Российская Академия Наук

РОССИЙСКАЯ АКАДЕМИЯ НАУК

ИНСТИТУТ ПРОБЛЕМ БЕЗОПАСНОГО РАЗВИТИЯ АТОМНОЙ ЭНЕРГЕТИКИ

RUSSIAN ACADEMY OF SCIENCES

NUCLEAR SAFETY INSTITUTE

Препринт ИБРАЭ № IBRAE-1998-02

Preprint IBRAE-1998-02

А.Глушко

ПРОГРАММА ПОДГОТОВКИ БАЗ ДАННЫХ ДЛЯ СПЕЦИАЛИЗИРОВАННОЙ ГИС RVS ПО РАДИАЦИОННО-ГИГИЕНИЧЕСКОЙ ОБСТАНОВКЕ ЗАРАЖЕННЫХ ТЕРРИТОРИЙ В РЕЗУЛЬТАТЕ АВАРИИ НА ЧАЭС

Москва 1998

Moscow 1998

УДК 621.311.25:621.039.538

Глушко А.В. ПРОГРАММА ПОДГОТОВКИ БАЗ ДАННЫХ ДЛЯ СПЕЦИАЛИЗИРОВАННОЙ ГИС RVS ПО РАДИАЦИОННО--ГИГИЕНИЧЕСКОЙ ОБСТАНОВКЕ ЗАРАЖЕННЫХ ТЕРРИТОРИЙ В РЕЗУЛЬТАТЕ АВАРИИ НА ЧАЭС. Препринт № IBRAE-98-02. Москва. Институт проблем безопасного развития атомной энергетики РАН. Февраль 1998. 35 с. — Библиогр.: 1 назв.

Аннотация

В настоящей работе описан комплекс для подготовки и обработки специализированных баз данных, разработанный в процессе работы по созданию Центрального банка обобщенных данных ИБРАЭ РАН по анализу последствий аварии на ЧАЭС.

Glushko A.V. DATABASES DEVELOPMENT PROGRAM FOR THE SPECIALIZED GEOGRAPHICAL INFORMATION SYSTEM RVS ON RADIATION AND HYGIENIC CONSEQUENCES OF THE CHERNOBYL ACCIDENT. Preprint IBRAE-98-02. Moscow. Nuclear Safety Institute. February 1998. 35 p. — Refs.: 1 items.

Abstract

The work presents the specialized databases development and processing program system. The system was developed while creating the Generalized Databank on the Chernobyl accident consequences in IBRAE RAS.

Программа подготовки баз данных для специализированной ГИС RVS по радиационно-гигиенической обстановке зараженных территорий в результате аварии на ЧАЭС

А.Глушко

ИНСТИТУТ ПРОБЛЕМ БЕЗОПАСНОГО РАЗВИТИЯ АТОМНОЙ ЭНЕРГЕТИКИ
113191, Москва, ул. Б. Тульская, 52
тел.: (095) 955-26-20, факс: (095) 230-20-29, Интернет: http://www.ibrae.ac.ru

Содержание

Введение

Программа разрабатывалась для подготовки данных при работе со специализированной ГИС RVSVIEW, предназначенной для представления и анализа данных по радиационно-гигиенической обстановке и дозам облучения населения по областям повышенного радиационного риска.

В настоящее время Центральный банк обощенных данных ИБРАЭ РАН располагает существенными объемами данных по радиационно-гигиеническому и социально-демографическому состоянию ряда регионов. Эти данные организованы и поддерживаются в актуальном состоянии с помощью специальной программной системы ИПС РГД, базирующейся на стандартном представлении информации в виде символьных DBF-файлов. Ориентация указанной программной системы на использование структур базы данных DBASE вполне оправдана, когда речь идет о подготовке, хранении и поддержке в актуальном состоянии значительных объемов числовой и символьной информации. Эта система позволяет также сравнительно эффективно обработать запросы, допускающие ответ в числовом и/или табличном виде.

Специализированная ГИС RVSVIEW предназначена для отображения данных, содержащихся в ИПС-РГД, в виде графических образов, размещенных на плане соответствующей местности, и определенного анализа этих данных.

Можно выделить три основных блока системы :

- средства доступа к фактографической базе данных;
- средства отображения плана местности;

- средства отображения и анализа фактографических данных.

Средства доступа к фактографичесой базе данных обеспечивают для пользователя текстовой интерфейс с DBF-файлами, поддерживаемыми ИПС -РГД. Сюда входят возможности просмотра, сортировки, отбора подмножеств записей по значениям их полей и.т.п.

Средства отображения плана местности дают возможность получить на экране один или более фрагментов карты. Базовым фрагментом плана местности выбрана карта области. Отображаемыми данными на плане могут быть: границы районов, населенные пункты, реки, водоемы, лесные массивы, шоссейные и железные дороги.

Пользователю предоставляется возможность выбрать фрагмент карты и отобразить его в одном из 3-4 х стандартных масштабов, изменить насыщенность карты, отменив отображение несущественных для него элементов.

Для работы с атрибутивной информацией выбран наиболее естественный подход - непосредственное использование DBF-файлов. При этом технологическая цепочка выглядит следующим образом. Средствами системы ИПС РГД подготавливается необходимая информация, включая и такие данные, как географические координаты населенных пунктов. На этом этапе определяется, какие именно из имеющихся в базе ИПС РГД полей подлежат передаче в подсистему визуализации и анализа. Далее средствами ИПС РГД подготавливаются DBF-файлы по каждому из интересующих нас регионов. Принято решение в качестве блока информации рассматривать данные по одной области. На уровне детализации, принятой в системе ИПС РГД это соответствует 1-4 тыс. записей, что достаточно удобно технологически. Полученные таким образом файлы и являются атрибутивной базой данных.

К атрибутивной информации относятся данные по радиационно-гигиенической обстановке и дозам облучения населения. Исходной является база данных формата DBF, каждая запись которой содержит данные по одному объекту системы. С точностью до подуровней она имеет следующую структуру

001.ГЕНЕРАЛЬНАЯ ИНФОРМАЦИЯ
002.СОЦИАЛЬНО-ЭКОНОМИЧЕСКОЕ РАЗВИТИЕ
003.ЗАГРЯЗНЕНИЕ ТЕРРИТОРИИ РАД/НУКЛИДАМИ
004.ДАННЫЕ О ЗАГРЯЗНЕНИИ ПРОДУКТ.ПИТАНИЯ
005.НАКОПЛЕННЫЕ ДОЗЫ ОБЛУЧЕНИЯ НАСЕЛЕНИЯ
006.ДОЗЫ ДЛЯ ПРИНЯТИЯ РЕШЕНИЙ
007.ПРОГНОЗ ДОЗЫ ОБЛ. НАСЕЛЕНИЯ ЗА ЖИЗНЬ
008.ДАТЫ ПРИНЯТИЯ РЕШЕНИЙ

При разработке программ отображения и представления картографической информации возникает проблема системной поддержки проектируемых и разрабатываемых программ. Одной из проблем является необходимость формировать специализированные базы данных на основе разнородных источников (различные текстовые списки, форматы баз данных DBF и MDBS). При этом приходится проводить сравнительное тестирование как лексико-графическое, так и на соответствие по административно-территориальному делению, включая проверку соответствия по координатам, обеспечивать преемственность проводимой корректировки данных.

Для выполнения указанной задачи была создана программа LBASE. Был разработан формат структуры баз данных, на его основе можно легко изменять структуру баз данных: корректировать наименования, порядок и размерность полей, а также объединять различные базы данных, создавать новые, и проводить различные процедуры по обработке баз данных. Учитывая специфику стоящих при подготовке баз данных проблем был разработан достаточно широкий набор утилит, выбор и использование, которых определяться конкретными данными по определенному региону.

Программа LBASET поддерживает работу с базой данных REGASS BD, разработанной НПО «Тайфун» в рамках системы REGASS и предназначенной для организации данных о загрязнениях природных сфер.

В названии разделов главы «2 Список процедур программы Lbase» в квадратных скобках приводится код процедуры, который используется для классификатора (4.4 Классификация процедур программы Lbase).

В настоящее время программа доступна по адресу:
http://glushko.ibrae.ac.ru/archive/lbase.arj

1. Список процедур программы Lbase

1. Вывод структуры *.dbf - файла [1.1]

Структура имеет вид текстового файла и может редактироваться в любом редакторе. См. 1.2, 2, 25.
По умолчанию берется база данных в текущей директории и вывод осуществляется в файл "LIST".
Пример:
lbase 1 File.Dbf Info.lst

2. Тестирование списка полей базы данных [1.2]

Тестирование производится на предмет выяснение неиспользуемых полей в базе данных '?'(поле считается неиспользуемым, если состоит из символов " 0."); указывается первая запись использованого поля; указываются интервалы для числовых полей, с указанием записей; указывается максимальный используемый размер для полей, с указанием записи, причем символом '•' помечаются используемые не в полном размере; также выводятся разница по смещению заголовка и смещения полей.

Выходной формат поддерживается командой 25.
Пример:
lbase 1 !! # File.Dbf Info.lst

3. Коррекция *.dbf - файла [2.1]

Коррекция осуществляется в 2 этапа:
1) Вывод структуры *.Dbf - файла. Пример выше.
2) Создание откорректированной базы данных, на основе откорректированной структуры, с возможным использованием системных процедур.
Можно использовать механизм согласования (cm. 24).

Если наименование поля не содержит служебных символов, то номер поля источника определяется по лексикографической идентификации.

После наименования поля могут следовать следующие служебные символы (в соединении с определенным номером, порядок использования приводимых символов соответствует порядку нижеследующего описания):

1. '@' - выполняется одна из системных операций:

0 - формирует XGRAD, с плавающей точкой. С помощью ключа '#', во входной команде передается номер XGRAD источника. Предполагается, поля XGRAD, XMIN, XSEC, YGRAD, YMIN,YSEC лежат последовательно. Пример приведен ниже.

1 - формирует YGRAD, с плавающей точкой;

2 - формирует POPUL1989. С помощью ключа '#', во входной команде передается номер поля MEN1989 источника. Предполагается, поля POPUL1989,MEN1989,WOM1989 располагаются последовательно.

Предлагается процедуру формирования полей POPUL1989,MEN1989, WOM1989 из полей POPUL1989,WOM1989 проводить в два этапа:

1) Формирование POPUL1989 (чистая), MEN1989 (= POPUL1989), WOM1989 (= WOM1989), расположенные последовательно, из POPUL1989 и WOM1989.

2) Заполнение поля POPUL1989 с помощью с помощью рассматриваемой операции. Можно данную процедуру осуществить за один этап (в '#' указывается номер POPUL1989 источника). Предложение проводить в два этапа опирается на общую рекомендацию: не перегружать коррекцию системной обработкой для большего удобства контроля, так как ошибка в системных указаниях приведет к искажению выходной базы данных и контролировать процедуры коррекции последовательно легче, чем все одновременно.

3 - пустое поле.

4 - сложение группы полей в обобщенной форме (см. символ '!' и об информационных строках).

14 - в 00.000 -> ' '

2. '&' - в данное поле пересылаются данные из поля источника по указанному номеру.Данная процедура используется для переменования поля.

3. '^' - номер базы данных. См. пример 4.

4. '!' - номер группы.

В листинге структуры могут находит информационные строки, имеющие форму:

.KOLG1.N11...N1N.KOLG2.N21...N2N.KOLGN.NN1...NNN.

где KOLG1,..,KOLGN - количество элементов в группе 0..N,

N11,.. - перечисление полей в группах.

Дополнительные ключи:

!Ncod - обрабатывается извлечение из базы данных по 4-х цифровому коду области cod. Данный ключ можно использовать для извлечения базы данных области из совокупной базы данных.

Пример 1: Переменовать поля X и Y в поля XGRAD и YGRAD.

> lbase 1 File.Dbf Info.lst

Получили примерный текст описания структуры:

<<<<<<<< Info.lst >>>>>>>>>

Name of File: File.Dbf
Number of Fields: 25
Number of Records: 886
Header Size: 834
Record Size: 157

Field Information

0 | ADMCODE C 11 0
1 | PLACENAME C 35 0
2 | X N 7 3
3 | Y N 7 3
4 | AVEPUP N 6 4
5 | POPUL1986 N 7 0
.....

<<<<<<<< >>>>>>>>>

Модифицируем структуру:

<<<<<<<< Inf.lst >>>>>>>>>

Name of File: File.Dbf
Number of Fields: 25
Number of Records: 886
Header Size: 834
Record Size: 157

Field Information

0 | ADMCODE C 11 0
1 | PLACENAME C 35 0
2 | XGRAD&2 N 6 3
3 | YGRAD&3 N 6 3
4 | AVEPUP N 6 4
5 | POPUL1986 N 7 0
.....

<<<<<<<< >>>>>>>>>

И выполняем коррекцию:

> lbase 2 $ File.Dbf Inf.lst FileNew.Dbf

Нам осталось проверить корректность данных FileNew.Dbf, удалить File.Dbf, переменовать FileNew.Dbf в File.Dbf.

Обратим внимание, формат данных X,Y был (7,3), а формат данных XGRAD,YGRAD является (6,3), в соответствии с требованиями RVS.

Общее замечание: редактирование или создание новой базы данных происходит только, если во входной строке присутствует ключ '$'.

Пример 2: На основе полей XGRAD, XMIN, XSEC, YGRAD, YMIN, YSEC сформировать поля XGRAD,YGRAD, в соответствии с требованиями RVS.

> lbase 1 File.Dbf Info.lst

Получили примерный текст описания структуры:

<<<<<<<< Info.lst >>>>>>>>>

Name of File: File.Dbf
Number of Fields: 25
Number of Records: 886
Header Size: 834
Record Size: 157

Field Information

0 | ADMCODE C 11 0
1 | PLACENAME C 35 0
2 | X N 7 3
3 | Y N 7 3
4 | XGRAD N 3 0
5 | XMIN N 2 0
6 | XSEC N 2 0
7 | YGRAD N 3 0
8 | YMIN N 2 0
9 | YSEC N 2 0
10 | AVEPUP N 6 4
11 | POPUL1986 N 7 0
.....

<<<<<<<< >>>>>>>>>

Модифицируем структуру:

<<<<<<<< Inf.lst >>>>>>>>>

Name of File: File.Dbf
Number of Fields: 25
Number of Records: 886
Header Size: 834
Record Size: 57

Field Information

0 | ADMCODE 1 0

```
 1 | PLACENAME      5 0
 4 | XGRAD@0        3 0
 7 | YGRAD@1        N 3 0
10 | AVEPUP         N 6 4
11 | POPUL1986      N 7 0
 ........

<<<<<<<<  >>>>>>>>>
```

Замечание: верхнее описание структуры не подлежит редактированию и первая колонка порядковых номеров полей не несет какой-либо нагрузки при коррекции базы данных.

И выполняем коррекцию:

> lbase 2 $ #4 File.Dbf Inf.lst FileNew.Dbf

Пример 3: На основе полей POPUL1989, WOM1989 сформировать поля POPUL1989, MEN1989, WOM1989.

> lbase 1 File.Dbf Info.lst
Получили примерный текст описания структуры:

```
<<<<<<<< Info.lst  >>>>>>>>>

Name of File:          File.Dbf
Number of Fields:      25
Number of Records:     886
Header Size:           834
Record Size:           157

Field Information

 0 | ADMCODE          C 11 0
 1 | PLACENAME       C 35 0
 2 | POPUL1986       N 7 0
 3 | WOM1986         N 7 0
 ........

<<<<<<<<  >>>>>>>>>
```

Модифицируем структуру:

```
<<<<<<<< Inf.lst  >>>>>>>>>

Name of File:          File.Dbf
Number of Fields:      25
Number of Records:     886
Header Size:           834
Record Size:           157

Field Information

 0 | ADMCODE          C 11 0
 1 | PLACENAME       C 35 0
 2 | POPUL1986       N 7 0
 3 | MEN1986&2       N 7 0
 3 | WOM1986&3       N 7 0
```

.....

<<<<<<<< >>>>>>>>>

Выполняем первую коррекцию:

> lbase 2 $ File.Dbf Inf.lst FileNew.Dbf

Модифицируем снова структуру:

<<<<<<<< Inf.lst >>>>>>>>>

Name of File: File.Dbf
Number of Fields: 25
Number of Records: 886
Header Size: 834
Record Size: 157

Field Information

0 | ADMCODE C 11 0
1 | PLACENAME C 35 0
2 | POPUL1986 N 7 0
3 | MEN1986 N 7 0
4 | WOM1986 N 7 0
.....

<<<<<<<< >>>>>>>>>

Выполняем вторую коррекцию:

> lbase 2 $ #3 FileNew.Dbf Inf.lst FileNewR.Dbf

Пример 4: Выходная база данных будет включать поля F1 из B1.DBF, F2 из B2.DBF, F3 из B3.DBF, а также поле SUM1, как сумму полей A1,A2 и поле SUM2, как сумму полей B1,B2,B3;

С помощью операции 1 команды LBASE, получаем листинги структур баз данных:

<<<<<<<< B1.lst >>>>>>>>>

.2.4.5.3.6.7.8.
Name of File: B1.Dbf
Number of Fields: ***
Number of Records: ***
Header Size: ***
Record Size: ***

Field Information

0 | ADMCODE C 11 0
1 | PLACENAME C 35 0
2 | POPUL1986 N 7 0
3 | F1 N 7 0
4 | A1 N 7 0
5 | A2 N 7 0
6 | B1 N 7 0
7 | B2 N 7 2

8 | B3 N 7 3
.....

<<<<<<<< B2.lst >>>>>>>>>

Name of File: B2.Dbf
Number of Fields: ***
Number of Records: ***
Header Size: ***
Record Size: ***

Field Information

0 | F2 N 7 0
.....

<<<<<<<< B1.lst >>>>>>>>>

Name of File: B3.Dbf
Number of Fields: ***
Number of Records: ***
Header Size: ***
Record Size: ***

Field Information

0 | F3 N 7 0
.....

Формируем структуру выходной базы данных для коррекции:

<<<<<<<< B1NEW.lst >>>>>>>>>

Name of File: B1.Dbf
Number of Fields: ***
Number of Records: ***
Header Size: ***
Record Size: ***

Field Information

0 | ADMCODE C 11 0
1 | PLACENAME C 35 0
2 | POPUL1986 N 7 0
3 | F1 N 7 0
4 | F2^1 N 7 0
5 | F3^2 N 7 0
6 | SUM1!1 N 7 0
7 | A1 N 7 0
8 | A2 N 7 0
9 | SUM2!2 N 7 2
10 | B1 N 7 0
11 | B2 N 7 2
12 | B3 N 7 3

<<<<<<<< >>>>>>>>>

Генерируем выходную базу данных:

> lbase 2 $ #3 F1.Dbf B1NEW.lst B1NEW.Dbf TMP B2.DBF B3.DBF

Пример 5: Выходная база данных будет включать поля XGRAD,YGRAD из B2.DBF, используя механизм согласования.

Файлы EDITXY.EQ, EDITXY.WRN получаем согласно описанию в команде 24.

С помощью операции 1 команды LBASE, получаем листинги структур баз данных:

<<<<<<<< B1.lst >>>>>>>>>

```
Name of File:          B1.Dbf
Number of Fields:      ***
Number of Records:             ***
Header Size:                   ***
Record Size:                   ***

Field Information

 0 | ADMCODE                   C 11 0
 1 | PLACENAME                 C 35 0
 2 | XGRAD                     N  6 3
 3 | YGRAD                     N  6 3
 4 | ADEKZNACH                 C 37 0
 5 | CTRLFLD          C  2 0
 6 | POPUL1989        N  7 0
 7 | MEN1989                   N  7 0
 8 | WOM1989                   N  7 0
 ........
```

<<<<<<<< B2.lst >>>>>>>>>

```
Name of File:          B2.Dbf
Number of Fields:      ***
Number of Records:             ***
Header Size:                   ***
Record Size:                   ***

Field Information

 0 | ADMCODE                   C 11 0
 1 | PLACENAME                 C 35 0
 2 | XGRAD                     N  6 3
 3 | YGRAD                     N  6 3
 4 | ADEKZNACH                 C 37 0
 5 | CTRLFLD                   C  2 0
 6 | POPUL1989                 N  7 0
 7 | MEN1989                   N  7 0
 8 | WOM1989                   N  7 0
 ........
```

Формируем структуру выходной базы данных для коррекции:

<<<<<<<< B1NEW.lst >>>>>>>>>

```
Name of File:           B1.Dbf
Number of Fields:       ***
Number of Records:              ***
Header Size:                    ***
Record Size:                    ***

Field Information

 0 | ADMCODE                     C 11 0
 1 | PLACENAME                   C 35 0
 2 | XGRAD&2^1                   N  6 3
 3 | YGRAD&3^1                   N  6 3
 4 | ADEKZNACH                   C 37 0
 5 | CTRLFLD                     C  2 0
 6 | POPUL1989                   N  7 0
 7 | MEN1989                     N  7 0
 8 | WOM1989                     N  7 0
 ........

<<<<<<<<<  >>>>>>>>>
```

Генерируем выходную базу данных:

```
> lbase 2 $ F1.Dbf B1NEW.lst B1NEW.Dbf TMP B2.DBF EDITXY.EQ EDITXY.WRN
```

Пример 6: Введение по пустым полям из согласованной базы данных

```
<<<<<<<<< F1.lst  >>>>>>>>>

Name of File:           F1.Dbf
Number of Fields:       ***
Number of Records:              ***
Header Size:                    ***
Record Size:                    ***

Field Information

 0 | ADMCODE                     C 11 0
 1 | PLACENAME       C 35 0
 2 | XGRAD&2^1                   N  6 3
 3 | YGRAD&3^1                   N  6 3
 4 | ADEKZNACH&3^1               C 37 0
 5 | CTRLFLD                     C  2 0
 6 | POPUL1989&6^1               N  7 0
 7 | MEN1989&7^1                 N  7 0
 8 | WOM1989&8^1                 N  7 0
 ........

<<<<<<<<<  >>>>>>>>>
```

```
> lbase 2 $ !+ F1.Dbf F1.lst T.Dbf T B2.DBF EDIT_XY.EQ EDIT_XY.WRN
```

4. Сравнительное тестирование 2-х баз данных по группе полей [2.2]

В процессе тестирования выводится информация о расхождениях по следующим градациям:

1. Информация по пропущенным и обнуленным полям ("CMPDBF.CMP").

2. О больших значениях ("CMPDBFB.CMP").
3. О меньших значениях ("CMPDBFL.CMP").
4. По уровням отклонения ("CMPDBF[n].CMP", где n - номер уровня отклонения).

По каждой градации и каждому уровню информация выводится в указанные отдельные файлы.
Уровни отклонения могут задаваться в абсолютных или относительных величинах.
Признак градации определяется по первому расхождению в цепочке групп полей.
При тестировании используется механизм согласования.
Выходная информация (*.cor) может использоваться для коррекции данных.

Информационные строки с начальным символом '.' определяют массив пар соответствующих полей:

.A1.B1...AN.BN.

где AI,BI - пара соответствующих полей 2-х баз данных.

Информационные строки с начальным сиимволом '+'(ОТНОСИТЕЛЬНЫЕ) ил '%'(АБСОЛЮТНЫЕ) определяют массив уровней:

+U1...UN.

или

%U1...UN.

где UI - соответствующий уровень

Системные информационные строки:

"+%" - тестирование POPUL1989, MEN1989, WOM1989 (см. ОБЩИЕ ПОЛОЖЕНИЯ, 1; уровни(отн): 0.05, 0.2).

"+%%" - тестирование дополнительного 9 параметра (см. ОБЩИЕ ПОЛОЖЕНИЯ, 1; уровни(отн): 0.05, 0.2, 0.5, 1).

Пример информационной строки (все поля числовые):

.10.10.11.11.12.12.13.13.14.14.15.15.
+%

Пример:

Тестирование POPUL1989,MEN1989,WOM1989. Базы данных соответствуют примеру 5 команды 2.1 F1.Dbf и B2.DBF.

Обратим внимание, что вызовы LBASE в примерах 5 от 2.1 и от 2.2 полностью совпадают. Определяющим отличием служит наличие информационной строки "+%" в структуре выходной базе данных.

Формируем структуру выходной базы данных для тестирования:

<<<<<<<< B1NEW.lst >>>>>>>>>

+%

Name of File:	B1.Dbf
Number of Fields:	***
Number of Records:	***
Header Size:	***
Record Size:	***

Field Information

0	ADMCODE	C	11 0
1	PLACENAME	C	35 0
2	XGRAD&2^1	N	6 3
3	YGRAD&3^1	N	6 3
4	ADEKZNACH	C	37 0
5	CTRLFLD	C	2 0
6	POPUL1989	N	7 0

7 | MEN1989 N 7 0
8 | WOM1989 N 7 0
.....

<<<<<<<< >>>>>>>>>

Генерируем выходную базу данных:

> lbase 2 $ F1.Dbf B1NEW.lst B1NEW.Dbf TMP B2.DBF EDITXY.EQ EDITXY.WRN

5. Коррекция базы данных по результатам сравнительного тестирования [2.3]

Данные для коррекции формируются в обработке по команде 2.2 в файлах *.cor, которые играют роль связующих элементов и несут информацию о коррекции.

Пример:

Провести коррекцию по превышению данных.

> lbase 2 $ F1.Dbf B1NEW.lst B1NEW.Dbf TMP B2.DBF CMPDBFB.CMP

6. Информация об *.mdf - файлах [3]

Информация об находящихся в текущей директории *.Mdf - файлах выводится в "mdf.rep".

Пример:

> lbase 3

7. Тестирование базы данных (*.dbf) [4]

Тестированию подвергается база данных находящаяся в текущей директории. Для тестирования требуется также, либо "REGION.OBL" (файл, содержащий вставку для "REGION.TXT"),либо непосредственно "REGION.TXT".

Выводится в файл "BS.REP" следующая информация:

1. Населенные пункты с неотождествленным типом кода
2. Коды населенных пункты с неотождествленным районным центром
3. Населенные пункты с неотождествленным районным центром (с коорд.)
4. Населенные пункты с неотождествленным районным центром (сорт. по коду)
5. Населенные пункты, включенные в район по ключу "101"
6. Населенные пункты с совпадающими характеристиками
7. Областной центр, НЕвключенный в базу данных (абс. совп.)
8. Областной центр, с отсутствием координат
9. Областной центр, с НЕсоответствием по координатам (*.dbf,region.txt)
10. Районные центры, НЕвключенные в район (абс. совп.)
11. Районные центры, с отсутствием координат
12. Районные центры, с НЕсоответствием по координатам (*.dbf,region.txt)
13. Районные центры, НЕимеющие населенные пункты
14. Районные центры, НЕимеющие населенные пункты (сорт. по коду)
15. Районные центры
16. Районные центры, с населенными пунктами
17. Содержание базы данных

Для сравнения строковых выражений применяется понятие "абсолютное совпадение" (см. П. 19).

Из базы данных в качестве основных выводятся поля AdmCode, PlaceName.

Для части "Насел. пункты с совп. хар-ми", может также выводится дополнительная информация по полям. Используется ключ в командной строке: ".N1.N2.N3...NN." - N1..NN - номера полей, или ".." - выводятся поля 6,4.

При обработке используется файл "REGION.TXT".

Также для системного контроля выводится файл "TEMP4".

Пример:

> lbase !7 4 ..

8. Сравнительная информация об *.mdf-файлах и "region.txt" (или "region.obl"). Формирование "region.add". [5]

Сравнительная информация об находящихся в текущей директории *.Mdf - файлах, в отношению к файлу, либо "REGION.OBL", либо непосредственно "REGION.TXT", выводится в "dirmdf.rep".

Файл "REGION.ADD", аналогичен "REGION.OBL", только содержит районные пункты, имеющие соответствующие ссылочные *.Mdf-файлы.

Пример:

> lbase 5

9. Сортировка базы данных [6]

Сортировке подвергается база данных находящаяся в текущей директории. Сортировка проводится по второму полю, предполагается, что это поле PlaceName.

Создается база данных TMP.DBF.

Пример:

> lbase 6 $

10. Поиск в текстовых порайонных списках по файлу запроса [7.1]

Предполагается, файл запроса "FIND" формируется на основе листинга BS.REP, полученного в результате тестирования базы данных (4). Осуществляется просмотр порайонных текстовых файлов (признак: наличие числа в составе имени *.Txt - файла), в текущей директории, на предмет лексикографического сравнения (см. 17.2).

В результирующий файл "FIND.RES" выводятся данные по следующим руппам:

1. Абсолютное совпадение (см. 4);
2. Вложенное совпадение;
3. Соответствие на начальные части;
4. Соответствие на минимум начальных частей;
5. Соответствие на 3 символа.
6. Отсутствие соответствия.

Населенные пункты выводятся вместе с координатами.

Также для системного контроля выводится файл "TMP".

Пример:

```
<<<<<<<<<<  FIND >>>>>>>
Населенные пункты с НЕотождествленным типом кода 3
  0 1286      >                       ПОДДУБНЫЙ            ? размер
  1 1782      > 1170230         01    ШАХОВСКОЕ            ? пробелы
  2 1812      > 1170            01    ЮДИНО                ? пробелы
Населенные пункты с НЕотождествленным районным центром  13
  0   33      > 1170404               АЛЕКСИН
  1  147      > 1170408               БОГОРОДИЦК
  2  416      > 1170412               ДОНСКОЙ
  3  462      > 1170406               ЕФРЕМОВ
  4  652      > 1170420               КИМОВСК
  5 1156      > 1170424               НОВОМОСКОВСК
  6 1169      > 1170412563            НОВОУГОЛЬНЫЙ
  7 1236      > 1170432554            ПЕРВОМАЙСКИЙ
  8 1418      > 1170412580            РУДНЕВ
  9 1669      > 1170401               ТУЛА
 10 1679      > 1170428               УЗЛОВАЯ
 11 1803      > 1170432               ЩЕКИНО
- 12 1812      > 1170      01          ЮДИНО
<<<<<<<  >>>>>>>
```

Примечание: наличие первым символом НЕ ' ', и НЕ цифры указывает, что данная строчка обрабатываться не будет. Наличие системных сообщений BS.REP допускается. Также допускается наличие инструкций для коррекции базы данных (см. 7.2).

> lbase 7

Предположим, что в результате данной процедуры выяснилась какая необходима коррекция для большинства пунктов однозначно.Для оставшихся пунктов, имеющих неоднозначное отождествление, предлагается обратитьс к базе данных и использовать полученные данные, в том числе и координаты. На данном опыте это было достаточно, при этом обычно использовался дополнительный код.

После выявления необходимых коррекций с помощью следующей команды изменения вносятся в базу данных.

11. Редактирование базы данных по файлу запроса [7.2]

Файл запроса "FIND" для данной команды отличается от предыдущей формы, только тем, что содержит инструкции для редактирования базы данных (как отмечалось, эти инструкции не препятствуют для предыдущей операции):

1) Замена кода. Символ '&';

2) Вставка в код номер района. Символ '@';

3) Наличие в начале любой строки символа '#', будет перекрывать редактирование базы данных.

Символы '&','@' должны располагаться после наименования пункта,но до символа '?',если он имеется в строке.

Редактированию подвергается база данных находящаяся в текущей директории.

Также для системного контроля выводится файл "TMP", в который выводятся контрольная информация о входных данных. Можно в качестве контроля вызвать команду Lbase.exe без ключа '$', тогда редактирование базы данных производиться не будет, но контрольный файл сформируется.

Пример (продолжение 7.1):

```
<<<<<<<<<<  FIND >>>>>>>
Населенные пункты с НЕотождествленным типом кода 3
 0 1286          >  ПОДДУБНЫЙ        &1170244808                  ? размер
 1 1782          > 1170230           01 ШАХОВСКОЕ        &1170230825    ? пробелы
 2 1812          > 1170              01 ЮДИНО      &1170212856    ? пробелы
Населенные пункты с НЕотождествленным районным центром  13
 0   33          > 1170404           АЛЕКСИН       @202
 1  147          > 1170408           БОГОРОДИЦК    @208
 2  416          > 1170412           ДОНСКОЙ       @244
 3  462          > 1170406           ЕФРЕМОВ       @220
 ..............
<<<<<<<  >>>>>>>>>
> lbase 7 !1 $
```

12. Переименование mdf-файлов [8.1]

В текущей директории ко всем *.Mdf-файлам, имеющim в наименовании 3 цифры, добавляется в наименование 4 символа из наименования базы данных *.Dbf, находящейся в текущей директории.

Пример:

```
> lbase 8 !1
```

13. Переименование txt-файлов [8.2]

В текущей директории все *.Txt-файлы, имеющие в наименовании 3 цифры и дополнительные символы, переименовываются по цифровой составляющей, с исключением дополнительных символов.

Примеры:

1. Тип переменование (COD.TXT,где COD - 3-х цифровой код района)

```
> lbase 8
```

2. Переменование по формату. Позиция COD указывается символом $

```
> lbase 8 :np$-2:a$:
```

Файлы np%%%-2.txt переменуются соответственно в a%%%.txt

14. Создание файла "region.obl" [9]

Файл "region.obl" создается на основе файла [NUMBER].TXT (см.12,13). Файл "REGION.OBL" содержащит вставку по текущему району для "REGION.TXT", и соответствует ему по форме (см. 5).

Пример:

> lbase 9

15. Удаление записей с совпадающими наименованиями и координатами [10]

Редактированию подвергается база данных находящаяся в текущей директории. Предполагается, что база данных отсортирована.

Удаляются записи определяемые, как, имеющие "абсолютное совпадение" (см. 4) и если расхождение по координатам по каждой из осей не превышает 0.01. Обычно эта команда должна работать корректно, так как предполагается, что база данных отсортирована с учетом населения (см. ОБЩИЕ ПОЛОЖЕНИЯ, 3) и удаляя пункты с меньшим количеством населения мы удалим менее значительные пункты, но все-таки рекомендуется вместо этой команды пользоваться возможностями оманды 11 и явно указавать сдвоенные пункты для удаления, учитывая омментарий и количество населения.

Рекомендуется эту команду после тестирования базы данных (4) удаления записей с недопустимым кодом (размер кода меньше 7), епосредственно указывая с помощью следующей команды 11.

Также для системного контроля выводится файл "TMP", в который ыводятся данные о данных коррекции. Можно в качестве контроля вызвать оманду Lbase.exe без ключа '$', тогда редактирование базы данных роизводиться не будет, но контрольный файл сформируется.

Пример:

> lbase 10 $

16. Редактирование базы данных по файлу запроса в общей форме [11]

Предполагается, файл запроса, указываемый в командной строке, формируется на основе листинга BS.REP, полученного в результате тестирования базы данных (4). Возможны следующие операции:

1) Удаление записи. В начало строки помещается символ '-'.

2) Редактирование полей. В начало строки помещается символ '@' и в любом месте строки вводится строка инструкций для коррекции в форме:

"..... &NumberField1_NewField1_&NumberField2_NewField2_ ... ",

где NumberField1, NumberField2, ... - номера редактируемых полей (0..),

NewField1, NewField2, ... - соответственно, их новое содержание.

Редактированию подвергается база данных находящаяся в текущей директории.

Также для системного контроля выводится файл "TMP", в который выводятся данные о данных коррекции. Можно в качестве контроля вызвать команду Lbase.exe без ключа '$', тогда редактирование базы данных производиться не будет, но контрольный файл сформируется.

Пример: Файл запроса:

```
<<<<<<<<<<  FILEEDIT.TXT >>>>>>>
Населенные пункты с НЕотождествленным типом кода 3
-     0 1286   >                        ПОДДУБНЫЙ            &1170244808     ? размер
@     1 1782   > 1170230        01       ШАХОВСКОЕ            &0_1170230825_          ? пробелы
@     2 1812   > 1170           01       ЮДИНО          &0_1170212856_          ? пробелы
!     2 1812   > 1170           01       ЮДИО                 !ЮДИНО
Населенные пункты с НЕотождествленным районным центром  13
@     0   33     > 1170404                ЛЕКСИН               &0_1170202404_
<<<<<<<  >>>>>>>>>
```

> lbase 11 $ FILEEDIT.TXT

17. Создание файла "obld.txt" [12]

Данный файл "obld.txt" содержит построчно: код района, координаты районного центра (отметим, он не содержит наименование районного центра).

Отметим, наличие файла "obl.txt", содержащего список районных центров области построчно: код района, наименование района, наименование районного центра (остальные строки должны начинаться символа '-' или быть пустыми).

Данная процедура и процедура следующая процедура 13 нужны только ля создания файла [NUMBER].TXT, содержащего список районных центров области в форме (построчно): код района, наименование района, наименование районного центра, координаты (остальные строки также олжны начинаться с символа '-' или быть пустыми), где NUMBER - код области. Файл [NUMBER].TXT используется в процедуре 9.

Осуществляется просмотр порайонных текстовых файлов (признак: наличие числа в составе имени *.Txt - файла), в текущей директории. Признак районного центра: цифра '9' в 32 позиции строки. Коды районов, у которых районный центр не характеризуется номером 9, выводятся в конце списка; обычно таких райнов абсолютное меншинство и для них требуется самостоятельно внести данные в файл "obld.txt", исходя из знания районных центров области "obl.txt" и непосредственно искомого района.

Обработка ведется только текстовых файлов имеющих форму 1, т.к. во 2-й форме не содержится административного кода (см. 17.2).

Пример:

```
> lbase 12
```

18. Создание файла [number].txt [13]

Для создания файла используются данные файлов "obl.txt", "obld.txt", а также для формирования наименования [NUMBER].TXT имя базы данных в текущей директории (см. 12,9,5).

Пример:

```
> lbase 13
```

19. Переименование кода района [14]

Редактированию подвергается база данных находящаяся в текущей директории и создается база данных TMP.DBF.

Коды изменяемого района и присваемаемого района вводятся из командной строки.

Может использоваться файл, содержащий построчно список кодов для переименования; файл должен иметь расширение *.ren.

Для системного контроля выводится файл "TMP14".

Примеры:

1. Заменить район 242 на 212

```
> lbase 14 $ 242 212
```

2. Провести замену по файлу

```
<<<<<<<<<<  1170.REN >>>>>>>
401 232 ТУЛА
404 202 АЛЕКСИН
408 208 Богородицк
412 226 ДОНСКОЙ
416 220 ЕФРЕМОВ
420 226 Кимовск
234 244 НОВОМОСКОВСК
424 244 НОВОМОСКОВСК
428 244 УЗЛОВАЯ
432 248 Щекино
<<<<<<<  >>>>>>>>>
> lbase 14 $ 1170.REN
```

20. Изменение размерности admcode с 11 на 13 [15]

Редактированию подвергается база данных находящаяся в текущей директории и создается база данных TMP.DBF, с измененной размерностью AdmCode. Коррекция осуществляется по принципу: для формирования нового содержания поля AdmCode берется 4 символа из наименования базы данных, и соответственно переносятся с 3 по 11 символы AdmCode источника.

Для системного контроля выводится файл "TMP".

Пример:

> lbase 15 $

21. Удаление (сохранение) записей с пропущенными или обнуленными полями [16]

Редактированию подвергается база данных находящаяся в текущей директории и создается база данных TMP.DBF.

В командной строке укавается номер поля координат (0..),с помощью ключа '#' (по умолч. 2).

Для опции "СОХРАНЕНИЕ" используется ключ "!1"

Для системного контроля выводится файл "TMP".

Примеры:

> lbase 16 $ #2 - удаление

> lbase 16 $!1 - сохранение

22. Ввод в базу данных записей из файла запроса [17.1]

Файл запроса "INSERT" содержит данные для ввода в базу данных в форме (построчно): код района (3 симв.), название населенного пункта, служебный символ '@', координаты.

Редактированию подвергается база данных находящаяся в текущей директории и создается база данных TMP.DBF.

Для системного контроля выводится файл "TMP17".

Пример:

```
<<<<<<<<<< TMP17 >>>>>>>
 203 АГАПОВКА              @  59.135   53.268
 206 АРГАЯШ                @  60.879   55.490
 212 БРЕДЫ           @  60.348   52.395
 255 УВЕЛЬСКИЙ             @  61.355   54.444
 244 НЯЗЕПЕТРОВСК          @  59.600   56.050
<<<<<<<<<< >>>>>>>
> lbase 17 $
```

23. Ввод в базу данных записей из текстовых файлов в директории [17.2]

Вводится информация из текстовых файлов, содержащих код района, и имеющих одну из двух принятых форм для районных файлов (отличительным признаком 1-й формы служит наличие "---" в начале строки или '-' в 44 позиции строки).

Редактированию подвергается база данных находящаяся в текущей директории и создается база данных TMP.DBF.

Для системного контроля выводится файл "TMP17".

Пример 1 (форма 1):

```
<<<<<<<<<< 209.TXT >>>>>>>
-------------------------------------------------------------------------
                    Признак   Координаты центра населенного пункта
-------------------------------------------------------------------------
 162 Сухая Атя              0  57.399 -  54.826       57 23 57 -  54 49 33
 163 Виляи                  0  57.308 -  54.781       57 18 30 -  54 46 49
 164 Квартал 49-й           0  57.250 -  54.844       57 14 59 -  54 50 37
 165 Новострой              0  57.202 -  54.834       57 12  8 -  54 50  3
<<<<<<<<<< >>>>>>>
```

Пример 2 (форма 2):

```
<<<<<<<<<< S-226.TXT >>>>>>>
```

```
НОВАЯ ДЕРЕВНЯ          60.815    55.735   60 48 55    55 44  4  65
БЕЛАПАХОВО               61.160    56.318   61  9 34     56 19  4  86
ДАВЫДОВО                 61.589    56.162   61 35 18    56  9 43 107
БОЛ. КЫЗЫЛОВА          61.653    56.007   61 39 12    56  0 23 121
<<<<<<<<<<  >>>>>>>
> lbase 17 !1 $
```

24. Установка пустых полей в базе данных по файлу запроса [18]

По файлу запроса [NUMBER].FLD, содержащий список контрольных полей, программа ставит символ '0' в позицию пропущенного поля.

Редактированию подвергается база данных [NUMBER].DBF, находящаяся в текущей директории и создается база данных TMP.DBF.

Для системного контроля выводится файл "TMP17".

Пример:

```
<<<<<<<<<< 1175.FLD >>>>>>>
POPUL1989
MEN1989
WOM1989
PROCNAC89
PROCN89
<<<<<<<<<<  >>>>>>>
> lbase 18 $
```

25. Сравнительное тестирование базы данных и текстовых файлов по коду района [19]

Тестированию подвергается база данных находящаяся в текущей директории.

Общие принципы: латинские символы заменяются соответствующими из кирилицы, приводятся к одному регистру, игнорируются пробелы и служебные символы, игнорируются части заключенные в скобки '(',')', игнорируются цифры.

При тестировании проводится следующая градация:

1.Абсолютное совпадение. Посимвольное полное совпадение, с учетом вышеприведенных принципов, с соответствием по номерам частей.

2.Вложенное совпадение. Посимвольное полное совпадение последней части, вложенное совпадение предыдущих частей. Соответствие по номерам частей. При совпадении координат или если координаты не используются, и только в этих случаях, обработка проводится по типу "абсолютного совпадения".

3.Соответствие на вхождение. Каждая часть из одного выражения, вложено входит в какую-то из частей другого.

4.Соответствие на начальные части. Совпадение первых частей.

5.Соответствие на минимум начальных частей. Совпадение на минимум из первых частей.

6.Соответствие на 3 символа

7.Соответствие по координатам

8.Отсутствует соответствие

Населенные пункты выводятся вместе с координатами,включая признак совпадения.

Основной вывод осуществляется в файл "TxtBs.Rep", в формате операций поиска (см. 21.1-3, 4).

Подробная информация содержится в файле "PRTCL19.REP".

Для системного контроля выводятся файл "TMP7".

Дополнительно, при наличие файла "FIND_COR"(см. 21.3), проводится коррекция его по номерам записей базы данных (по наименованиям и коду). Выводится новая версия данной формы "FIND_COR.NEW", причем в ней исключаюся коррекции текстовых файлов районов. Также выводится контрольный файл "FIND_COR.RES".

Дополнительные ключи:

```
#     - без использования координат.
!E    - загрузка всех данных из *.Dfb по районам.
!M    - групповые списки по району
```

!V - коррекция наименований на основе файла NONAME.WRN (см. 29)

Пример:

```
> lbase 19
```

26. Извлечение файлов типа пр.* из архивных файлов, с проходом по поддиректориям [20]

Программа осуществляет проход по поддиректориям, извлекает из архивных файлов (*.arj) файлы с именем "пр.*" и переименовывает их в файлы с начальной частью по наименованию поддиректории.

В качестве параметров передаются наименование диска обрабатываемой директории и рабочего диска (должны отличаться).

Пример:

```
> lbase 20 D: C:
```

27. Выбор в базе данных 10 ближайших пунтов по файлу запроса [21.1]

Файл запроса "FIND_DBF" содержит данные для поиска в форме пункта листинга по операции 4 "Населенные пункты с неотождествленным районным центром (с коорд.)".

Выходные данные будут выводится в файл "FIND_DBF.RES" в форме:

AdmCode,PlaceName,координаты и суммарное отклонение по координатам.

Предполагаются, что поля AdmCode, PlaceName, Xgrad, Ygrad расположены последовательно.

Данная операция позволяет определить по координатам - к какому примерно району может принадлежать неотождествленный населенный пункт.

Пример:

```
<<<<<<<<<<  FIND_DBF >>>>>>>
  0  231 > 1175208101    ЕМАНЖЕЛИНСК
 > DBF 61.321 54.749
  5  387 > 1175101431    КОРКИНО
 > DBF 61.403 54.909
  6  478 > 1175101438    МАГНИТОГОРСК
 > DBF 59.028 53.396
  9  879 > 1175101455    УСТЬ-КАТАВ
 > DBF 58.174 54.894
 11  968 > 1175101464    ЮЖНОУРАЛЬСК
 > DBF 61.263 54.441
<<<<<<<<<<  >>>>>>>
> lbase 21
```

28. Выбор в директории 10 ближайших пунтов по файлу запроса [21.2]

Файл запроса "FIND_DBF" содержит данные для поиска в форме пункта листинга по операции 4 "Населенные пункты с неотождествленным районным центром (с коорд.)" (см. 19).

Выходные данные будут выводится в файл "FIND_TXT.RES" в форме: AdmCode,PlaceName,координаты и суммарное отклонение по координатам.

Предполагаются, что поля AdmCode, PlaceName, Xgrad, Ygrad расположены последовательно.

Данная операция позволяет определить по координатам - к какому примерно району может принадлежать неотождествленный населенный пункт, в частности, определить лексикографические рассогласования и по коду района по обобщенному сравнительному листингу операции 19.

Пример:

```
<<<<<<<<<<  FIND_DBF >>>>>>>
  0  231 > 1175208101    ЕМАНЖЕЛИНСК
 > DBF 61.321 54.749
  5  387 > 1175101431    КОРКИНО
 > DBF 61.403 54.909
<<<<<<<<<<  >>>>>>>
> lbase 21 !1
```

29. Редактирование базы данных и текстовых файлов районов по файлу запроса [21.3]

Файл запроса "FIND_COR" содержит данные для редактирования в форме формируемой на основе листинга "TxtBs.Rep" операции 19, предваряя каждые 2 строки содержимых пунктов, требующих редактирования, строкой - инструкцией в допустмых формах:

>!*** - коррекция наименования в базе данных;

>>*** - коррекция кода в базе данных;

># - коррекция координат в базе данных;

>!*** - коррекция наименования в текстовом файле района.

># - коррекция координат в текстовом файле района (символ может также находится в строке координат).

Помимо коррекции также выводится отчетный файл о проведенном редактировании "CORRECT.REP".

Для системного контроля выводится файл "TMP7B", в который выводятся контрольная информация о входных данных. Можно в качестве контроля вызвать команду Lbase.exe без ключа '$', тогда редактирование данных производиться не будет, но контрольный файл сформируется. Также выводится системный файл "TMP".

Пример: <<<<<<<<<< FIND_COR >>>>>>>

```
>@252
0 985  □ 1175206840706                ЯНГИЮЛ
> TXT 61.280 55.593
>#
0 985  > 1175206840706        ЯНГИЮЛ
> TXT 63.280 55.593

>!КРУГЛОЕ
0 285  > 1175226840706        КРУГЛЦЕ
> TXT # 65.280 65.593

>#
0 985  > 1175206840706        ЯНГИЮЛ
> TXT 61.280 55.593

>!МЕДИАК
 20 526  > 1175252830705              МИДИАК
> TXT 61.039 55.300

>!МЕДИАК
  0   20 > 252141                    МИДИАН
> TXT 61.039 55.300 ¦  0.000 @1211
<<<<<<<<<< >>>>>>>
> lbase 21 !2 $
```

30. Сортировка и приведение текстовых файлов [22]

Данные в текстовых файлах сортируются и приводятся в соответствии с формой 2 (см. 17.2). Выходные файлы будут иметь расширение ".t".

Ключ '#': без использования координат.

Использование (примеры):

1. Обработка единичного файла:

```
> lbase 22 NP202.NXT
```

2. Обработка файлов в текущей директории:

```
> lbase 22
```

31. Сравнение текстового файла с текстовым файлом или со списком района из базы данных [23]

В результате обработки формируются следующие выходные файлы (при 2-х проходах):

1. TXTTXT1.REP,TXTTXT2.REP - сравнительная информация об расхождениях
2. TXTTXT1.,TXTTXT2. - полная сравнительная информация
3. TXTTXT1.EQ,TXTTXT2.EQ - списки по совпадению
4. TXTTXT1.DIF,TXTTXT2.DIF - списки по расхождению

а также (при 1-м проходе):

5. NON_TXT.WRN,NON_DBF.WRN - списки по расхождению
6. NONAME.WRN - сравнительные списки по расхождению,

причем используется следующая градация:

* Абсолютное совпадение
* Вложенное совпадение
* Отсутствует соответствие

При формировании файла NONAME.WRN производится составление соответствий, как для ввода данных от 1-го ко 2-му, аналогу района базы данных, т.е. один в один; и все незадействованные данные, в соответствии вышеприведенной градацией, выводятся в данный файл. Отметим, что информация о дублях, если есть совпадение по сравнению, выводится только в этот файл.

При производном ключе 123, также выводятся (при 2-х проходах):

7. TXTTXTF1.DIF,TXTTXTF2.DIF - списки по расхождению (абс.совп.).

Для системного контроля выводятся файл "TMP".

Использование (примеры):

1. Сравнение 2-х текстовых файлов:

> lbase 23 FILE1.TXT FILE2.TXT

2. Сравнение текстового файла со списком района из базы данных:

> lbase 23 FILE.TXT

32. Извлечение списка района из базы данных [24]

Ключ "#": без использования координат.

Ключ "!T": тип "taifun".

Ключ "!TT": тип "taifun",без пунктов с нулевыми координатами.

Ключ "!K": обработка по механизму согласования, для согласования дополнительных баз данных с основной. Механизм согласования состоит из двух процедур, в результате которых мы получаем связующие файлы EDITXY.EQ, EDITXY.WRN, по использованию аналогичные EDITXY.EQ, EDITXY.WRN (см. 29). В дальнейшем указанные файлы файлы для согласования должны приводиться после связанной с ними базы данных. Данный механизм может использоваться в командах 2.1-2.3.

Использование (примеры):

1. Извлечение из базы данных списка района по коду района:

> lbase 24 COD ,где COD - 3-х цифровой код района

2. Извлечение из базы данных всех списков района:

> lbase 24

3. Механизм согласования.

3.1 Извлечение из дополнительной базы данных всех списков района

> lbase 24 !K [*].DBF

3.2 Согласование с основной базой данных

> lbase 29 !K [MAIN].DBF

33. Приведение структуры базы данных по запросам rvs [25]

Используется для выяснения списка используемых полей в DsFields.idx или DsBase.def и уменьшения общего объема базы данных. При обработке файлы из списка для тестирования рассматриваются как бинарные и сравнение осуществляется на вхождение. Приведенная структура дополняется до стандартной структуры (см. ОБЩИЕ ПОЛОЖЕНИЯ, 1).

Ключ "#": при сравнении используется регистр.

Ключ "!": выводятся все поля.

По умолчанию, при сравнении не используется регистр и в выходной структуре базы данных содержатся только подтвержденные поля.

Пример:

> lbase 25 # ! DsFields.idx H:DsFields.idx DsBase.def

34. Сравнение текстового файла по директории [26]

Аналогично операции 23, но сравнение осуществляется последовательно со всеми текстовыми файлами в директории.

В результате обработки формируются следующие выходные файлы:

1. TxtTxt.Rep - совпадение наименований и координат
2. TxtTxtEq.Rep - совпадение только координат

Ключ "!": при сравнении всех тестовых файлов со всеми, сравнение осуществляется с вышестоящими по коду районами, т.е. в один проход.

Ключ "#": без использования координат.

Использование (примеры):

1. Сравнение текстового файла:

> lbase 26 COD ,где COD - 3-х цифровой код района

2. Общее сравнение по директории:

> lbase 26

35. Генерация формата dbf из формата обмена данных [27.1]

Требуется соединить в один файл структуру базы данных и текстовой файл записей и над этим файлом выполнить команду:

>lbase 27 DATA ,где дата - вышеуказанный файл.

36. Коррекция поля placename в базе данных [27.2]

Осуществляется замена латинских символов,удаление лишних пробелов, формируются сокращения для некоторых характерных слов.

Пример:

>lbase 27 #1 $

37. Пересортировка базы данных по образцу [27.3]

Предполагается, что количество записей в базах данных совпадает и они соответствуют друг другу.

Может использоваться файл содержащий подстановки кодов.

Пример:

1. Простая пересортировка.

> lbase 27 #2 DB1.DBF DB2.DBF $

2. Пересортировка с файлом подстановок кодов.

<<<<<<<<<< RECOD. >>>>>>>

1129405|1129214405

1129410|1129220410

<<<<<<<<<< >>>>>>>

> lbase 27 #2 DB1.DBF DB2.DBF RECOD $

38. Ввод координат в базу данных [27.4]

Координаты вводятся в базу данных на основе согласованного файла, имеющего форму файлов EDITXY.EQ,EDITXY.WRN функции 29.

Наличие непробельного символа в строке - пропуск данных по наименованию.

Пример:

> lbase 27 #3 EDITXY.EQ $

39. Ввод дополнительных пунктов в базу данных [27.5]

Данные по пунктам вводятся в базу данных на основе файла,имеющего форму файлов INSERT.NEW,INSERT.WRN функции 29.

Наличие непробельного символа в строке - пропуск данных по наименованию.

Пример:

> lbase 27 #4 INSERT.NEW $

40. Подготовка к коррекции базы данных и текстовых файлов [27.6]

Обработка списков *.COR или файла Find_Cor или произвольного файла File.Cor (в этом случае расширение *.Cor обязательно) с целью коррекции исходных данных перед последующим согласованием и внесением информации из текствовых файлов или из другой базы данных (см. 27.6).

В результате обработки формируются следующие выходные файлы:

Find_Res.New - согласованные собранные входные данные

Find_Res.Inp - собранные входные данные

Для системного контроля выводятся файл "TMP33".

Дополнительные ключи:

- без использования координат.

Примеры:

> lbase 27 #5

или

> lbase 27 #5 File.Cor

41. Коррекция базы данных и текстовых файлов [27.7]

Коррекция выполняется на основе согласованного файла Find_Cor, имеющего форму файла Find_Res.New функции 27.5.

Пример:

> lbase 27 #6 $

42. Генерация базы данных регионов [28.1]

На основе районных файлов, находящихся в текущей директории, формируется база данных.

Пример:

> lbase 28 NNNN $,где NNNN - 4-х цифровой код области.

43. Ввод в базу данных записей из текстовых файлов в директории [28.2]

Данные из районных файлов, находящихся в текущей директории, вводятся в базу данных.

Пример:

> lbase 28 $

44. Сравнительное тестирование базы данных и текстовых файлов (первичная обработка) [29]

В результате обработки формируются следующие выходные файлы:

1. EDITXY.EQ - для 27.3 (ввод координат,абс.совп.)
2. EDITXY.WRN - для 27.3 (ввод координат,влож.совп.)
3. INSERT.WRN - для 27.4 (ввод пунктов, альт.EDITXY.WRN)
4. INSERT.NEW - для 27.4 (ввод пунктов)
5. TXT_TXT.REP - сравнительная информация об расхождениях
6. TXTTXT.REP - полная сравнительная информация
7. TXT_TXT_.REP - вывод списков (совпадение,расхождение)
8. NON_TXT.WRN,NON_DBF.WRN - списки по расхождению (см.23)
9. NONAME.WRN - сравнительные списки (см.23)

В файл INSERT.NEW вносятся наимененования, ненашедшие связи по совпадению с базой данных, и несовпадающие между собой (т.е. из повторяющихся вводится одно).

Для системного контроля выводятся файл "TMP".
Ключ "!K": обработка по механизму согласования (cm. 24)
Пример:
> lbase 29

45. Создание базы данных из текстового формата [30]

Ключ "!": формирование входных строковых номеров.

A) Генерация, включая список полей. Пример:

```
<<<<<<<< File.txt  >>>>>>>>>

WARNING
PLACENAME
ID
# #                                  # * #
  Москва

<<<<<<<<  >>>>>>>>>
```

> lbase 30 $ # File.txt File.Dbf

A) Генерация, с использованием файла описания. Пример:

```
<<<<<<<< File.lst  >>>>>>>>>

Name of File:                  File.Dbf
Number of Fields:      ***
Number of Records:             ***
Header Size:                   ***
Record Size:                   ***

Field Information

 0 | WARNING          C 2 0
 1 | PLACENAME       C 50 0
 2 | ID                       C 4

<<<<<<<<  >>>>>>>>>

<<<<<<<< File.txt  >>>>>>>>>

  Москва

<<<<<<<<  >>>>>>>>>
```

> lbase 30 $ File.txt File.lst File.Dbf

46. Перевод базы данных в текстовой формат [31]

Данные выводятся в формате текстового файла для обмена. А также, предварительно, выводится форматная строка полей.
Формируется файл описания.

A) Генерация простой формы. Пример:

Список параметров равен 3.

> lbase 31 File.Dbf

B) Генерация отчета, с использованием файла структуры.

Список дополнительных опций:

1 - вывод символа: C_0='|';
2 - вывод символа: C_0='|';
3 - вывод символа: C_0=' ';
4 - вывод порядкового номера;

<<<<<<<< FileRep >>>>>>>>>
....

Field Information

```
 0 | *@4              C 5   0
 0 | *@1              C 1   0
 0 | AREA             N 13  6
 0 | *@2              C 1   0
 1 | PERIMETER    N 13  6
 0 | *@2              C 1   0
 2 | COM_             N 11  0
 0 | *@1              C 1   0
 3 | COM_ID           N 11  0
 0 | *@1              C 1   0
 4 | TYPE             N  3  0
 0 | *@3              C 10  6
 5 | REGION           C 24  0
 6 | CONT             N  8  4
 7 | DIED             N  8  4
 8 | ILLN             N  8  2
 9 | COLOR            N  2  0
```

> lbase 31 File.Dbf FileRep

<<<<<<<< >>>>>>>>>

47. Обработка txtbs.rep [32]

Обрабатка подготовленного TxtBs.Rep проводится с целью генерации файла Find_Cor.R32, имеющую форму Find_Cor, удовлетворяющую функции 19 (см. 19,21.2). Файл TxtBs.Rep может иметь форму любой функции, в т.ч. функции 33, связанной с групповой обработкой списков по району.

Управляющие символы:

1. '+' - признак правильного имени
2. '-' - признак для исправления имени
3. '!' - признак правильных координат
4. '?' - признак для исправления координат

Требуется, чтобы

1. группы по одному наименованию отделялись пустыми строками, как они и содержатся в исходном тексте.

2. управляющие символы находились, в соответствии с интересующей информацией, в одной строке; они должны последовательно вставляться в начало строки, в любой последовательности.

3. новая информация должна вводится с новой строки, с соответствующим управляющим символом в начале строки и с соответствующей информацией со следующей позиции в строке, без вставки пробела.

Дополнительные ключи:

#	- без использования координат
!M	- групповые списки по району

Примеры:

1. Сравнение эталонных списков с ДВ, с координатами

```
<<<<<<<< TxtBs.Rep  >>>>>>>>>

!36.069 53.562
+  307    1 >      2041106                                КВАСОВА
?  > DBF 35.069 54.562
     Соответствие по координатам
-?¦ 0    1 > 33.069 53.562 > 2041106                 КВАСОВО

+УШАКОВА
!36.069 53.562
-  307    1 >      2041106                                ШАКОВА
?  > DBF 35.069 54.562
     Соответствие по координатам
-?¦ 0    1 > 33.069 53.562 □ 2041106                    УШАКОВО
 ........

<<<<<<<<  >>>>>>>>>

> lbase 32
```

2. Сравнение эталонных списков с ДВ, без координат

```
<<<<<<<< TxtBs.Rep  >>>>>>>>>
-   26  165 >       2048287                 БОЛОТОВО
  > DBF
     Соответствие на минимум начальных частей
¦    0   27 >             > 204468           БОЛ.ЧЕРНЬ
     Соответствие на 3 символа
+¦  0   28 >             > 20448            БОЛОТОВА
¦    1   29 >             > 204366           БОЛХОВ
¦    2   30 >             > 204598           БОЛХОВО

+БУДЕННЫЙ
   31   82 >       2048137                 БУДЕНЫЙ
-  > DBF
     Соответствие на 3 символа
¦    0   35 □           > 204296           БУДЕННОВСКИЙ
¦    1   36 □           > 204593           БУДОЛБИНО
 ........

<<<<<<<<  >>>>>>>>>

> lbase 32 #
```

3. Групповое сравнение

```
<<<<<<<< TxtBs.Rep  >>>>>>>>>

!36.069 53.562
+  307    1 >      2041106                                КВАСОВА
```

```
?  > DBF 35.069 54.562
     Соответствие по координатам
-?¦ 0    1 □ 33.069 53.562 > 2041106                КВАСОВО              "U204.TXT"

+УШАКОВА
!36.069 53.562
-  307    1 >       2041106                          ШАКОВА
?  > DBF 35.069 54.562
     Соответствие по координатам
-?¦ 0    1 > 33.069 53.562 > 2041106       УШАКОВО              "U204.TXT"
 ........

<<<<<<<<  >>>>>>>>>

> lbase 32 !M
```

48. Групповое сравнение по району [33]

Программа формирует узлы имен, разбивая на 3 группы по совпадению ("абсолютное совпадение", "вложенное совпадение", "несовпадение") в 3 прохода:

1. Абсолютное совпадение
2. Вложенное совпадение, присоединенное к абсолютному совпадению (т.е. входят в абсолютное совпадение).
3. Вложенное совпадение.

Остаток войдет в "несовпадение".

Группы по совпадению разбиваются на подгруппы по совпадению координат и единичные неподтвержденные элементы. Данные подгруппы характеризуются признаком - содержат ли элемент из контрольного списка (по умолчанию, контрольным списком объявляется первый по сортировке, не включая *.dbf, но из командной строки, можно указать любой, в т.ч. и *.dbf).

В результате обработки формируются следующие выходные файлы:

1. TXTBS.REP - сравнительная информация об расхождениях
2. PRTCL19.REP - подробная сравнительная информация;
3. LIST33.REP - структура обобщенных списков, общая справка.

NNN.EQ - обобщенный список района NNN, имеющий следующую градацию:

1. Общая справка
2. Абсолютное совпадение, включая координаты (Ctrl.txt) В подгруппе имеется элемент из контрольного списка.
3. Доп. Абсолютное совпадение, включая координаты (Other) Элементы подгруппы не входят в контрольный список, но есть подгруппа в группе, соответствующее типу 2.
4. Абсолютное совпадение, включая координаты (Other) Элементы подгруппы не входят в контрольный список, и нет подгруппы в группе, соответствующей типу 2.
5. Вложенное совпадение, включая координаты (Ctrl.txt)
6. Доп. Вложенное совпадение, включая координаты (Other)
7. Вложенное совпадение, включая координаты (Other)
8. Остаток. Абсолютное совпадение, включая координаты (Ctrl.txt)

Единичный элемент из узла, входящий в контрольный список (т.е. неподтвержденный по координатам), но по данному узлу есть подгруппа, соответствующая типу 2.

9. Остаток. Абсолютное совпадение, включая координаты (Other)
10. Остаток. Вложенное совпадение, включая координаты (Ctrl.txt)
11. Остаток. Вложенное совпадение, включая координаты (Other)
12. Абсолютное совпадение, НЕвключая координаты (Ctrl.txt)

Единичный элемент из узла, входящий в контрольный список (т.е. неподтвержденный по координатам), и по данному узлу нет подгрупп.

13. Абсолютное совпадение, НЕвключая координаты (Other)
14. Вложенное совпадение, НЕвключая координаты (Ctrl.txt)
15. Вложенное совпадение, НЕвключая координаты (Other)

16. Отсутствует соответствие (Ctrl.txt)

Элементы из контольного списка, невошедшие в списки по совпадению.

17. Other. Списки по отсутствию соответствия

Элементы из остальных списков, невошедшие в списки по совпадению, располагаются последовательно.

Принципы:

1. Символом '¦' отмечена наиболее важная по типу информация.

2. Элементы выводятся по подгруппам и по одиночным элементам. В выводе элементов из подгрупп, выводятся - вся элементы, входящие в контрольный список, либо по одному в альтернативном случае.

3. Если элементы выводятся из одного узла, то последующие элементы, в соответствующих позициях, отмечаются символами:

"+" - вывод из различных подгрупп

"++" - вывод из одной подгруппы (для контрольного списка)

Использование(примеры):

1. Контрольный список определяется по умолчанию.

> lbase 33 COD ,где COD - 3-х цифровой код района

2. С указанием контрольного списка (текстовой файл FILE.TXT).

> lbase 33 COD FILE.TXT

3. С указанием контрольного списка (по базе данных BASE.DBF).

> lbase 33 DBF

или

> lbase 33 BASE.DBF

49. Бинарный поиск в файле (сист) [34]

Предмет поиска и обработка результатов поиска задается в тексте программы программистом.

Использование:

> lbase 34 FILE

50. Добавление записей в базу данных из дополнительной базы данных [35]

Переносятся первые 9 полей.

Использование:

> lbase 35 BASE_IN.DBF BASE_OUT.DBF NBEG NKOL

где BASE_IN.DBF - база данных - приемник,
BASE_OUT.DBF - база данных - источник,
NBEG - номер начальной записи, или "BEG", или "END".
NKOL - количество записей.

51. Регенерация region.txt по базе данных [36]

Отсутствующие или нулевые координаты в файле Region.txt тестируются на наличие в базе данных и при наличие - данные вносятся в выходную версию "region.new".

Использование:

> lbase 36

52. Генерация *.ren по базе данных [37]

Данные с административным кодом более 4000000000000 (REGASS)
выводятся из базы данных в файл "*.ren".

> lbase 37

53. Специальная сортировка по коду (сист) [38]

Процедура предназначается для сортировки баз данных дополнительной информации извлеченных из системы " REGASS".

> lbase $ 38

54. Извлечение доп. информации из базы данных (REGASS) [40]

A) Извлечение по умолчанию (цезий в почве, AVECS1992P):
> lbase $ 40

B) Извлечение по коду:
> lbase $ 40 File.DBF NumCod
где File.DBF - исходная база данных,
NumCod - допустимый код (REGASS).

C) Извлечение по коду, с учетом границы (анал. B):
> lbase $ 40 File.DBF NumCod Lev
где Lev - нижняя граница

55. Ввод в базу данных доп. информации (REGASS) [41]

Данные вводятся в базу данных по 9 полю (см. пример, поле AVECS1992P) из базы данных извлеченной по процедуре 40. Указанные базы данных используют коды типа REGASS.

Пример:

<<<<<<<< ***.lst >>>>>>>>>

Name of File: ***.DBF
Number of Fields: ***
Number of Records: ***
Header Size: ***
Record Size: ***

Field Information

0 | ADMCODE C 13 0
1 | PLACENAME C 35 0
2 | XGRAD N 6 3
3 | YGRAD N 6 3
4 | ADEKZNACH C 37 0
5 | CTRLFLD C 2 0
6 | POPUL1989 N 7 0
7 | MEN1989 N 7 0
8 | WOM1989 N 7 0
9 | AVECS1992P N 6 2
.....

<<<<<<<< >>>>>>>>>

> lbase $ 41 FileMain.DBF FileAdd.DBF

2. Список процедур программы Lbaset (REGASS)

1. Генерация базы данных населенных пунктов (без населения)

Пример (извлечение из базы данных по коду 11NN):
> lbaset 1 11NN,
где 11NN - номер кода области.

2. Генерации базы данных дополнительной информации

> lbaset 1 11NN !+D
где D - указатель диска ('C','D','E','F','G'...).

3. Генерация базы данных населенных пунктов

Пример (извлечение из базы данных по коду 11NN):
> lbaset 2 11NN
или
> lbaset 11NN

4. Формирование списка областей

Список областей помещается в файл "REGION.COD".
Пример:
> lbaset 5

3. Приложение

1. Общие положения

1. Для работы с базой данных с помощью программы LBASE рекомендуется привести начальную структуру полей базы данных с помощью команд 1.1, 2 к принятой стандартной форме:

1. ADMCODE	C 13 0	- код объекта
2. PLACENAME	C 35 0	- наименование
3. XGRAD	N 6 3	- координаты
4. YGRAD	N 6 3	- ...
5. ADEKZNACH	C 37 0	- комментарий
6. CTRLFLD	C 2 0	- зарезервированное поле
7. POPUL1989	N 7 0	- общее население
8. MEN1989	N 7 0	- мужское население (не обяз.)
9. WOM1989	N 7 0	- женское население (не обяз.)
10. AVECS1992P	N 6 2	- загрязнение по цезию (не обяз.)

2. Принципы разработки обычно позволяют хранить в качестве необходимых данных (базы данных, списки) только исходную информацию, выходную информацию и промежуточные связующие звенья, непривязанные к конкретным исходным данным и позволяющие автоматизировано, как восстатовить, так и получить на основе новой отредактированной версии исходных данных новую версию выходных данных, с учетом раннее проведенной коррекции.

3. В программе в качестве принципов сортировки применяется приведение данных к общему виду, учитывая принятые соглашения (см. 19), а также учитывается в качестве дополнительного параметра: население(для *.dbf), тип населенного пункта (для текстовых списков).

4. Для текстовых списков используются 2 принятые формата хранения как равноправные (см. 17.2).

5. Для коррекции данных в командной строке требуется приводить символ '$', в противном случае, как правило, формируются выходные отчеты, но коррекция не выполняется.

6. Список ключей общего применения:

#	- без использования координат
$	- редактирование данных
!T	- база данных извлечена из системы REGASS
!Ncod	- обрабатывается извлечение из базы данных по 4-х цифровому коду области cod.

!O - отладочный режим (биты):
1 - дополнительная печать
2 - единичный проход
!K - обработка по механизму согласования (см. 24)
!I - используется механизм индексирования
!M - групповые списки по району
!= - в TxtBs.Rep включается "Наличие Совпадение по наименованию"
!E - загрузка всех данных из *.Dfb по районам. По умолчанию, для команд сравнения данных базы данных и текстовых файлов используется перезагрузка (для экономии динамической памяти).

*.COR - при работе с коррекцией, файл с данным расширением рассматривается в качестве данных по коррекции.

2 Версия первоначальной подготовки данных

2.1. Формирование region.txt

1. 12 Формирование координат районных центров ("obld.txt").
2. 13 Формирование обобщенного файла описаний районных центров ("NNNN.TXT, где NNNN - 4-х цифровой код области).
3. 9 Формирование "Region.Obl", по структуре соответствующего "Region.Txt".

В процессе обработке по вышеуказанным командам требуется неидентифицированные величины вносить непосредственно.

2.2. Тестирование *.mdf - файлов

Тестирование проводится с помощью операции 5, с использование файла "Region.Txt".

2.3. Подготовка списков районов. Редактирование списков районов и базы данных. Ввод координат в базу данных.

Используя данные разных источников требуется подготовить эталонные списки по области. Рекомендуется, чтобы основная часть наименования эталонного списка района состояла только из кода района. В процессе составления списков возможна коррекция как самих списков так и баз данных.

Для подготовки данных предлагается использовать следующие команды:

8.2 - переменование текстовых файлов
19 - сравнение текстовых файлов с базой данных (с помощью 32 и 21.2 или 27.5(без координат), редактирование эталонных списков и БД)
22 - приведение текстового файла
23 - сравнение 2-текстовых файлов или с базой данных
24 - извлечение списка из базы данных
26 - сравнение текстового файла с директорией
27.5 - работа с файлами коррекции *.COR (с помощью 27.6 редактирование группы списков и БД)
28.1 - генерация базы данных из списков районов
28.2 - ввод в базу данных из текстовых файлов
33 - групповое сравнение файлов по району (с помощью 32 и 21.2 редактирование группы списков и БД)

Для ввода координат в базу данных предлагается использовать следующие команды в указанной последовательности:

29 - предварительная обработка
27.3 - введение координат. Можно без внимания ввести данные по файлу EDITXY.EQ и внимательно проследить за данными EDITXY.WRN, и в случае несогласия с предложениеем по наименованию - вставить в соответствующей строке EDITXY.WRN символ '-' и заменить в соответствующей строке INSERT.WRN символ '•' на пробел и использовать данный файл в следующей процедуре.

27.4 - введение новых пунктов в базу данных. Можно без внимания ввести данные по файлу INSERT.NEW и, если были внесены коррекции в INSERT.WRN в связи с EDITXY.WRN (см. 27.3), то ввести данные и из INSERT.WRN.

2.4. Подготовка базы данных

Для подготовки базы данных предлагается использовать следующую последовательность команд:

25 Приведение базы данных по запросам RVS.

16 Удаление записей с пропущенными координатами.

6 Сортировка

4 Тестирование. Выявляются расхождения с данными файла REGION.TXT (исправления вносятся в редакторе в файл REGION.TXT), записи с неотождествленными кодами и сдвоенными пунктами (удаляются с помощью следующей команды 11).

11 Удаление отмеченных при тестировании записей.

Дальнейшая подготовка проводится непосредственно в пакете RVS заключается в генерации индексов и средних значений.

3. Версия подготовки данных на основе базы банных REGASS BD

1. Lbase 14 $ *.REN - для базы данных типа "REGASS";
2. lbase 1119 $ *.COR !V [возм. !==, возм. !*, - зависят от типа*.COR];
3. lbase 21 $!2 [NEW] *.COR - коррекция данных;
4. lbase 24 !T !K - ! вывод списков в формате связи;
5. lbase 29 !K - генерация файлов связи;
6. lbase 2 $ ... *.dbf *.EQ ... - тестирование и генерация данных для ввода; (+%)
7. lbase 2 $ ... *.dbf *.CMP - ввод данных [CMPDBFL.CMP]

[8. lbase 2 $!O !Q ... INF_11_14 #_11_14, где 10/14: @2. - контрольный режим (?)]

4. Классификация процедур программы Lbase

Классификация проводится на основе кодов процедур, которые приводятся в названии разделов главы «2 Список процедур программы Lbase» в квадратных скобках.

Специализированная подготовка данных для RVC:

Подготовка Region.txt:	5, 9, 12, 13, 36
Поддержка MDF-файлов:	5, 8.1
Поддержка списковых файлов:	8.2, 17.2, 20, 22-24, 26, 27.5, 27.6, 31.1

Общая поддержка работы с базой данных:

Информация:	1.1, 1.2
Генерация:	27
Коррекция:	2.1-2.6, 6, 35

Специализированная поддержка работы с базой данных:

Информация:	4, 21.1, 21.2, 31.2
Сравнение:	7.1, 19, 26, 29, 33
Генерация:	28.1, 30
Коррекция:	7.2, 10, 11-18, 21.3, 25, 27.1-27.6, 28.2, 32

Поддержка работы с базой данных REGASS DB:

Генерация:	40, 41