ФИНАНСОВЫЕ РЫНКИ

В. А. Бухвалова
PhD, Associate Professor, Department of Finance, BI Norwegian Business School

В. В. Бухвалова
канд. физ.-мат. наук, доцент кафедры исследования операций Санкт-Петербургского государственного университета

AWK: ОТ ФИНАНСОВОЙ ИНФОРМАЦИИ К ФИНАНСОВЫМ ДАННЫМ

1. Введение

Экономические и финансовые исследования часто опираются либо на публично недоступную информацию, либо на дорогие коммерческие данные. При этом огромные объемы информации доступны в Интернете абсолютно бесплатно. Однако эта информация не всегда, а точнее, почти никогда не структурирована в удобной для исследователей форме. Статьи, авторам которых удается обуздать такую информацию, появляются в топ-журналах по экономике и финансам. Эти авторы не торопятся делиться собранными данными, алгоритмами и написанными ими кодами программ для компиляции данных. В большинстве случаев читатель не сможет узнать даже, какой язык программирования был использован. Конечно, существуют исключения (Garcia, Norli, 2012a).

У каждого языка программирования есть свои достоинства и недостатки. Достоинства языка AWK— это портативность и простота кодов программ. Кроме того, AWK позволяет быстро обрабатывать большой объем текстовых файлов даже на скромных компьютерах.

AWK сравнительно мало известен, особенно среди пользователей операционной системы Windows, несмотря на то что AWK включен во все стандартные инсталляции UNIX и Linux. Как мы увидим , благодаря его простоте AWK может сделать много полезного, даже когда код программы состоит всего из одной строки. В связи с этим многие думают, что AWK – это команда, тогда как это язык программирования.

Для структуризации текстовой информации можно использовать многие другие языки. Например, Гарсия и Норли (Garcia, Norli, 2012b); работая с данными SEC, находящимися в свободном доступе на https://www.sec.gov/edgar.shtml, используют Perl. Сложно сказать, какой из языков программирования наиболее популярен, поскольку, как уже упоминалось, большинство авторов такую информацию не разглашают, но можно с уверенностью заявить, что для пользователей с ограниченными знаниями программирования не существует лучшего выбора, чем AWK. А пользователей с обширными познаниями в области программирования AWK послужит полезным подспорьем в тех случаях, когда можно обойтись без написания длинных программ, выполняющих те же операции, что и лаконичные программы на языке AWK (Бухвалова, Никандрова, 2001; Бухвалова, Бухвалова, 2013).

Дабы не быть голословными, мы выбрали иллюстративный пример с простой структурой, использующий бесплатную и легко доступную информацию – рейтинги компаний, публикуемые газетой «*Financial Times*».

Когда газета «*Financial Times*» публикует свои очередные ежегодные рейтинги, которые коротко называют *FT* 500[1], ведущие финансовые издания публикуют анализ этих рейтингов. В качестве примеров таких публикаций сошлемся на несколько обзоров этих рейтингов в газете «Ведомости» (Пестон, 2000; Брискоу, 2003; 2005; Оверченко, 2014; 2015). Однако далеко не всегда в этих обзорах содержится вся информация. Например, пользователю может быть интересно сопоставить результаты последнего рейтинга с одним из рейтингов предыдущих лет. О том, как решить эти и подобные проблемы, и пойдет речь в настоящей статье.

Рейтинги *FT* 500 имеются в свободном доступе, чаще всего – в виде текстовых файлов. Поэтому естественным является желание использовать для анализа рейтингов какое-либо доступное программное обеспечение. Чтобы проиллюстрировать компактность написания на AWK традиционных программ-запросов, начнем с двух программ, которые должны произвести впечатление на тех, кто впервые знакомится с этим языком программирования.

Предположим, что у нас имеется текстовый файл со списком рейтинга *FT Global* 500 за 2015 г. Если мы хотим узнать, какие российские компании вошли в этот список, то программа на языке AWK, которая позволит получить ответ на этот вопрос, состоит всего из одной короткой строки:

```
$4==/Russia/{print $3}
```

Следующая программа, которая вычислит, сколько компаний из России вошло в этот список, состоит из двух строк:

```
$4==/Russia/{nr+=1}

END{print nr}
```

Не будем здесь комментировать эти программы (сделаем это позже, в разделе 4), отметим только, что компактность программ связана с тем, что AWK *специально предназначен* для решения задач по обработке структурированных текстов.

Многие языки программирования богаче языка AWK по средствам программирования, но они, без сомнения, уступают ему по *читаемости* кода и компактности интерпретаторов. Чтобы выполнить AWK-программу практически на любом компьютере, достаточно иметь с собой флешку с интерпретатором. Образно говоря, AWK – *это язык, который всегда с тобой*[2].

Уже много лет авторы этой статьи активно используют AWK для обработки различных видов информации: фактических данных для лекций по финансам и исследованию операций; данных для построения различных моделей, изучаемых в этих дисциплинах; данных для дальнейшего использования со специализированными средствами (например, MATLAB, SAS). С этими целями авторами разработана методика обработки информации, базирующаяся на использовании языка программирования AWK, изложение которой и является главной целью настоящей статьи.

Статья имеет следующую структуру. Раздел 2 посвящен описанию и обсуждению рейтингов, которые с 1997 г. составляет газета «*Financial Times*». Раздел 3 содержит краткий обзор языка AWK; описана структура программы и основные элементы языка. Информации, имеющейся в этом разделе, должно быть

[1] Газета «*Financial Times*» начиная с 1997 г. ежегодно составляет группу рейтингов *FT 500. FT Global* 500 – один из них.

[2] Сказанное верно, если использовать exe-файлы интерпретаторов. Пользователям Linux и UNIX отдельно загружать AWK не придется. Он входит в стандартные дистрибутивы этих систем.

достаточно для понимания программ, представленных в следующих разделах этой статьи. В разделе 4 приведены компактные программы-запросы на языке AWK. С помощью этих программ проведены анализ и обработка информации, содержащейся в рейтингах *FT Global* 500 за 2012 и 2015 гг. и *FT Europe* 500 за 2015 г. В ряде случаев для проведения анализа данных авторы рекомендуют создавать сначала *дополнительные рейтинги типа FT*. Процесс создания такого рейтинга на конкретном примере описан в разделе 5. Некоторые проблемы, которые могут возникнуть при подготовке файлов с данными, способы их решения обсуждаются в разделе 6. Там же даются рекомендации по хранению коллекции программ на языке AWK. В заключении (раздел 7) обсуждается, для каких видов данных применима предложенная в статье методика анализа.

2. Рейтинги газеты «*Financial Times*»

Английская деловая газета «*Financial Times*» (далее – *FT*)[1], основанная в 1888 г., публикует новости из области финансов и бизнеса со всего мира. *FT* является частью медиагруппы *FT Group*. На настоящий момент 100% акций этой медиагруппы принадлежат японской медиагруппе *Nikkei*. Газета издается на английском языке в 24 городах мира. В конце 2015 г. у газеты было 750 тыс. подписчиков (из них 550 тыс. подписчики электронной версии).

Начиная с 1997 г. *FT* ежегодно публикует группу рейтингов, которые получили название «рейтинги *FT* 500». Впервые на русском языке эти рейтинги были опубликованы (одновременно с английским оригиналом) в 2000 г. газетой «Ведомости» (Пестон, 2000). Оценка компаний в этих рейтингах проводится по их рыночной капитализации. Рыночная капитализация определяется ценой и количеством выпущенных акций на 31 марта года, для которого составлен рейтинг. Цены акций пересчитываются в единую валюту для возможности сравнения. Все суммы в главном *FT* 500 рейтинге (*FT Global* 500) приведены в долларах на эту дату.

При составлении рейтингов учитываются только те компании, у которых в свободном обращении не менее 15% акций. По капитализации отдельно составляется список 500 крупнейших компаний в мире (*Global* 500), Европе (*Europe* 500), Великобритании (*UK* 500), США (*USA* 500), Японии (*Japan* 500) и развивающихся странах (*Emerging* 500).

Каждый из рейтингов *FT* 500 является фактически таблицей, у которой число строк равно числу компаний (500), а число столбцов – числу параметров компании. Число, порядок следования параметров и тип файла, в котором эта таблица записана, менялись за годы существования рейтинга. В последние годы таблицы доступны в виде xls-файлов, но были и pdf-файлы (2012), и csv-файлы[2]. В 2015 г. всего было указано 14 параметров для каждой компании. В табл. 1 приведен список названий этих параметров (столбцов таблиц с данными). Заметим, что порядок следования столбцов в другие годы мог быть несколько иным. Например, в 2012 г. столбцы 5 (Market value $m) и 6 (Sector) шли в обратном порядке.

Принадлежность компании к той или иной стране определяется тем, где преимущественно размещены ее акции. Например, одна из крупнейших авиастроительных

[1] Официальный сайт газеты: http: //www.ft.com.

[2] Для программы Excel csv-формат является родным, она открывает и сохраняет файлы в этом формате.

Таблица 1

Имена и последовательность столбцов в рейтингах *FT* 500 за 2015 г.

Column	Name	Column	Name
1	Global/Europe rank 2015	8	Net Income $m
2	Global/Europe rank 2014	9	Total assets $m
3	Company	10	Employees
4	Country	11	Price $
5	Market value $m	12	P/E ratio
6	Sector	13	Dividend yield (%)
7	Turnover $m	14	Year End

компаний в мире *Airbus* считается французской, так как основная часть ее акций обращается на парижской фондовой бирже *Euronext*. Отнесение компании к определенной стране выполняется на основании рекомендаций, ранее составленных для *FTSE* 100 *Index*[1]. Однако для некоторых компаний в рейтингах указываются две страны. Например, в *FT Global* 500 2015 имеется 5 таких компаний: *BHP Billiton* (*Australia/UK*), *Unilever* (*Netherlands/UK*), *Rio Tinto* (*Australia/UK*), *Carnival* (*US/UK*), *Reed Elsevier* (*Netherlands/UK*). Некоторые объяснения на эту тему приведены на сайте *FT* в разделе «*FT* 500 2015 *Introduction and methodology*».

У рейтингов *FT* 500 имеется немало критиков. Большинство из них отмечают неадекватное, по их мнению, отображение этими рейтингами мировой экономики и финансов и связывают это с тем, что в рейтинги не входят крупные государственные и частные компании. Ответы на большинство замечаний критиков можно найти в публикациях (Брискоу, 2003; 2005), автором которых является бывший редактор отдела статистики *FT* Саймон Брискоу (Simon Briscoe). История показывает, что отчетность большинства крупных государственных и частных компаний рано или поздно становится публичной. В качестве примера приведем относительно новую информацию о планах саудовской нефтегазовой госкомпании *Aramco*. Эта компания, по мнению многих аналитиков, считается крупнейшей в мире по добыче нефти и стоимости нефтяных запасов. Предположительная стоимость компании составляет 10 трлн долл. США (Omran, Said, 2016). В начале 2016 г. компания объявила о возможной приватизации части компании через процедуру *IPO*. Ряд экспертов считает, что это позволило бы ей стать крупнейшей публичной компанией в мире по капитализации, обогнав американскую *Apple*.

Список из 39 секторов (упорядоченный в алфавитном порядке), на которые разбиты компании по роду своей основной деятельности в рейтингах *FT* 500 в последние годы, приведен в первом столбце табл. 2. Во втором, третьем и четвертом столбцах этой таблицы содержится информация о количестве компаний в каждом секторе для рейтингов *FT Global* 500 2012, *FT Global* 500 2015 и *FT Europe* 500 2015 соответственно. Вся информация для этой и всех следующих таблиц в этой статье была получена как результат выполнения программ-запросов на языке AWK. Программа, с помощью которой получена информация для табл. 2, приведена в разделе 4.

[1] *FTSE* 100 *Index* – основной индекс Лондонской фондовой биржи. Рассчитывается с 30 декабря 1983 г. и ведет свой отчет от 1000.

Таблица 2

Секторы и распределение компаний по ним в 2012 и 2015 гг.

Секторы FT 500	Global 2012	Global 2015	Europe 2015
Aerospace & defence	7	10	13
Alternative energy	0	1	1
Automobiles & parts	17	17	13
Banks	68	71	52
Beverages	12	10	9
Chemicals	16	16	25
Construction & materials	3	7	17
Electricity	14	11	13
Electronic & electrical equipment	4	7	5
Financial services	16	28	23
Fixed line telecommunications	15	11	11
Food & drug retailers	5	8	15
Food producers	9	8	12
Forestry & paper	0	0	3
Gas, water & multiutilities	9	5	13
General industrials	12	11	8
General retailers	17	16	12
Health care equipment & services	13	14	8
Household goods & home construction	3	2	8
Industrial engineering	12	13	18
Industrial metals & mining	12	1	9
Industrial transportation	8	10	14
Leisure goods	4	3	0
Life insurance	12	10	13
Media	13	16	21
Mining	13	5	10
Mobile telecommunications	16	15	10
Nonlife insurance	12	15	22
Oil & gas producers	43	31	22
Oil equipment & services	10	7	4
Personal goods	11	13	15
Pharmaceuticals & biotechnology	21	30	17
Real estate investment & services	3	6	4
Real estate investment trusts	6	8	10
Software & computer services	13	16	8
Support services	4	5	21
Technology hardware & equipment	16	19	8
Tobacco	8	7	3
Travel & leisure	10	17	13

3. Язык программирования AWK

Первоначально язык программирования AWK возник и был реализован в операционной системе UNIX. В этой системе имеются специальные программы — *программируемые фильтры*, которые читают входной поток (файлы-аргументы или стандартный входной поток), выполняют простые операции над ним и записывают результат в выходной поток. Имя программы составлено из первых букв фамилий ее создателей: А. Ахо (Alfred Aho), П. Вайнбергер (Peter Weinberger), Б. Керниган (Brian Kernighan). Выполнение программы на языке AWK заключается в последовательном просмотре входных файлов с данными с целью поиска строк, которые соответствуют одному из образцов, заданных в программе. Язык, на котором программируется поведение программы AWK, больше всего похож на язык программирования С: имеются аналогичные по форме записи условные операторы, циклы, переменные и функции пользователя. Однако набор средств, необходимых для записи базовых операций поиска и обработки текстовых файлов, столь невелик, что незнание языка программирования С не будет препятствием в овладении навыками использования языка AWK.

3.1. Интерпретаторы

Язык для команды AWK был по достоинству оценен программистами, и в 90-е гг. прошлого века появились интерпретаторы этого языка – программы, которые анализируют и тут же выполняют программу на каком-то языке программирования. В настоящее время наиболее популярны следующие версии языка AWK и интерпретаторы для них: GNUAWK (gawk) и One True Awk (nawk, awk 95). Все эти интерпретаторы имеются в свободном доступе в Интернете. Для выполнения программ для этой статьи мы использовали сначала интерпретатор awk 95 (написан в 2000 г. самим Б. Керниганом), а затем проверяли совпадение полученных результатов с результатами при использовании интерпретаторов, входящих в пакет Gnu Win 32, v. 3.1.6-1.

В системе *Windows* запуск любого интерпретатора осуществляется командой

<интерпретатор> -f awkfile data1file data2file . . .

Текст программы читается из файла awkfile, а данные – последовательно из файлов data1file, data2file и т. д. Результат работы программы выводится в выходной поток. Можно переадресовать выходной поток в файл resfile обычным способом, например

gawk -f awkfile datafile>resfile

До начала сканирования (анализа) первого файла data1file интерпретатор проверяет программу (файл awkfile) на наличие в ней синтаксических ошибок. Если обнаруживается ошибка, то выдается диагностическое сообщение.

3.2. Литература о языке AWK

Те, кто только начинает программировать на AWK, смогут найти много полезных примеров в статьях (Бухвалова, Бухвалова, 2013; Бухвалова, Никандрова, 2001; Романовский, 2013). Грамотному программированию на языке AWK учит книга (Robbins, 2015). Полное описание последней версии языка AWK из проекта GNU приведено в (Robbins, 2016). Оба этих руководства написаны Арнольдом Роббинсом – известным программистом и автором публикаций по технологии программирования на языке AWK. Следует посоветовать также состоящее из трех частей практическое руководство

по программированию на этом языке его однофамильца и тоже известного программиста Даниэля Роббинса (Robbins, 2000; 2001a; 2001b). Тем, кому захочется больше узнать о языке AWK и его связи с другими языками программирования, советуем использовать книгу (Бьянкуцци, Уорден, 2011).

3.3. Структура программы и ее элементы

Программа на языке AWK представляет собой набор строк вида

```
BEGIN {начальные действия}
шаблон    {действия}
.................
шаблон    {действия}
END {конечные действия}
```

После того как интерпретатор языка AWK проверит программу на отсутствие синтаксических ошибок и до чтения каких-либо данных из файлов, выполняются начальные действия, приписанные к шаблону BEGIN. Далее последовательно читаются все строки всех файлов с данными (если только где-либо в действиях не предусмотрено изменение или прекращение этого процесса). В каждый момент доступна только одна строка – текущая. В текущей строке ищутся *шаблоны* в том порядке, в котором они перечислены в программе. Если шаблон в строке обнаружен (*шаблон соответствует строке*), то со строкой выполняются действия, которые соответствуют этому шаблону. После чтения и обработки всех строк выполняются конечные действия, приписанные к шаблону END. Шаблоны BEGIN и/или END могут быть опущены. Допускается опускать одну из частей конструкции шаблон – действия. Действия без шаблона выполняется с каждой строкой входного потока. Если не указаны действия, то строки, соответствующие шаблону, копируются в выходной поток.

Комментарий может быть вставлен в конце любой строки программы. Он начинается с символа # и заканчивается в конце строки.

Поля. Каждая входная строка файла с данными автоматически разбивается на поля. По умолчанию поля – это последовательности символов без пробелов, разделенные пробелами и/или символами табуляции. Разделитель полей контролируется встроенной переменной FS. Присваивание в шаблоне BEGIN переменной FS любого символа, отличного от пробела, делает этот символ разделителем полей.

Входные строки читаются последовательно по одной. Значение текущей строки присваивается встроенной переменной $0. Поля текущей строки от начала к концу присваиваются встроенным переменным $1, $2,... $NF, где NF – *встроенная* переменная, значение которой устанавливается равным числу полей в текущей строке.

Типы данных. В языке AWK есть только два базовых типа данных: числа и строки символов. Способы записи числовых констант и набор операций с числами такие же, как в популярных языках программирования. Имеется еще *строково-числовой* тип, данные которого имеют одновременно и числовые, и строковые значения. Какой именно тип использовать при выполнении операции, определяется контекстом. Переменные, не являющиеся встроенными, определяются самим фактом их использования. По умолчанию они инициируются строково-числовым значением null, числовым значением 0 и строковым значением "" (пустая строка). Преобразование типов, когда это нужно, выполняется автоматически.

Шаблоны. В рассмотренных в этой статье программах шаблоны будут использоваться для поиска строк, удовлетворяющих или не удовлетворяющих какому-либо критерию. Это наиболее типичное их использование в программах на

языке AWK.Если шаблоны используются только для простой проверки данных, то отсутствие выходного потока в таких программах свидетельствует о том, что все данные удовлетворяют заданным шаблонами критериям.

Ассоциативные массивы. В большинстве популярных языков программирования индекс массива – это целое число, а в языке AWK в качестве индекса можно использовать любое значение (строку). Имеется специальная форма цикла for для перебора всех индексов ассоциативного массива. Такой цикл устанавливает значение переменной индекса равным каждому значению индекса поочередно. При этом порядок перебора индексов непредсказуем, поэтому может возникнуть необходимость в дополнительной сортировке.

Встроенные функции. Как и в других языках программирования, в языке AWK имеются встроенные функции, предназначенные прежде всего для работы со строками и файлами. Далее в программах будут примеры применения некоторых из них.

Приведенной в этом разделе информации должно хватить для понимания программ на языке AWK из этой статьи. Полную информацию об этом языке можно найти в источниках из списка в конце статьи (см. п. 3.2).

4. Анализируем рейтинги *FT* 500

В этом разделе показано, как с помощью языка AWK можно получать ответы на вопросы, которые часто возникают при анализе финансовой информации. В качестве источников информации мы выбрали три текстовых файла: рейтинги *FT Global* 500 за 2012 и 2015 гг. и *FT Europe* 500 за 2015 г. Будем считать, что информация в этих файлах уже представлена в удобном для работы с языком AWK виде: 500 строк, в которых содержится информация о компаниях; каждая строка состоит из 14 полей, которые разделены символом табуляции (\t); пустой графе в таблице соответствует поле *пустая строка* (""). На самом деле часто приходится сначала преобразовывать файл с данными, чтобы привести его к такому виду. То, как выполняются подобные преобразования, обсуждается в разделе 6.

Интерпретатор использует в качестве разделителя полей значение встроенной переменной FS. Чтобы все программы из этого раздела работали корректно, необходимо в начале каждой программы в разделе BEGIN указать значение разделителя полей FS:

```
BEGIN {FS = "\t"}
```

Анализ информации организован в форме «вопрос-ответ», где ответом будет текстовый файл, созданный в результате выполнения работы программы на языке AWK. Конец обсуждения ответа на вопрос будет отмечаться знаком ■. Так как в большинстве примеров речь идет о списке *FT Global* 500 в 2015 г., далее будем ссылаться на него как на *список*. Начнем с вопроса из введения к этой статье.

Вопрос 1. *Какие компании из России вошли в список и каковы у них рейтинги?*

Ответ. Российским компаниям соответствуют строки, у которых в поле 4 (Country) написано Russia. Названия компаний записаны в поле 3 (Company), а их рейтинги (места в порядке возрастания) в поле 1 (Global rank 2015). Поэтому ответ на поставленный вопрос будет получен после выполнения следующей программы на AWK:

```
$4~/Russia/{print $1 ":" $3}
```

Если мы уверены, что строки, содержащие шаблон /Russia/, относятся только к российским компаниям, то возможен сокращенный вариант записи:

```
/Russia/ {print $1 ":" $3}
```

Результат выполнения этой программы – файл, состоящий из 5 строк (рис. 1).

```
170:  Gazprom
213:  Rosneft
271:  Lukoil
421:  Mmc Norilsk Nickel
441:  Surgutneftegas
```

Рис. 1. Ответ на вопрос 1: Российские компании в *FT Global* 2015

Выполнение этой же программы с рейтингом *FT Global* 500 за 2012 г. напомнит, что три года назад в список входило 10 российских компаний, а места оставшихся в списке 2015 г. компаний стали много хуже (рис. 2).

```
31:  Gazprom
79:  Rosneft
86:  Sberbank of Russia
132:  Lukoil
180:  Novatek
182:  Surgutneftegas
222:  Norilsk Nickel
343:  Gazprom Neft
366:  VTB Bank
374:  Uralkali
```

Рис. 2. Ответ на вопрос 1: Российские компании в *FT Global* 2012

Если теперь выполнить программу с рейтингом *FT Europe* 500 за 2015 г., то видим, что к ранее приведенному списку добавятся еще 12 российских компаний (рис. 3). ■

```
44:  Gazprom
60:  Rosneft
78:  Lukoil
106:  Mmc Norilsk Nickel
111:  Surgutneftegas
135:  Sberbank of Russia
140:  Novatek
159:  Magnit
217:  VTB Bank
265:  Tatneft
281:  Megafon
286:  Severstal
300:  Alrosa
307:  Mobile Telesystems
331:  Novolipetsk Steel
336:  Uralkali
373:  Bashneft
```

Рис. 3. Ответ на вопрос 1: Российские компании в *FT Europe* 2012

Представляет интерес динамика изменения рейтинга российских компаний. Поэтому получим ответ на следующий вопрос.

Вопрос 2. *Какие компании из России являются новичками в списке?*

Ответ. Новички – это компании, которые не входили в список в 2014 г., поле 2 (Global/Europe rank 2014) в этом случае является пустой строкой:

```
($4~/Russia/)&&($2=="") {print $1":"$3}
```

Обратите внимание на то, что при проверке равенства знак равенства пишется два раза: ==, а логическое и обозначается &&. После выполнения этой программы мы установим, что в *FT Global* 500 в 2015 г. не появилось ни одной новой российской компании (отсутствует выходной поток). Выполним теперь программу с рейтингом *FT Europe* 500 за 2015 г., и получим такой же результат – ни одной новой российской компании. То, что так было не всегда, можно установить, выполнив программу с *FT Global* 500 за 2012 г. Новичком в том году стала компания «Уралкалий», которая заняла в рейтинге 374-е место. ■

Для оценки информации в целом определим, компании каких стран вошли в рейтинг и сколько компаний этих стран в списке.

Вопрос 3. *Какие страны вошли в списки и сколько их компаний в списке?*

Ответ. Воспользуемся для ответа на этот вопрос *ассоциативным массивом*, которому присвоим имя cmp. Индексами в этом массиве будут сроки – названия стран. Значения элементов массива – количество компаний в списке из страны-индекса. Для перебора всех значений индекса (в программе он обозначен с) ассоциативного массива (к сожалению, в неизвестном порядке) существует специальная форма оператора цикла for. Текст программы, которая выводит необходимую информацию, гораздо короче текста этих пояснений:

```
{cmp[$4]+=1}

# в строке очередная компания из страны $4[1]

END{for(c in cmp) print c ":" cmp[c]}
```

После выполнения этой программы получаем файл (выходной поток) из 35 строк. Однако стран в список вошло только 32. Объясним это расхождение. Как мы уже отмечали ранее (см. раздел 2), 5 компаний из *FT Global* 500 2015 отнесены к двум странам: *BHP Billiton* (*Australia/UK*), *Unilever* (*Netherlands/UK*), *Rio Tinto* (*Australia/UK*), *Carnival* (*US/UK*), *Reed Elsevier* (*Netherlands/UK*). Поэтому в файле имеется три *лишние* строки: *Australia/UK: 2, Netherlands/UK: 2, US/UK: 1.* Как воспринимать эти строки (удалять с корректировкой числа компаний из соответствующих стран или оставлять как пример сотрудничества), должно решать лицо, проводящее анализ.

У файла, который был получен при ответе на вопрос 3, есть еще один *недостаток*: названия стран в нем перечислены в случайном порядке (свойство оператора for (... in ...)). Однако это можно исправить, использовав встроенную функцию сортировки asorti, которая отсортирует индексы массива cmp и запишет их в массив cnt (n – размер этого массива):

```
{cmp[$4]++}
END{n=asorti(cmp,cnt);
for (i=1; i<=n; i++) print cnt[i]":" cmp[cnt[i]]}
```

На рис. 4 приведены из экономии места только первые пять строк файла, который будет получен после выполнения этой программы, чтобы показать, что список стран выводится в алфавитном порядке.

[1] В операторе cmp[$4]+ =1 использована сокращенная форма записи для cmp[$4] = cmp[$4]+1. Далее будет использовано другое возможное в этом случае сокращение – cmp[$4]++.

Мы использовали полный аналог этой программы для составления табл. 2 (заменили в тексте программы $4 на $6), которая содержит отсортированный список секторов экономики и количества компаний из этих секторов. ■

```
Australia: 8
Australia/UK: 2
Belgium: 2
Brazil: 6
Canada: 19
```

Рис. 4. Фрагмент ответа на вопрос 3: Список стран в FT Global 2015

Программа-ответ на следующий вопрос не только позволяет получить полезную информацию, но и демонстрирует использование оператора **next** и логической операции **или (**обозначается ||).

Вопрос 4. *Какая компания максимально улучшила свой рейтинг в 2015 г.?*

Ответ. Прокомментируем приведенную ниже программу. Во-первых, просматривая построчно файл со списком, будем пропускать строки, относящиеся к компаниям, которые не входили в рейтинг в 2014 г. или которые не улучшили свой рейтинг в 2015 г. Оператор **next** обеспечивает переход в этом случае к анализу следующей строки в файле с данными. Среди оставшихся строк выбираем строку с максимальным ростом – значением разности $2-$1:

```
($2==" ")||($2 <= $1){next} # ненужные строки
($2-$1)>max{max=$2-$1;com=$3;cntr=$4;sec=$5}
END{print "Максимальный рост:"com" ("cntr", "sec")"}
```

Выполнив эту программу, узнаем, что максимальный рост рейтинга в 2015 г. был у китайской автомобильной компании «*Saic Motor*», которая поднялась с 456-го на 224-е место:

Максимальный рост: *Saic Motor* (*China*, *Automobiles & parts*). ■

Финансистов, несомненно, интересует положение банков в рейтинге, поэтому ответим на три вопроса о них.

Вопрос 5. *Сколько банков в списке?*

Ответ. С точки зрения программирования программа для ответа на этот вопрос мало чем отличается от программы для вопроса 1. Добавим вторую строку с оператором печати и вычислим долю банков в процентах:

```
/Banks/ {nb++}
END{print "Число банков в списке: " nb "(" nb/5 "%)"}
```

Результатом выполнения этой программы будет строка:

Число банков в списке: 71 (14.2%)

Результат выполнения этой программы для *FT Europe* 500 за 2015 г. показывает, что в Европе крупных банков существенно меньше:

Число банков в списке: 52 (10.4%) ■

Вопрос 6. *Какова доля капитализации банков в общей капитализации компаний из списка?*

Ответ. Программа для ответа на этот вопрос немного длиннее. Напомним, что капитализация компаний (в млн долл.) записана в $5. Для того чтобы сделать сумму общей капитализации банков более наглядной, разделим ее на 1000 и укажем, что сумма указана в млрд долл.:

```
# cm – капитализация всех компаний из списка
# bm – капитализация банков
{cm+=$5}
/Banks/{bm+=$5}
END{print "Капитализация банков: $" bm/1000 "b (" bm/cm*100 "%)"}
```

Результат выполнения этой программы для *FT Global* 500 за 2015 г. показывает, что средняя капитализация банков немного больше средней капитализации компаний, входящих в рейтинг:

Капитализация банков: $4884.56b (15.08%)

Для списка *FT Europe* 500 за 2015 г. сумма общей капитализации банков меньше примерно в три раза этого показателя для *FT Global* 500:

Капитализация банков: $1544.52b (13.48%) ■

Определим теперь, какой банк занимает самую высокую позицию в рейтинге. В программе для ответа на этот вопрос используется оператор **exit**. Его используют, когда необходимая информация найдена и следует прервать просмотр оставшихся строк файла с данными.

Вопрос 7. *Какой банк из списка имеет самый высокий рейтинг?*

Ответ. Так как компании в списке перечислены в порядке убывания рейтинга, необходимо определить первый встретившийся в списке банк. После этого просмотр файла с данными следует остановить и вывести полученную информацию:

```
$6~/Banks/{com=$3;cntr=$4;r=$1;exit}
END{print"Лучшийбанк(2015):"com"("cntr",rank="r")"}
```

Теперь мы точно знаем, что крупнейший банк в 2015 г. – это американский банк «*Well Fargo*», который в 2015 г. занял седьмую строчку в рейтинге:

Лучший банк: *Wells Fargo* (*US*, 7)

Если выполнить эту программу для *FT Global* 500 в 2012 г., то увидим, что тогда крупнейшим был китайский банк «*Industrial & Commercial Bank of China*» (шестая позиция), который теперь занимает девятую позицию в списке. ■

В заключение приведем два вопроса, на которые хотелось бы иметь ответы, отличные от тех, что были получены.

Вопрос 8. *Какие компании из стран Балтии вошли в FT Europe 500 2015 и каковы у них рейтинги?*

Ответ. Компаниям из стран Балтии соответствуют строки, у которых в поле 4 (Country) написано Latvia, Lithuania или Estonia. Берем программу из вопроса 1 и вносим в нее небольшое изменение:

```
/Latvia/||/Lithuania/||/Estonia/ {print $1 ": " $3}
```

В результате выполнения этой программы мы получим файл длины 0 (пустой!). Грустный результат. ■

Просматривая исходные таблицы, мы обнаружили, что, несмотря на всемирную борьбу с курением, в списке имеются табачные компании.

Вопрос 9. *Сколько табачных компаний имеется в списке?*

Ответ. Для ответа на этот вопрос достаточно знать, что табачным компаниям соответствует сектор Tobacco:

```
/Tobacco/{nt+=1}
END{print"Число табачных компаний:"nt}.
```

Выполнив программу, мы узнаем, что в списке 7 табачных компаний:

Число табачных компаний: 7

Так как нам было интересно, какова динамика, мы проанализировали список в 2011 г. (7 компаний) и 2012 г. (8 компаний) и увидели удивительную стабильность, несмотря на борьбу с курением во всем мире! ■

Для более обстоятельного анализа часто приходится одновременно использовать информацию, которая хранится в нескольких файлах. Чтобы показать, что и в этих случаях программы на языке AWK обычно занимают несколько строк, ответим на следующий вопрос.

Вопрос 10. *Сколько компаний из FT Europe 500 не вошли FT Global 500* в *2015 г.?*

Ответ. Для ответа на этот вопрос нам понадобятся два файла с данными, которые имеют равное число строк (500). Чтобы различать, из какого файла текущая строка, будем использовать встроенную переменную NR – порядковый номер текущей строки. Если NR<=500, то текущая строка из *FT Europe* 500 (этот файл должен быть указан первым в командной строке), при NR>500 – из *FT Global* 500. При просмотре первого файла с данными составим список компаний с их рейтингами, а при просмотре второго файла – исключим из него компании, которые вошли и во второй список. Попытка удаления элемента массива с несуществующим индексом в языке AWK не будет ошибкой при исполнении программы – это еще один пример толерантности интерпретаторов AWK:

```
NR<=500{rank[$3]=$1}#FT Europe 500
NR >500{delete rank[$3]} # FT Global 500
END{for(com in rank)nc++;
print"Не вошли (2015): " nc}
```

Нет смысла приводить список этих компаний – это 372 компании из *FT Europe* 500, занимающие места с 129-го по 500-е. Границу отсечения в списке *FT Europe* 500 можно определить и по-другому: в список *FT Global* 500 из списка *FT Europe* 500 попали компании, капитализация которых не меньше капитализации последней компании в этом списке *FT Global* 500. Однако приведенная программа является универсальным методом исключения пересечения списков, так как она не предполагает их упорядочивания. ■

Предположим теперь, что требуется провести обстоятельный сравнительный анализ двух или более стран (групп стран). Разумеется, можно для каждой операции сравнения писать свою программу, которая сначала будет осуществлять поиск необходимой информации, а затем выполнять с ней соответствующую операцию сравнения. Однако организация процесса анализа станет много компактней и наглядней, если создать сначала *новый рейтинг типа FT 500*. Более того, при таком подходе гораздо легче расширять список операций сравнения, добавляя небольшую программу, содержащую несколько строк.

При создании нового рейтинга длина программ на языке AWK и их сложность существенно увеличиваются, поэтому те, кто предполагает ограничиться описанным выше анализом, могут пропустить следующий раздел.

5. Составляем новый рейтинг

В этом разделе в качестве примера описан процесс создания нового рейтинга *US-Japan* 500. В данном случае исходные данные – это xls-файлы *FT Japan* 500 и *FT US* 500 за 2015 г. Сразу отметим, что соответствующие им таблицы имеют не 14 (как у ранее

рассматриваемых файлов с рейтингами), а 13 столбцов: отсутствует столбец, в котором указывается страна. В том новом рейтинге *US-Japan* 500 будет 14 столбцов, порядок которых будет совпадать с порядком столбцов в основных рейтингах *FT*. Поэтому для нового рейтинга будут корректно работать все программы из раздела 4.

Процесс создания нового рейтинга состоит из трех этапов. На *первом* этапе открываем в программе Excel имеющиеся xls-файлы и сохраняем их в *текстовом формате с разделителями табуляции*. Для этого выбираем этот тип файла при выполнении команды **Сохранить как** из вкладки **Файл**). Чтобы программы на языке AWK, которые будут выполнять дальнейшие преобразования файлов, были короче, советуем сразу удалить из полученных файлов начальные и конечные строки, которые содержат поясняющую информацию. Сделать это можно вручную (используем клавишу delete) или с помощью следующей программы, которая создает копию 500 строк, содержащих информацию о компаниях (FNR – встроенная переменная, номер текущей строки в текущем файле):

```
BEGIN{FS = "\t"; OFS = "\t"; nlt = 4; nall=505; }
FNR<=nlt{ next}
{print $0; next}
FNR==nall{exit}
```

Обращаем внимание на то, что в этой программе предусмотрено сохранение пятой строки с заголовками столбцов в рейтинге *FT Japan* 500 и учтено, что в рейтинге *FT US* 500 заголовок занимает только четыре строки.

На *втором* этапе выполняем программу на языке AWK, которая, во-первых, убирает пробелы, разделяющие группы разрядов при записи чисел в столбцах 4, 6–10 таблиц, а во-вторых, вставляет последний столбец с названием страны. Для выполнения этого преобразования используем встроенную функцию gsub, которая имеет три аргумента: что заменяем, чем заменяем, где заменяем. Для рейтинга *FT US* 500 эта программа имеет вид:

```
{gsub(" ","",$4); gsub(" ","",$6); gsub(" ","",$7); gsub(" ","",$8);
gsub(" ","",$9);gsub(" ","",$10); $14="US"; print $0; }
```

Для рейтинга *FT Japan* 500 в первой строке с названиями столбцов не нужно убирать пробелы и следует добавить заголовок последнего столбца (Country):

```
NR==1{ $14="Country "; print $0; next;}
{gsub(" ","",$4); gsub(" ","",$6); gsub(" ","",$7); gsub(" ","",$8);
Gsub(" ","",$9);gsub(" ","",$10); $14="Japan"; print $0; }
```

Можно выполнить только последнюю программу, применив ее к списку из двух файлов. Только тогда файл с *FT Japan* 500 должен быть первым.

На *третьем* этапе выполняем программу, которая выбирает 500 крупнейших по капитализации компаний из 1000 компаний, имеющихся в двух рейтингах. Алгоритм, который мы применим для этого, принято называть *сортировкой слиянием* (Ахо, Хопкрофт, Ульман, 2003). Суть этой сортировки состоит в том, что попарно сравниваются элементы из двух отсортированных массивов. В массив-результат помещается тот элемент, который лучше в смысле используемого критерия. В нашем случае это компания, у которой капитализация больше. После этого выбывший элемент заменяется следующим из соответствующего массива. В общем случае процесс сравнения продолжается, пока один из массивов не иссякнет. В нашем случае условием окончания процесса является условие nl=lrank : отобраны 500 лучших компаний. Заметим, что приведенная далее программа является универсальной

в том смысле, что при замене значения переменной lrank на любое значение, не превосходящее 1000, будет образовываться рейтинг соответствующей длины. Например, при lrank=100 в рейтинге будут 100 крупнейших компаний из США и Японии.

```
BEGIN{FS = "\t"; nall=501; iw =1; jr=1; nl=0; lrank=500;}
# копируем строку с названиями столбцов
NR==1 {print $0; next}
# копируем информацию о Японии в массивы:
# list – строки таблицы, mrkt – капитализация компаний
NR<=nall{list[iw]=$0; mrkt[iw]= $4; iw++; next}
# попарное сравнение компаний USи Japan
# пока US хуже Japan ==> вставляем строку Japan
{while($4<mrkt[jr]){print list[jr]; jr++;
nl++; if(nl>lrank) exit;}
# US лучше Japan ==> вставляем строку US print $0; nl++; if(nl>lrank)
exit; }#
```

На *четвертом* этапе завершаем создание нового рейтинга. Во-первых, добавляем в него стандартный заголовок (4 строки) и стандартные последние строки (2 строки). Пока этот рейтинг новый, у него отсутствует значение в 2014 г. ($2=""). Но больше всего места в программе занимает вставка столбца 14 на место столбца 4 и сдвиг столбцов с 4-го по 13-й в следующий столбец:

```
BEGIN{FS = "\t"; nall=501; rank=1;}
# формируем стандартный заголовок рейтинга
BEGIN {print "US-Japan 500 2015";
print "Market values and prices at 31 March 2015\n\n"}
# первая строка таблицы с названиями столбцов
NR==1{$1="US-Japan rank 2015"; $2="US-Japan rank 2014";cnt=$14;
for (i=14; i>=5; i--) {$i=$(i-1)}; $4=cnt; print $0; next}
# строки с информацией о компании
{$1=rank; rank++; $2=""; cnt=$14;
for (i=14; i>=5; i--) {$i=$(i-1)};
$4=cnt; print $0}
NR==nall{exit}
 # заключительные строки рейтинга
END{print "\nData from Thomson ONE Banker, Thomson Reuters
Data stream and individual companies."}
```

Теперь мы можем анализировать новый рейтинг *US-Japan* 500 так, как это описано в разделе 4. Начнем с выявления информации о компаниях из Японии.

Вопрос 11. *Сколько компаний из Японии вошли в рейтинг US-Japan 500 и какова доля их капитализации в общей капитализации компаний из этого списка?*

Ответ. Напомним, что и в новом рейтинге капитализация компаний (в млн долл.) записана в $5.

```
BEGIN{FS = "\t"; nlt=5; nall=505; }
NR<=nlt {next}
{cm+=$5}
/Japan/{nj++; jm+=$5}
NR==nall{exit}
END{print "ЧислокомпанийизЯпонии: " nj " (" nj/5 "%)"
print "Доля их капитализации: " jm/cm*100"%"}
```

Результат работы этой программы подтверждает феномен послевоенного развития экономики Японии:

Число компаний из Японии: 96 (19.20%)
Доля их капитализации: 12.31%

Более скромный результат в доле капитализации объясняется тем, что лучшая японская компания *Toyota Motors* (официальное название *Toyota Motor Corporation*) занимает в нашем рейтинге *US-Japan* 500 позицию 10. Однако следует иметь в виду, что в рейтинге *FT Global* 500 в 2015 г. эта компания улучшила свое положение на 8 позиций, перейдя с 23-й (2014 г.) на 15-ю позицию. ■

В заключение этого раздела отметим, что для анализа рейтинга *US-Japan* 500 годятся все программы из раздела 4, если применять их к копии этого рейтинга, содержащей только строки с информацией о компаниях.

6. О предварительной обработке файлов с данными

При подготовке файлов с данными для программ на языке AWK практически всегда приходится проводить некоторую предварительную обработку исходных файлов[1]. В этом разделе рассмотрены типичные преобразования, которые приходится при этом выполнять.

Как уже отмечалось выше, язык AWK предназначен прежде всего для обработки структурированных текстовых файлов. При этом предполагается, что *текстовый файл* – это файл, содержащий информацию в виде последовательности текстовых символов, разделенных символами конца строки.

Самый распространенный текстовый формат, предназначенный для представления табличных данных, называется *CSV* (*Comma Separated Values*). Например, в таком виде доступна большая часть информации, распространяемой Организацией экономического сотрудничества и развития (ОЭСР). В формате *CSV* каждой строке таблицы соответствует строка файла. Значения отдельных граф таблицы разделяются специальным символом. Раньше в качестве этого символа использовалась только запятая, но в настоящее время стандарт *CSV* допускает использование и других символов в качестве разделителя. В частности, если запятая используется в таблице, то в качестве разделителя в файле часто используется точка с запятой (;) или знак табуляции (\t).

Приведем два примера предварительного преобразования файлов, опираясь на нашу практику. В первом примере речь идет о нетривиальном случае, в котором, возможно, рядовому пользователю компьютера придется обратиться за помощью к профессионалу.

В 2013 г. на сайте *www.ft.com* свободный доступ к рейтингам предоставлялся в виде pdf-файлов. Например, pdf-файл с *FT Global* 500 (2012 г.) состоял из 14 страниц мелкого шрифта. Каждая из этих страниц – таблица, в которой 14 граф с заголовками. Количество строк в этих таблицах менялось от страницы к странице.

Не вдаваясь в детали выполненного преобразования данных в этом случае, сообщим только, что проведено оно было в два этапа. На первом этапе была использована программа ABBYY Fine Reader, которая перевела pdf-файл в *csv*-формат. На втором этапе с помощью языка AWK было выполнено дополнительное преобразование полученного csv-файла. Подробно это описано в (Бухвалова, 2013).

[1] Разумеется, все изменения следует проводить с копией имеющихся данных.

Подготовка данных с рейтингами в 2015 г., о которых речь идет в этой статье, оказалась много проще. В этом году *FT* предоставила свободный доступ к своим рейтингам в формате xls-файлов. Однако и в этом случае подготовка данных проходила в два этапа, которые уже были описаны в разделе 5.

При подготовке данных может потребоваться и визуальный просмотр исходного файла. Например, в 2016 г. нами было обнаружено, что заголовок в файле с *FT US* 500 (2015 г.) содержит только четыре строки, в то время как заголовки остальных файлов содержали пять строк. В случае малого числа файлов подобные ошибки проще всего исправить, вставив руками недостающую пустую строку. Если файлов для обработки много, код программы можно сделать более гибким, обрабатывая только строки, содержащие правильное число полей (NF).

Опишем теперь удобный способ хранения коллекции программ на языке AWK, относящихся к обработке одних и тех же данных. Из многочисленных примеров в разделе 4 видно, что типичная программа-запрос на языке AWK содержит не более пяти строк, но программ таких может быть довольно много. Если хранить каждую программу в отдельном файле, то придется придумывать правило, по которому можно будет по имени файла определять, какой программе он соответствует[1].

Гораздо удобнее хранить все программы в *одном* файле, разделяя их строками комментариев. Прежде всего, такой список программ будет удобно редактировать (добавлять новые программы, корректировать и удалять старые).

Файл с программами должен начинаться с шаблона BEGIN, который *относится ко всем программам из файла*. Напомним, что для программ из раздела 4 в нем задается символ, который разделяет поля входных строк:

```
BEGIN {FS = "\t"}
```

В каждую программу можно включить ее собственный, дополнительный шаблон BEGIN, в котором задаются значения встроенных и локальных переменных. Заметим, что в программах из раздела 4 таких шаблонов не понадобилось.

Для получения ответа на конкретный вопрос все строки программ, кроме тех, которые соответствуют программе-ответу на этот вопрос, надо сделать комментариями.

В принципе возможно получение одновременного ответа на два или более вопросов, но это потребует более сложного редактирования файла с программами, и здесь этот вопрос не будет рассмотрен.

7. Заключение

Предложенные в статье инструментарий и методика анализа индексов, которые продемонстрированы на рейтингах *FT* 500, годятся не только для любых рейтингов, но и для практически любых других структурированных финансовых и экономических данных.

Продвинутые пользователи, у которых накопилась библиотека своих кодов для работы с данными, тоже могут извлечь пользу из AWK. Код программ AWK можно выполнять из других программ, например R и SAS. Более того (хотя, наверное, такая потребность редко возникнет) программы на других языках можно выполнять из AWK, например скрипты R и Perl.

Приведем еще один пример структурированных текстов с интересной информацией, находящихся в свободном доступе, – Google Trends. Отметим, что, как

[1] Еще раз отмечаем, что речь идет о программах, обрабатывающих одни и те же данные.

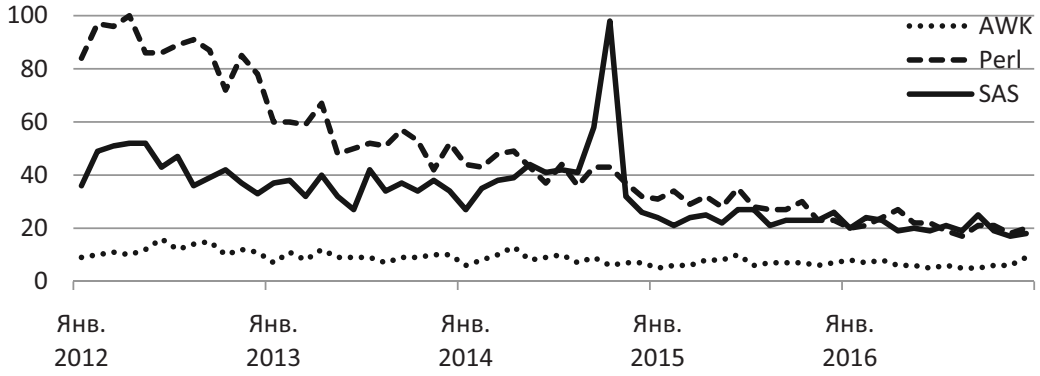

Рис. 5. Google Trends: относительная интенсивность поиска россиянами в Google AWK, Perl и SAS, 2012–2016 гг.

и другие данные из системы Edgar (*Electronic Data Gathering*, *Analysis*, *and Retrieval system*), данные Google Trends использовались в публикациях топ-журналов по финансам (см., напр.: Garcia, Norli, 2012a; Da, Engelberg, Gao, 2011).

Google Trends содержит информацию об интенсивности поиска любого интересующего слова или фразы в Google. Данные об интенсивности поиска можно экспортировать в формате csv. На рис. 5 приведены данные из Google Trends об интенсивности поиска россиянами в 2012–2016 гг. для языков программирования AWK, Perl и SAS. Хотя язык AWK наименее популярен, разница в интенсивности интереса к этим языкам существенно сократилась за последние два года. Читатель сможет проверить, поднимется ли линия языка AWK после этой публикации. По нашему мнению, дело остается только за креативным мышлением и навыками программирования на языке AWK.

Источники

Ахо А., Хопкрофт Дж., Ульман Дж. Структуры данных и алгоритмы. М., 2003.

Брискоу С. Бизнес на фотоснимке // Ведомости. 2005. 14 июня.

Брискоу С. Условности сравнения // Ведомости. 2003. 21 мая.

Бухвалова В. А., Бухвалова В. В. Применение языка AWK для оперативной обработки экономической информации // Компьютерные инструменты в образовании. 2013. № 3. С. 3–13.

Бухвалова В. В., Никандрова О. В. Язык программирования AWK как инструмент обработки экономической информации // Математические модели и информационные технологии в менеджменте. СПб., 2001. Вып. 1. С. 94–102.

Бьянкуцци Ф., Уорден Ш. Пионеры программирования. Диалоги с создателями наиболее популярных языков программирования. М., 2011.

Оверченко М. FT 500: рейтинг крупнейших компаний мира – США снова на коне // Ведомости. 2014. 17 июля.

Оверченко М. Российские компании в рейтинге *Financial Times* вернулись по рыночной капитализации на 10 лет назад // Ведомости. 2015. 21 июня.

Пестон Р. FT 550: исторический рейтинг // Ведомости. 2000. 4 мая.

Романовский И. В. Еще несколько примеров использования AWK // Компьютерные инструменты в образовании. 2013. № 3. С. 20–27.

Da Z., Engelberg J., Gao P. In Search of Attention // The Journal of Finance. 2011. Vol. 66. P. 1461–1499.

Garcia D., Norli Ø. Geographic Dispersion and Stock Returns // Journal of Financial Economics. 2012a. Vol. 106. № 3. P. 547–565.

Garcia G., Norli Ø. Crawling EDGAR // The Spanish Review of Financial Economics. 2012b. Vol. 10. P. 1–10.

Omran A., Said S. Saudi Arabia Could List Production Assets in Aramco IPO // The Wall Street Journal. Published on January 11, 2016.

Robbins A. Effective AWK Programming. O'Reilly Media, 2015.

Robbins A. GAWK: Effective AWK Programming. A User's Guide for GNU Awk. Ed. 4.1. August, 2016. URL: http://www.gnu.org/software/gawk/manual/gawk.pdf (дата обращения: 20.02.2017).

Robbins D. An Intro to the Great Language with the Strange Name // IBM developer Works. Published on December 01, 2000.

Robbins D. Commonthreads: Awk by Example, Part 2: Records, Loops, and Arrays // IBM developer Works. Published on January 01, 2001a.

Robbins D. Common Threads: Awk by Example, Part 3: String Functions and ... Checkbooks? // IBM Developer Works. Published on April 01, 2001b.