

БОЧАРОВ Б.П.

ВОЕВОДИНА М.Ю.

ГОМЗА Н.И.

г. Харьков

Технология конвертации информации с использованием программы AWK

В библиотечной практике нередко возникают достаточно трудоемкие задачи, решить которые с помощью стандартных средств даже очень «продвинутого» электронного каталога практически невозможно. Больше того, кроме трудоемкости, эти задачи отличаются еще и одновременным характером, т.е., как правило, они возникают один раз и впоследствии, с большой вероятностью, повторяться не будут.

Именно с такой задачей столкнулась одна из московских библиотек. Будучи организована совсем недавно, она располагала полученным от дарителей фондом в 50 тысяч наименований, автоматизированной системой ИРБИС и стояла перед острой необходимостью в считанные недели сформировать каталог библиотеки.

Последовательный ввод каждого наименования в каталог ИРБИСа потребовал бы нескольких месяцев напряженного труда, причем, не одного, а всех сотрудников библиотеки. Учитывая неподготовленность персонала к работе с программой, такая «всеобщая мобилизация» могла обернуться многочисленными ошибками.

Причем, в дальнейшем эти достаточно трудно устранимые ошибки могли оказывать влияние не только на качество каталога, но и на другие подсистемы программы.

Единственным благоприятным для библиотеки моментом было наличие списка литературы, выполненного в виде многостраничного word-овского файла. Однако библиотека не представляла себе, каким образом он может быть использован, а, кроме того, сам список не отвечал требованиям библиографического описания.

Позволяет ли описанная выше ситуация избежать колоссальной рутинной работы по вводу библиографических данных в электронный каталог? Безусловно. Для этого необходимы несколько совсем несложных программ и использование технологии автоматизированного преобразования информации, изложенной в [1-3]. Хотя отказаться от ручного ввода, т.е. полностью автоматизировать процесс нам не удастся, наши усилия будут существенно минимизированы.

В настоящей статье подробно описываются все этапы решения задачи. Файлы, которые упоминаются ниже, представлены в приложении к статье, которое находится на сайте журнала по адресу <http://lib-journal.ru/vsp/art8app.zip>

Для примера воспользуемся небольшим фрагментом списка (файл list.doc находится в каталоге part1). Итак, нам предстоит преобразовать в вид, понятный ИРБИСу, следующий файл:

Полка К1.

Бузескул, Владислав Петрович. История финской демократии. СПб. Типогр. М.М.Стасюлевича. 1909. 468 с.

Редкин, Петр Григорьевич. Из лекций по истории философии права в связи с историей философии вообще. Т. 6. СПб. Типогр. М.М. Стасюлевича. 1891. 498 с. - оторван переплет.

Кулаковский, Юлиан. Прошлое Тавриды. Краткий исторический очерк. Киев. Типо-литогр. И.Н. Кушнерева. 1906. 141 с. - с автографом: "Дорогому Алексею Ивановичу Соболевскому с приветом от автора. 18 июня 1906 г."

Полка К2.

Пыпин, А.Н. Спасович В.Д. История славянских литератур. 2-е изд., вновь перераб. и доп. Два тома. Т. 2. СПб. Изд. типографии М.М. Стасюлевича. 1881. 24+680+XIX с.

Воеводский Л.Ф. Введение в мифологию Одиссеи. Ч. 1. Одесса. Типогр. П.А. Зеленого (б. Г. Ульриха). 1881. 8+236 с.

Гиляров, А.Н. Предсмертные мысли XIX века во Франции. Киев. Тип. С.В. Кульженко. 1901. 663 с.

De usu praepositionum apud Homerum. Epistola ad Frid. Augustum Wolfium, Homeri inter Germanos sospitorem, auctore G. Frid. C. Gynthero, scholae Bernburgensis collega quarto. Halis Saxonum. E libraria Hemmerdeana. 1814. 44 S.

Очевидно, что решение этой задачи «в лоб», т.е. перенесение информации из файла в электронный каталог с помощью функций «копировать» и «вставить» не дает значительной экономии. Следовательно, необходимо преобразовать информацию таким образом, чтобы ее можно было экспортировать в ИРБИС. Самое естественное решение – осуществить разбивку по полям и преобразовать список в формат MARC.

В ИРБИСе задача слегка упрощается, так как можно экспортировать текстовые файлы.

Даже беглый взгляд на исходный файл показывает, что в файле есть повторения. Поэтому для избежания повторного ввода было бы совсем неплохо отсортировать исходный файл. При сортировке необходимо не потерять информацию о том, на какой полке книга лежит и, на всякий случай, сохранить позицию записи в исходном файле.

Кроме того, для дальнейшей работы было бы очень полезно всегда иметь возможность сопоставить записи до и после преобразования.

Таким образом, на первом этапе мы приводим информацию к виду удобному для работы.

Все программы, используемые на этом этапе, находятся в каталоге part1 приложения.

Сначала нам нужно открыть исходный файл в Word и сохранить его как текстовый (list.txt). Все остальные преобразования выполняются с помощью программ, вызываемых из командного файла go.bat (советуем ознакомиться с [1,3], детали вызова и взаимодействия программ в данной работе не повторяются).

Командный файл go.bat выполняет следующие действия:

1. Программа p1.awk дописывает в конец записи информацию о полке, на которой находится книга и номере записи в исходном файле.

```
..\gawk --re-interval -f p1.awk list.txt > list1.txt
```

2. С помощью программы transdwk происходит перекодировка информации и набора символов Windows-1251 в DOS (системная программа сортировки работает с набором символов DOS).

```
..\transdwk wd list1.txt
```

3. Сортировка с помощью стандартной системной программы sort.

```
sort list1.txt > list2.txt
```

4. Обратная перекодировка информации в набор символов Windows-1251.

```
..\transdwk dw list2.txt
```

5. Программа p2.awk преобразует информацию к специальному виду, удобному для разбивки по полям.

```
..\gawk --re-interval -f p2.awk list2.txt > src.txt
```

Программа p1.awk работает следующим образом:

В переменной nr будет храниться текущий номер записи.

```
BEGIN{  
  nr = 0;  
}
```

Если строка начинается со слова «Полка», из этой строки удаляются все символы, которые не являются цифрами.

Номер полки записывается в переменную `plk`.

```
    /^Полка/{
    plk = $0;
    gsub(/^[^0-9]/, "", plk);
    next;
    }
```

Формируется запись для вывода в виде:

```
исходный текст#№ записи#№ полки#
{
  nr++;
  printf("%s#%4.4d#%4.4d#\n", $0, nr, plk);
}
```

С помощью программы `p2.awk` представляем запись в виде двух строк, в начале второй строке сохраняется № записи и № полки.

Программа работает следующим образом:

В переменную `i` записывается номер символа `#` в строке.

```
{
  s = $0;
  i = match(s, /#/);
```

В переменной `s1` сохраняется исходный текст записи, а в переменную `s2` – информация о номере записи и номере полки.

```
  s1 = substr(s, 1, i-1);
  s2 = substr(s, i);
```

Информация выводится в файл.

```
  print s1;
  print s2 s1;
}
```

Результат совместной работы всех программ представлен в файле `src.txt`, фрагмент которого приведен ниже.

De usu praepositionum apud Homerum. Epistola ad Frid. Augustum Wolfium, Homeri inter Germanos sospitorem, auctore G. Frid. C. Gynthero, scholae Bernburgensis collega quarto. Halis Saxonum. E libraria Hemmerdeana. 1814. 44 S.

#0007#0002#De usu praepositionum apud Homerum. Epistola ad Frid. Augustum Wolfium, Homeri inter Germanos sospitorem, auctore G. Frid. C. Gynthero, scholae Bernburgensis collega quarto. Halis Saxonum. E libraria Hemmerdeana. 1814. 44 S.

Бузескул, Владислав Петрович. История афинской демократии. СПб. Типогр. М.М.Стасюлевича. 1909. 468 с.

#0001#0001#Бузескул, Владислав Петрович. История афинской демократии. СПб. Типогр. М.М.Стасюлевича. 1909. 468 с.

Воеводский Л.Ф. Введение в мифологию Одиссеи. Ч. 1. Одесса. Типогр. П.А. Зеленого (б. Г. Ульриха). 1881. 8+236 с.

#0005#0002#Воеводский Л.Ф. Введение в мифологию Одиссеи. Ч. 1. Одесса. Типогр. П.А. Зеленого (б. Г. Ульриха). 1881. 8+236 с.

Теперь приступим к разбивке информации по полям. Сначала найдем и тщательно изучим таблицу кодирования полей в ИРБИС. Легко увидеть, что из всего многообразия полей для кодирования нашей исходной информации вполне достаточно следующей таблицы:

Фамилия 1-го автора	700^A
Инициалы 1-го автора	700^B
Полное имя 1-го автора	700^G
Фамилия 2-го автора	701^A
Инициалы 2-го автора	701^B
Полное имя 2-го автора	701^G
Заглавие	200^A
Сведения, относящиеся к заглавию	200^E
Сведения об ответственности	200^F
Номер серии	225^V
Заглавие серии	225^A
Заглавие выпуска (часть тома)	923^I
Номер выпуска	923^H
Заглавие части (часть выпуска)	923^E

Номер части	923^K
Сведения об издании	205^A
Город(а) издательства	210^A
Издательство	210^C
Год издания	210^D
Объем (число страниц)	215^A
Примечание: пометки автора	395
Примечания о сохранности	399

Для экспорта в ИРБИС необходимо получить примерно такой текстовый файл (в примере представлена одна запись):

```
#700: ^АГиляров^ВА. Н.
```

```
#200: ^АПредсмертные мысли XIX века во Франции^ФА.
```

Н. Гиляров

```
#210: ^АКиев^СТип. С.В. Кульженко^D1901
```

```
#215: ^А663 с
```

```
*****
```

Коды полей будем вводить прямо в файл src.txt, полученный выше. Этот файл и программа, позволяющая перевести полученный файл в ИРБИС, находятся в каталоге part2 приложения.

При вводе полей в файл src.txt будем придерживаться следующих правил:

1. Из двух строк каждой записи редактированию подлежит только первая строка, вторую строку оставляем без изменений. По этой строке, которая содержит и дополнительную информацию, можно восстановить запись в случае ошибки или какой-нибудь другой непредвиденной ситуации.

2. Если в процессе кодировки возникают трудности, то запись можно не трогать, а проанализировать её позднее.

3. Если записи повторяются, то мы обрабатываем первую, а в первой строке остальных ставим символы |d. Отметим, что после сортировки все повторяющиеся записи следуют подряд.

4. Книги на иностранных языках пока обрабатывать не будем, просто отметим их символами |f. Разбивка на поля будет производиться по мере привлечения специалистов, владеющих языками, на которых написаны книги.

5. В каждой строке должно быть одно поле, строка начинается с кода поля (три цифры). Подполя отделяются друг от друга символами |^X (X – код подполя, одна латинская буква).

6. Несложно заметить, что в описании включенных в список произведений печати достаточно часто повторяется конструкция вида:

Одесса. Типогр. П.А. Зеленого (б. Г. Ульриха). 1881.

8+236 с.

Вероятно, составитель списка считал, что подобным образом следует записывать место издания, издательство, год издания, количество страниц. Зарезервируем для такой последовательности набор символов &0 и разобьем ее на поля несколько позднее с помощью программы.

Для того, чтобы отличить сокращенное название города от полного, после полного названия будем ставить двоеточие, как то предусмотрено ГОСТом.

Одесса.: Типогр. П.А. Зеленого (б. Г. Ульриха). 1881.

8+236 с.

Используемые нами правила очень просты, для обработки информации не нужно иметь высокую квалификацию, связанную с работой на компьютере, достаточно уметь правильно составлять библиографическое описание.

В результате разбивки по полям в файле src.txt находятся примерно такие записи:

700^АГиляров^ВА. Н.

200^АПредсмертные мысли XIX века во Франции
&0Киев.: Тип. С.В. Кульженко. 1901. 663 с.

#0006#0002#Гиляров, А.Н. Предсмертные мысли XIX
века во Франции. Киев. Тип. С.В. Кульженко. 1901. 663 с.

После завершения корректировки файла `src.txt`, переведем информацию в вид, понятный для ИРБИСа. Для этого используется программа `to_irb.awk` (вызывается из командного файла `to_irb.bat`).

Перейдем к подробному описанию программы `to_irb.awk`. В начале файла программы находятся функции, необходимые для дальнейшей работы.

Функция `trim` удаляет из строки все начальные и конечные пробелы, а также заменяет любое число пробелов (идущих подряд) в середине строки на один пробел.

```
function trim(str){
    trim_str = str;
    gsub(/^[+/,",",trim_str);
    gsub(/+$/,",",trim_str);
    gsub(/[2,},/,"“,trim_str);
    return trim_str;
}
```

Функция `get_sf` выделяет из строки `str` подполе, символ подполя задается параметром `sym`. Если подполе не найдено, то возвращается пустая строка (см. также [1]).

Переменные `i,n,arr,sym1,val` – локальные (см. [3]).

```
function get_sf(str,sym, i,n,arr,sym1,val){
    n = split(str,arr,FS);
    if(n == 0) return "";
    for(i=2; i<=n; i++){
        sym1 = substr(arr[i],1,1);
        if(sym1 == sym){
            val = arr[i];
            sub(/^[^.,",",val);
            return val;
        }
    }
    return "";
}
```

Функция `set_sf` записывает в строку `str` подполе, определяемое символом `sym` и значением `nval`. Если подполе уже есть в `str`, то его значение изменяется на `nval`, в противном случае в исходную строку добавляется новое подполе.

```
function set_sf(str,sym,nval, i,n,arr,sym1,val,pri,tmp){
    n = split(str,arr,FS);
    if(n == 0)
        tmp = str;
    else
        tmp = arr[1];
    pri = 0;
    for(i=2; i<=n; i++){
        sym1 = substr(arr[i],1,1);
        val = arr[i];
        sub(/^(.*/,sym1),val);
        if(sym1 == sym){
            tmp = tmp FS sym1 nval;
            pri = 1;
        }
        else tmp = tmp FS sym1 val;
    }
    if(pri == 0) tmp = tmp FS sym nval;
    return tmp;
}
```

Функция `out_str` переводит строку `str` в понятный для ИРБИС вид, например, строка `700^АГиляров^ВА.Н.` преобразуется в строку `#700: ^АГиляров^ВА. Н.`

```
function out_str(str, val, fld){
    val = str;
    fld = substr(val,1,3);
    sub(/^(.../,sym1),val);
    return sprintf("#%d: %s",fld,val);
}
```

Основная программа начинается с установки системной переменной FS (разделитель полей в строке, обрабатываемой программой AWK) и инициализации вспомогательных переменных.

```
BEGIN{
  FS = "^";
  pri = 0;
}
```

Если запись начинается с символа, отличного от цифры или &, то эта запись не записывается в выходной файл (и не преобразовывается для экспорта из ИРБИС). Повторяющуюся запись (начинается с |d) записываем в файл dbf.txt, запись книги на иностранном языке (начинается с |f) – в файл frn.txt. В любом другом случае записываем информацию в файл out.txt и оставляем для дальнейшей корректировки.

```
/^[^0-9#&|/ {
  t0 = $0
  t1 = substr(t0,1,2);
  sub(/\|[a-z|/, "", t0);
  fn = "oth.txt";
  if(t1 == "|d") fn = "dbf.txt";
  if(t1 == "|f") fn = "frn.txt";
  print t0 > fn;
  getline;
  print $0 > fn;
  next;
}
```

Если считана последняя строка в описании книги (она начинается с #), то мы завершаем запись описания в выходной файл (в файле экспорта для ИРБИС записи отделяются друг от друга строкой *****), устанавливаем признак начала записи (pri = 0) и очищаем служебную переменную otv (она будет использоваться для формирования сведений об ответственности, подполе 200^F).

```

/^#/{
  pri = 0;
  print "*****";
  otv = "";
  next;
}

```

В начало каждой записи добавляем служебную информацию для ИРБИС и сбрасываем признак начала записи (pri = 1).

```

{
  if(pri == 0){
    print "#102: RU";
    print "#101: rus";
    print "#920: PAZK";
    pri = 1;
  }
}

```

Обрабатываем поле «1-й автор» (строка начинается с 700) или «2-й автор» (строка начинается с 701). В переменную fam записываем фамилию автора, в переменную ini – его инициалы, а в переменную iot – полное имя.

```

/^700|^701/{
  tmp = $0;
  fam = get_sf($0,"A");
  ini = get_sf($0,"B");
  iot = get_sf($0,"G");
}

```

Если задано полное имя, а инициалы не определены, формируем подполе «инициалы» и записываем его в строку.

```

if(ini == "" && iot != ""){
  n = split(iot,arr_iot," ");
  for(i = 1; i <= n; i++){
    ini = ini substr(arr_iot[i],1,1) ".";
    if(i < n) ini = ini " ";
  }
}

```

```

    }
    tmp = set_sf(tmp,"B",ini);
  }

```

Формируем информацию об авторе, которую в дальнейшем запишем в сведения об ответственности (например, И.И. Иванов). Записываем поле 700 (701) в выходной файл.

```

if(otv != "") otv = otv " ";
otv = otv ini " fam;
print out_str(tmp);
next;
}

```

Обрабатываем поле «заглавие» (строка начинается с 200). Если часть сведений об ответственности уже записана в переменную otv, то дописываем ее в подполе 200^F. Затем записываем поле 200 в выходной файл.

```

/^200/{
  tmp = $0;
  otv0 = get_sf(tmp,"F");
  if(otv != ""){
    if(otv0 != "") otv0 = " ";
    otv0 = otv otv0;
    tmp = set_sf(tmp,"F",otv0);
  }
  print out_str(tmp);
  next;
}

```

Приступаем к обработке строки, в которой представлены место издания, издательство, год издания и количество страниц. Отдельные части записи отделены друг от друга точками, причем внутри описания издательства может быть сколько угодно точек.

Описание этого фрагмента программы приведем на конкретном примере:

&0Одесса.: Типогр. П.А. Зеленого (б. Г. Ульриха). 1881.

8+236 с.

∧&0/{

tmp = \$0;

Удаляем из строки начальную служебную информацию (&0).

sub(∧&0/,"",tmp);

Разбиваем строку на подстроки, символ-разделитель – точка.

Записываем подстроки в массив arr0 (в нижней строке таблицы находятся индексы подстрок в массиве):

Одесса: Типогр П А Зеленого (б Г Ульриха) 1881 236 с								
1	2	3	4	5	6	7	8	9

Исходная строка заканчивается на символ-разделитель, поэтому в массив arr0 запишется пустая строка с индексом 10. Количество элементов массива (nn0) равно 10.

nn0 = split(tmp,arr0,".");

Заносим город издания в переменную f210A (значение – Одесса).

f210A = trim(arr0[1]);

Заносим число страниц в переменную f210D (значение – 1881).

f210D = trim(arr0[nn0-2]);

Заносим год издания в переменную f210D (значение – 8+236 с).

f215A = trim(arr0[nn0-1]);

Заносим в переменную f210C издательство (элементы массива 3-7).

f210C = arr0[2];

for(i=3; i<nn0-2; i++)

f210C = f210C "." arr0[i];

f210C = trim(f210C);

Если в подполе «город издания» есть двоеточие (указано полное название города), то удаляем его, в противном случае добавляем в конец подполя точку (используется сокращенное название города).

```
if(f210A!~/:/$/)  
f210A=f210A“.”;
```

```
else  
sub(/:/$/,””,f210A);
```

Формирование и вывод поля 210.

```
tmp = “210^A” f210A “^C” f210C “^D” f210D;
```

```
print out_str(tmp);
```

Вывод поля 215.

```
tmp = “215^A” f215A;
```

```
print out_str(tmp);
```

```
next;
```

```
}
```

Все другие поля выводим в выходной файл без изменений.

```
{  
print out_str($0);  
}
```

Предварительная работа проделана, и все готово для ввода информации. После обработки нескольких сотен записей выяснилось, что при вводе кодов полей непосредственно в текстовом файле возникают некоторые проблемы.

1. При быстром вводе информации появляются ошибки, связанные с нажатием «не той» клавиши.

2. Для восстановления записи после ошибочного ввода требуется выполнить слишком много действий: удаление ошибочно сформированных строк, копирование и вставка информации. Альтернативный вариант – исправление ошибок – занимает зачастую еще больше времени.

3. При вводе кодов подполей приходится вводить каждый раз символ “^”

Это упражнение очень быстро приводит к вполне разумной мысли, что автоматическое добавление этого символа увеличит производительность труда в два раза. Попутно хотелось бы в типичных случаях вводить и коды полей (три цифры) с помощью нажатия одной клавиши.

Все эти проблемы можно решить, если контролировать ввод пользователя на уровне нажатия клавиш. К сожалению, программа AWK не предназначена для разработки интерактивных систем, поэтому необходимо использовать другую среду программирования. Выбор DELPHI обусловлен субъективными предпочтениями авторов.

Итак, необходимо достаточно быстро (напоминаем, что выполняется срочная работа) создать программу, которая позволяет оптимизировать ввод пользователя. Эта программа (мы назвали ее Quick Marc, исполняемый файл – qmarc.exe) представлена в каталоге part3 приложения. Исходный проект DELPHI находится в подкаталоге pgm.

Quick Marc работает с файлами, которые имеют расширение qmg (по сути, это – обычные текстовые файлы, специальное расширение используется для удобства, например, можно задать программу qmarc.exe для открытия файлов с таким расширением по умолчанию).

Перейдем к описанию функциональных возможностей программы Quick Marc. Главное окно программы представлено на рис. 1.

Поле для редактирования информации (белое) находится сразу же под меню. Ниже размещены два информационных поля («Текущая запись» и «Предыдущая запись») и поля навигации по файлу («N» – № обрабатываемой в настоящий момент записи, «Всего» – общее количество записей в файле).

Навигация по файлу осуществляется с помощью клавиш:

- | | |
|----------|--------------------------------|
| PageUp | – переход к предыдущей записи, |
| PageDown | – переход к следующей записи, |

- Ctrl+PageUp – переход в начало файла,
Ctrl+PageDown – переход в конец файла.

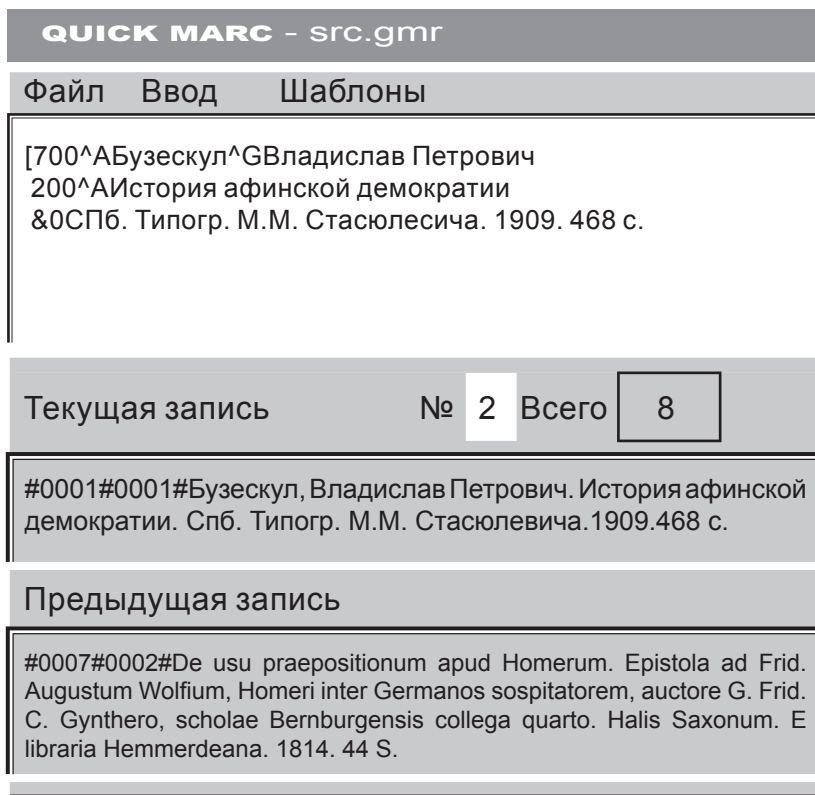


Рис. 1. Главное окно программы Quick Marc

Информационные поля «Текущая запись» и «Предыдущая запись» предназначены для быстрого определения повторяющихся записей.

Работа с файлами в программе упрощена до предела, предусмотрены только опции открытия и сохранения файла (см. рис. 2).

Открыть
Сохранить Ctrl + S

Рис. 2. Меню “Файл”

Предусмотрены два режима ввода информации. В первом режиме пользователю разрешено вводить только латинские буквы и следующие символы (нажатие остальных клавиш игнорируется):

- латинские буквы;
- пробел;
- «.» (точка);
- «,» (запятая);
- «:» (двоеточие);
- «;» (точка с запятой);
- «[» (открывающаяся квадратная скобка);
- «]» (закрывающаяся квадратная скобка);
- «(» (открывающаяся круглая скобка);
- «)» (закрывающаяся круглая скобка);
- «&» (амперсant);
- «-» (дефис).

В этом случае цвет текста в поле редактирования информации – черный.

Если введена латинская буква, то перед ней добавляется символ «^». После символа «&» добавляется «0».

В другом режиме пользователю разрешен ввод любого символа (цвет текста в поле редактирования – синий).

Переключение между режимами осуществляется комбинацией клавиш Ctrl+I (см. рис. 3).

Переключить режим ввода	Ctrl + I
Восстановить текущую запись	Ctrl + R
Отметить повторную запись	Ctrl + D

Рис. 3. Меню “Ввод”

Кроме того, предусмотрена возможность восстановить текущую запись и отметить повторяющуюся (поставить перед строкой).

700^A	Первый автор	F1
700^A	Следующие авторы	Ctrl + F1
200^A	Заглавие	F1
205^A	Издание	F1
210^A	Город	F1
215^A	Количество страниц	F1
225^A	Название серии	F1
923^H	Название 1-ой единицы деления	F1
395	Пометки (дарственная надпись и т.д.)	F1
399	Сохранность книги	F1

Рис 4. Меню «Шаблоны».

И, наконец, последнее средство, облегчающее ввод информации – быстрый ввод кода поля (см. рис. 4). Например при нажатии клавиши «F1» выводится «700^A».

Логику работы программы описывать не будем (читатель, владеющий основами программирования на DELPHI, легко в ней разберется самостоятельно).

Отметим только, что программу можно легко переделать для конвертации любых списков литературы в формат MARC.

Литература

1. Бочаров Б.П., Воеводина М.Ю. AWK - универсальная программа работы с текстовыми файлами. // Библиотеки учебных заведений.- 2002.- N 4.- с. 39-53.
2. Бочаров Б.П., Воеводина М.Ю. Глобальная корректировка БД с использованием программы AWK. // Библиотеки учебных заведений.- 2003.- N 7.- с. 37-59.
3. Бочаров Б.П., Воеводина М.Ю. Формирование отчетов в электронных каталогах. // Библиотеки учебных заведений.- 2003.- N 10.- с. 42-60.