УДК 004.62

ИЗУЧЕНИЕ ТЕХНОЛОГИИ ПАРСИНГА ЭЛЕКТРОННЫХ ПИСЕМ КАК СПОСОБА АВТОМАТИЗАЦИИ ЗАПОЛНЕНИЯ БАЗЫ ДАННЫХ.

Каплун А.В., студент гр. ИТб-171, IV курс,
Алексеева Г.А., старший преподаватель
Научный руководитель: Чичерин И.В., к.т.н., доцент
Кузбасский государственный технический университет имени Т.Ф. Горбачева
г. Кемерово

В XXI веке повсеместная автоматизация различных процессов производственной и бизнес-среды позволила сократить трудозатраты работников и повысить эффективность рабочего процесса. В связи с этим многие организации стараются автоматизировать всевозможные бизнес-процессы. К примеру, работа с электронной почтой также может быть автоматизирована посредством автоматического занесения данных из входящих электронных писем в базу данных. Для решения такой задачи используется технология парсинга данных.

Парсинг (от англ. parsing – «разбор, анализ») – процесс сопоставления строки естественного языка или языка программирования с определенным шаблоном или правилом анализа данных.

Синтаксический анализатор / парсер – подпрограмма, осуществляющая автоматизированный синтаксический и лексический анализ или разбор большого объема входных данных (чаще всего текстовых) с целью выделения из них структурированных блоков, использующихся для какой-либо дальнейшей работы [2].

Парсер является многофункциональным инструментом работы с данными. Перечислим некоторые направления его использования:

1) Парсер сайтов. Сбор информации в больших объемах в Интернете – достаточно трудоемкая задача для ручного выполнения, требующая автоматизированной реализации в виде парсера. Применяется в различных целях:

– Оперативное отслеживание новостных данных – необходимо для новостных компаний, риэлтерских агентств, компаний по перепродаже товаров;

– Парсинг аудитории - сбор данных о действующих и потенциальных клиентах по профилям в социальных сетях – быстрый способ получить контактную информацию, а также проанализировать предпочтения клиента для дальнейшей работы с ним (принцип таргетированной рекламы);

– Парсинг товаров - сбор информации о товарах конкретных магазинов с целью проведения детального анализа ценовой политики конкурентов.

2) Автоматизированные лингвистические переводчики.

3) Встроенные парсеры трансляторов (компиляторов и интерпретаторов) различных сред разработки программного обеспечения – используются для преобразования текста, записанного на каком-либо языке программирования, в машинно-ориентированный язык (например, ассемблер) [1].

4) Индустрия компьютерных игр – парсеры применяются при работе с текстовыми файлами, хранящими параметры 3D графики.

5) Синтаксический разбор баз данных, сохраненных в файлах различных текстовых форматов (CSV, XML и т.д.)

6) Программы автоматической проверки информации на текстовые заимствования (антиплагиат) – принцип их работы заключается в сравнении содержимого сотен web-страниц с проверяемым текстом.

7) Спам-рассылка – благодаря парсингу аудитории спамеры находят адреса электронной почты пользователей или номера телефонов и используют для рассылки массовых сообщений.

Независимо от того, на каком языке программирования написан парсер или для каких целей, общий алгоритм работы остаётся неизменным (рисунок 1):

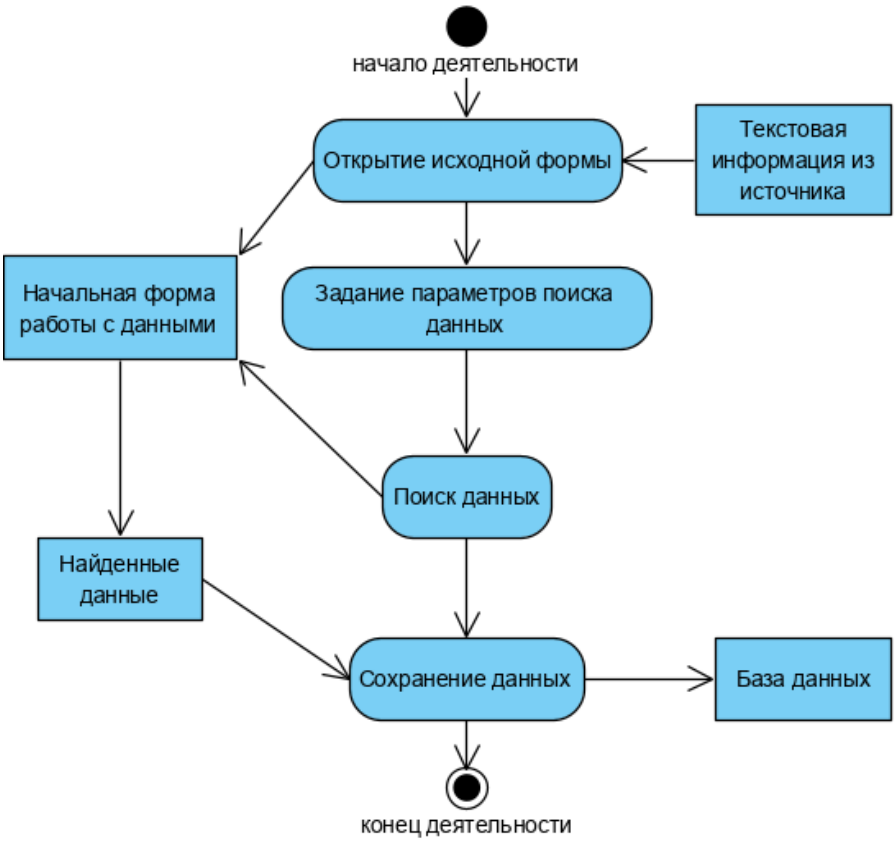

Рисунок 1 – Диаграмма деятельности процесса парсинга

1) Сбор информации – загрузка кода страницы сайта или любой текстовой информации, из которой нужно извлечь конкретные данные.

2) Поиск информации, соответствующей правилам парсера и извлечение – разбиение всего текста на лексемы, анализ, поиск необходимых участков текста. Для создания правил поиска чаще всего используются регу-

лярные выражения (англ. regular expressions) – строки или шаблоны, составленные по определенным правилам и критериям поиска нужной информации в тексте [3].

3) Сохранение результатов поиска – после извлечения необходимой информации, необходимо её сохранить. Структурированные данные оформляются в виде таблиц, записи из которых удобно заносить в базы данных.

Из выше сказанного, можно сделать вывод, что парсер значительно ускоряет и оптимизирует работу с большими объёмами данных. В качестве достоинств использования парсеров можно отметить:

– бесперебойный сбор данных любого объема в течение неограниченного времени;

– соблюдение всех параметров поиска;

– нивелировать ошибок, связанных с «человеческим фактором» – ошибки из-за невнимательности или усталости работника;

– предоставление собранной информации в любом удобном формате;

– равномерное распределение нагрузки на сайты, где происходит парсинг (одна страница за 1-2 секунды), чтобы не спровоцировать активацию эффекта DOS-атаки.

Основным недостатком, связанным с использованием парсинга, считается заимствование чужого контента и использование его в личных целях. Однако данное действие не является противозаконным, в случае если информация на каком-то Интернет ресурсе размещена в свободном доступе, а также её парсинг не нарушает следующие законодательные нормы, описывающие работу с интеллектуальной собственностью и размещаемой в сети Интернет информации:

– анализируемая информация не относится к коммерческой или государственной тайне;

– заимствование анализируемой информации не нарушает авторских или смежных прав;

– данная информация не относится к категории информации, охраняемой законом.

Также работа парсера не должна приводить к нарушениям в работе ресурсов-источников информации. Подобные сбои могут быть вызваны большим количеством подключений в секунду парсера к веб-ресурсу, что приведет к значительному увеличению потока входящего трафика на сервер. При 200-250 подключений в секунду работа парсера расценивается как DOS-атака, что может стать причиной блокировки Интернет ресурса на некоторое время. Именно поэтому существует ошибочное мнение о том, что парсер – это вредоносное ПО наравне с компьютерным вирусом. Однако отличительной особенностью вируса является его автономность и способность к «размножению». Парсер инсталлируется на персональный компьютер и не покидает его пределов.

Рассмотрим работу парсера на примере консольного приложения на языке C# в среде разработки Visual Studio 2017. Допустим, что в какой-то компании появилась необходимость в автоматической выгрузке структурированных данных из текста входящих электронных писем в базу данных. По мнению этой компании, автоматизация данного процесса поможет уменьшить трудозатраты сотрудников и соответственно оптимизировать работу компании на n%.

Предположим, что текст писем будет сохраняться в виде текстовых файлов формата .txt примерно следующего содержания (рисунок 2).

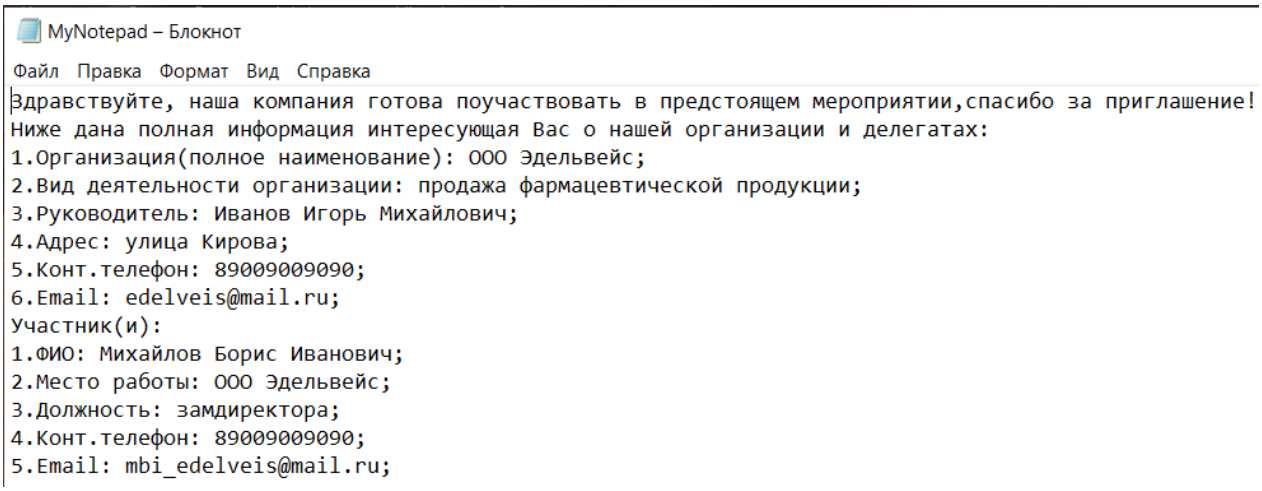

Рисунок 2 – Содержание электронного письма в формате .txt

Задачей программы-парсера является извлечение данных [4], предоставленных отправителем о компании и участниках, которые будут её представлять на предстоящем мероприятии и добавление этих данных в базу данных посредством обращения к хранимым процедурам добавления на стороне сервера.

Листинг программы с комментариями приведен на рисунках 3 и 4.

```
class Program
{
    static void Main(string[] args)
    {
        //создание экземпляра компонента для работы с данными бд
        PracticaDataSetTableAdapters.QueriesTableAdapter qta = new PracticaDataSetTableAdapters.QueriesTableAdapter();
        //создание экземпляра класса ArrayList для создания коллекции объектов разных типов данных
        ArrayList list = new ArrayList();
        //путь до файла, с которого нужно парсить данные
        string path = @"C:\Users\Анастасия\Desktop\MyNotepad.txt";
        //экземпляр класса для чтения строк тектсового файла
        StreamReader sr = new StreamReader(path);
        //правило поиска данных в тексте -регулярное выражение
        var regex = new Regex("([^:]*); ", RegexOptions.Singleline | RegexOptions.IgnorePatternWhitespace);
        //задаем строку, для которой нужно применить правило поиска
        var matches = regex.Matches(Convert.ToString(sr.ReadToEnd()));
```

Рисунок 3 – Листинг программы (часть 1)

```
        //цикл заполнения коллекции объектов и вывод на экран данных для проверки
            int i = 0;
            foreach (Match m in matches)
            {
                list.Add(m.Value.Trim(';'));
                Console.WriteLine(list[i]);
                i++;
            }
        //вызов хранимых процедур добавления данных в БД
        qta.AddCompany(Convert.ToString(list[0]), Convert.ToString(list[1]),Convert.ToString(list[2]),
        Convert.ToString(list[3]), Convert.ToInt64(list[4]), Convert.ToString(list[5]));

        qta.AddCompany_staff(qta.ID_Company_staff() + 1,Convert.ToString( list[6]), Convert.ToString(list[7]),
        Convert.ToString(list[8]), Convert.ToInt64(list[9]), Convert.ToString(list[10]));

        Console.ReadLine();
    }
}
```

Рисунок 4 – Листинг программы (часть 2)

В результате работы программы найдены необходимые участки текста и переданы в хранимые процедуры (рисунки 5, 6).

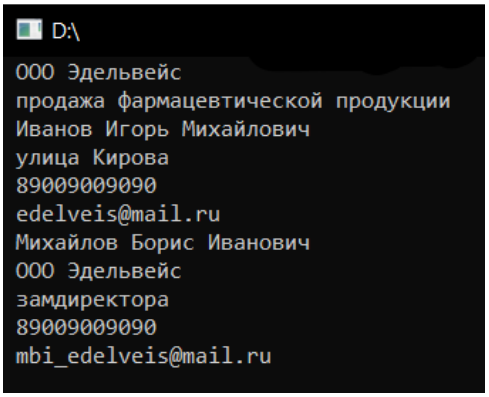

Рисунок 5 – Результат выбора текста с помощью правила

Результаты | Сообщения

	Name	Field	Boss	Address	Phone	Email
1	ООО Эдельвейс	продажа фармацевтической...	Иванов Игорь ...	улица Кирова	89009009090	edelveis@mail.ru
2	Администрация...	исполн.орган гос.власти	Цивилёв С.Е.	Кемерово пр....	3842363409	postmaster@ako.ru
3	Кемеровский г...	адм-ция муницип.образования	Середюк И.В.	Кемерово пр....	3842364610	admin@kemerovo.ru
4	Мин. жилищно-...	исполн.орган гос.власти	Ивлев О.В.	Кемерово пр....	3842583841	http://жкх.рф
5	Мин.природных...	исполн.орган гос.власти	Высоцкий С.В.	Кемерово пр....	3842585556	http://kuzbasseco.ru
6	Мин.промышле...	исполн.орган гос.власти	Старосвет Л.В.	Кемерово пр....	3842587861	http://kemdep.ru

Результаты | Сообщения

	ID...	FIO_human	Name_work	Post_human	Phone_human	Email_human
1	0	Иванов И.И.	Администрация Правительст...	секретарь	8	@
2	1	Петров А.А.	Кемеровский городской округ	-	89005006070	-
3	2	Сидоров А.К.	Мин. жилищно-коммун.и дор...	-	89995005670	-
4	3	Капралов В.Н.	Мин.природных ресурсов и э...	-	899950345670	-
5	4	Абрамов Е.А.	Администрация Правительст...	-	899950340000	-
6	5	Горбунов К.М.	Администрация Правительст...	-	89998760000	-
7	6	test1111	Кемеровский городской округ	test11111	89005006070	test1111
8	7	Михайлов Борис Ив...	ООО Эдельвейс	замдиректора	89009009090	mbi_edelveis@mail.ru

Рисунок 6 – Добавление записей о компании и участнике в соответствующие таблицы базы данных

Использование программы-парсера позволило сократить временные затраты на выбор необходимой информации и предоставить данные в структурированной форме для дальнейшего автоматического занесения их в базу данных.

Список литературы:

1. А. Ахо, Дж. Ульман. Теория синтаксического анализа, перевода и компиляции. Т. 1. Пер. с англ. В. Н. Агафонова под ред. В. М. Курочкина. М.: Мир, 1978. 614 с.
2. Синтаксический анализ [Электронный ресурс]. – URL: - https://ru.wikipedia.org/wiki/Синтаксический_анализ (дата обращения 25.03.2021)
3. Регулярные выражения [Электронный ресурс]. – URL: - https://metanit.com/sharp/tutorial/7.4.php (дата обращения 25.03.2021)
4. Чтение и запись текстовых файлов. StreamReader и StreamWriter [Электронный ресурс]. – URL: - https://metanit.com/sharp/tutorial/5.5.php (дата обращения 25.03.2021)