одном сеансе работы программа позволяет многократно выполнять операции поиска и модификации текста.

При работе с программой пользователю не требуется больших знаний о назначении каждого метасимвола в алфавите регулярного выражения, так как приложение носит и обучающий характер: при наведении курсора на кнопку с соответствующим метасимволом появляется справка с его описанием, рисунок 3. Поэтому построение шаблона для поиска становится несложным действием.

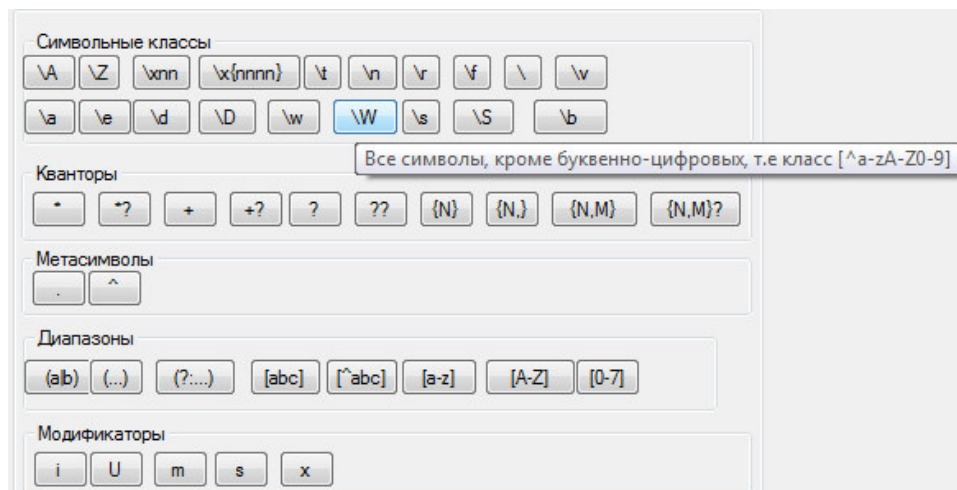


Рисунок 3 – Окно программы с алфавитом языка регулярных выражений

Таким образом, разработанное программное приложение позволяет с помощью регулярных выражений выполнять наиболее характерные операции с текстовыми строками и файлами, используется в учебном процессе при изучении темы «Регулярные выражения» дисциплины «Теория формальных языков» и позволяет значительно повысить эффективность изучения названной темы.

Список литературы

- 1 **Гойвертс, Я., Левитан, С.** Регулярные выражения. Сборник рецептов. Regular Expressions: Cookbook / Я. Гойвертс, С. Левитан – СПб.: «Символ-Плюс», 2010. – 608 с.
- 2 **Смит, Б.** Методы и алгоритмы вычислений на строках (regex) Computing Patterns in Strings / Б. Смит – М.: «Вильямс», 2006. – 496 с.
- 3 **Пасынкова Ю. В., Ясюкович Э. И.** Разработка ПО для работы с регулярными выражениями / Ю. В. Пасынкова, Э. И. Ясюкович // Материалы 53-й студенческой науч. – тех. конф., 4 – 5 мая 2017 г. – Могилев: Бел. – Рос. ун – т, 2017, – с.168,