Криворожский государственный педагогический институт
Кафедра информатики и прикладной математики

Курсовая работа

на тему

Использование СУБД FoxPro 2.0 для построения и обслуживания БД "Расписание занятий"

Выполнил
студент группы МИ-93-2
Семериков С.А.

Проверил:
Ст. преп. Лисина Л.А.

Кривой Рог 1997

Вступление

Программа составления расписания занятий - не лучшая иллюстрация к системам управления базами данных, обычно их иллюстрируют задачами складского учета, учета автомобилей в автоинспекции, больных в стоматологии или служащих на предприятии. Наш выбор обусловлен с одной стороны приобретаемой вами специальностью школьного учителя, для которого маловероятно занятие учетными задачами приведенных классов, а с другой стороны тем, что составление расписания относится к "неудобным" задачам например в части формирования экранного отображения базы данных - дело в том, что большинство практических задач отличаются от стандартных иллюстраций именно "неудобностью" и мы хотели использовать неклассический вариант для освоения некоторых методов преодоления возникающих трудностей. Третий фактор - полезность программы в вашей будущей работе; наряду с созданием ряда компьютерных обучающих программ вам может пригодиться и эта программа - разумеется после соответствующей доводки и надежной защиты от неправильных действий пользователя.

Будем считать, что перед нами стоит задача создать компьютерную программу, позволяющую составлять, изменять и оптимизировать по некоторым критериям расписание занятий на один семестр в высшем учебном заведении, включающем некоторое множество факультетов, каждый из которых ведет обучение по нескольким специальностям в течение ряда семестров. Каждая специальность на отдельно взятом курсе обучения может содержать несколько групп обучаемых, а каждая группа может разбиваться на подгруппы для проведения например лабораторных занятий в специализированных лабораториях.

Составить расписание для конкретного факультета - значит определить для каждой подгруппы на каждую 2-часовку (или более - например 4-часовое занятие) каждого дня недели изучаемый предмет, проводящего занятие преподавателя и соответствующую аудиторию. При этом и преподаватель и аудитория в соответствующей 2-часовке не могут фигурировать в расписании занятий других факультетов и любых подразделений обучаемых - не должно быть так называемых "накладок". Оптимизация расписания состоит в исключении "дыр" в занятиях студентов и в загрузке преподавателей, в размещении предметов по 2-часовкам каждого дня занятий в соответствии с категорией сложности предметов - более трудные должны идти раньше легких, в расстановке занятий в разных корпусах через большую перемену для обеспечения времени на перемещение студентов и преподавателей и пр.

Программа должна предоставлять серию необходимых услуг по работе с готовым расписанием - например по запросам пользователя выдавать на экран и принтер расписание занятий любой заданной группы учащихся на четную или нечетную неделю, такое же расписание для любого заданного в запросе преподавателя или всех преподавателей кафедры, недельную аудиторную нагрузку преподавателя или группы студентов и пр.

Совершенно очевидно, что нам предстоит обрабатывать некоторый

файл записей, каждая из которых содержит "ведущие" или ключевые поля, одни из которых определяют "столбец" в табличной форме расписания:

-курс;
-специальность;
-группа;
-подгруппа;

другие поля определяют строку в таблице расписания занятий:

-день недели;
-номер 2-часовки (пары, ленты и пр);

а некоторые поля являются как бы зависимыми, ведомыми:

-предмет;
-преподаватель;
-аудитория;

Еще одна группа полей может носить общий характер:
-факультет;
-тип недели (четная - нечетная, числитель- знаменатель и пр)

Таким образом мы будем работать с файлом табличной структуры и должны обеспечить возможность выполнения в этом файле ряда стереотипных действий - создание файлов и заполнение записей, их корректировку при необходимости, поиск записей с заданным содержимым некоторых полей, обработку заданных последовательностей записей по некоторому алгоритму и пр.

Очевидно, что мы могли бы обойтись в нашем случае одним файлом с перечисленными выше полями, но в этом случае имела бы место нерациональеая избыточность по расходу памяти - например названия факультетов, специальностей, фамилии преподавателей и многое другое многократно повторялись бы в различных записях в неэкономичной текстовой форме. Поэтому такие повторяющиеся многократно сведения обычно размещают в отдельных файлах списков совместно с их числовыми кодами, а в основном файле базы данных оперируют только этими более экономичными в части расходования памяти кодами. Таким образом база данных представляет собой обычно совокупность многих взаимосвязанных файлов.

Опыт применения ЭВМ для построения прикладных систем обработки данных показывает, что более эффективным инструментом здесь могут быть не универсальные алгоритмические языки высокого уровня, а специализированные (проблемно - ориентированные) системы и встроенные в них входные языки . Такие средства обычно включают в состав систем управления базами данных (СУБД), но они могут существовать и отдельно. СУБД дают возможность пользователям осуществлять непосредственное управление данными, а программистам быстро разрабатывать более совершенные программные средства их обработки.

По способу установления связей между данными различают реляционную, иерархическую и сетевую модели.

Реляционная модель является простейшей и наиболее привычной формой представления данных в виде таблицы. В теории множеств таблице соответствует термин отношение (relation), который и дал название модели. Для нее имеется развитый математический аппарат - реляционное исчисление и реляционная алгебра, где для баз данных (отношений) определены такие хорошо известные теоретико - множественные операции, как объединение, вычитание, пересечение и т.д.

Достоинством реляционной модели является сравнительная простота инструментальных средств ее поддержки, недостатком - жесткость структуры данных (невозможность, например, задания строк таблицы произвольной длины, отсутствие списковых полей, отсутствие прямой поддержки нестандартных типов данных - звука, изображения) и зависимость скорости ее работы от размера базы данных. Для многих операций, определенных в такой модели, может оказаться просмотр всей базы данных.

Иерархическая и сетевая модели предполагают наличие связей между данными, имеющими какой-либо общий признак. В иерархической модели такие связи могут быть отражены в виде дерева - графа, где возможны только односторонние связи от старших вершин к младшим. Это облегчает доступ к необходимой информации, но только если все возможные запросы отражены в структуре дерева. Никакие иные запроосы удовлетворены быть не могут.

Указанные недостаток снят в сетевой модели, где возможны связи "всех со всеми". Использование ерархческой сетевой модел ускоряет доступ к информации в базе данных. Но поскольку каждый элемент данных должен содержать ссылки на некоторые другие элементы, требуются значительные ресурсы как дисковой, так и основной памяти ЭВМ. Нехватка ОЗУ, естественно, снижает скорость обработки данных. Кроме того,для таких моделей характерна сложность реализации СУБД.

FoxPro 2.0 фирмы Fox Holdings Software, ныне приобретенная Microsoft Corp., претендует на то, чтобы быть реляционной, однако до этого ей еще далеко - из 300 правил Кодда для реляционных систем в ней выполняется едва ли 40, однако этот пакет достаточно неприхотлив к системным ресурсам (для работы достаточно обычного "Поиска-2"), имеет неплохой командный режим, позволяющий выполнять все операции с базой без программирования вообще и предоставляет в Ваше распоряжение Windows - подобный событийно - ориентированный интерфейс.

*
*   Полный текст программы "Расписание занятий"
*   ==
*
* Как показывает практика, в СУБД самое важное - это продуманные
*структуры данных, поэтому проектирование системы для
*обслуживания расписания мы начнем с составления структур баз
*данных. Следующим этапом будет заполнение этих баз, и затем -
*самая нужная часть, ради которой делается львиная доля работы:
*обслуживание базы данных, под которым мы будем понимать
*удовлетворение различных запросов к ней.
* Вспомним, как выглядит обычное расписание. На большом листе
*чертится или отпечатывется большое количество клеточек. Строки в
*расписании соответствуют номеру пары в какой-то из дней недели,
*столбцы определяют соответственно потоки, группы и подгруппы. А
*в клеточке мы видим название предмета, имя преподавателя и номер
*аудитории. Таким образом, для того, чтобы однозначно определить
*эту клеточку на стене в базе данных, нам необходимо пять полей:
*группа, пара, предмет, аудитория и преподаватель.
* Естественно, заносить эти данные в базу так как это делает
*диспетчер, по меньшей мере нерационально - может быть, ей и приятно
*раз 15 написать чьи-то реквизиты, но я бы предпочел обозначить их
*каким нибудь-кодом и заносить в расписание не строку "Великий
*инквизитор", а какое-то однозначно определяемое число, и так во
*всем. Конечно, где-то придется держать список пар типа код -
*преподаватель (а может и не пар) - в другой базе, к примеру.
* Итак, в базе расписания мы введем 5 числовых полей, в которые
*будем заносить коды групп, пар, предметов, аудиторий и
*преподавателей.
* Код группы мы будем формировать следующим образом :
* код_группы=код_потока*100+номер_группы*10+номер_подгруппы.
*При этом предполагается, что групп на потоке не может быть
*больше 9 (в противном случае код теряет смысл или, что еще
*хуже, приобретает неверный), а код потока формируется в
*отдельной базе потоков.
* Чем удобен именно такой код? Легкостью в поиске - если номер
*группы в коде равен нулю, то эта пара соответствует лекции, если
*номер группы ненулевой, а номер подгруппы нулевой, то это
*практичка, иначе - лабораторка. Следовательно, общий код мы
*никогда не будем делать нулевым, и 0 в коде рассматривать как
*ошибку.
* Обозначив числитель через 1, знаменатель - через 2 и
*пронумеровав дни недели от 1 до 5, мы получим удобный код пары:
* код=номер_недели*100+номер_дня*10+номер_пары
* Коды предметов, аудиторий и преподавателей мы будем
*формировать в соответствующих базах.
* База данных по предмету будет, естественно, содержать код и
*название предмета; кроме того, для удобства возложим на нашу
*программу обязанность старост - подсчитывать запланированное и
*реальное число часов, в неделю и общее. Дополнительно введем
*признак связи предмета с аудиторией, необходимый для того,
*чтобы, скажем, лабораторные работы по химии проводились в хим.
*лаборатории, а не в спортзале.
* Аудиторию мы будем характеризовать номером корпуса, номером
*аудитории, наличием какой-либо специализации (например,
*компьютерный класс) и емкостью - поток, группа, подгруппа.

```
*Некоторые аудитории настолько специфичны, что этих данных для
*кодирования недостаточно; не каждый догадается, что аудитория
*номер 1 в шестом корпусе - это мастерская ОТД, поэтому введем
*необязательное (то есть memo) поле, в котором и прокомментируем
*эти особенности.
*  Поток традиционно характеризуется специальностью - кодом,
*номером курса и количеством групп. У потоков филфака есть
*интересная особенность: часть групп занимается с часу, часть - с
*7.30 и т.д., причем те, кто занимаются с часу, считают свою пару
*первой, хотя для нас она - третья. Для таких вот экзотических
*специальностей введем особое поле смещения, показывающее, на
*сколько пар они отстали от жизни.
*  Каждый преподаватель имеет имя, звание, приписан к какой-то
*кафедре; кроме этого, в институте он занимает какую-то
*должность, что тоже надо учитывать при составлении расписания
*  Факультет, специальность и кафедру можно охарактеризовать
*кодом и именем.


*  Итак, очистим экран и память, сделаем необходимые установки и
*выведем в системном окошке сообщение:

wait window nowait [Начальная инициализация]
clear all
set talk off
set deleted on
set resource on
set date british
clear


*  В различных рабочих областях откроем необходимые нам при
*работе базы данных вместе со своими структурными индексными
*файлами, индицируя каждое действие соотвествующим сообщением на
*экране. Если файл базы данных не существует, то мы его создаем
*средствами SQL, попутно индексируя необходимые нам поля.

select a&&
*Вспомогательный файл предметов
wait window nowait [Открытие базы предметов]
if !file('predmet.dbf')
  CREATE TABLE predmet (code N(4,0),name C(65),link L,;
                        chas_week N(3,0),chas_all N(4,0))
  index on code tag code
  index on name tag name
  index on link tag link
  index on chas_week tag chas_week
  index on chas_all tag chas_all
endif
use predmet index predmet

select b&&&
wait window nowait [Открытие базы аудиторий]
*Вспомогательный файл аудиторий
if !file('auditor.dbf')
  create table auditor (code n(4,0),korpus n(2,0),aud n(3,0),;
```

```
                          emkost n(3,0),special L,comment M)
  index on code tag code
  index on korpus tag korpus
  index on aud tag aud
  index on emkost tag emkost
  index on special tag special
endif
use auditor index auditor

select d&&
wait window nowait [Открытие базы потоков]
if !file("potok.dbf")
  create table potok (code n(5,0),special n(3,0),;
                      kurs n(1,0),delta n(2,0),grpcount n(2,0))
  index on code tag code
  index on special tag special
  index on kurs tag kurs
  index on delta tag delta
  index on grpcount tag grpcount
endif
use potok index potok

select c&&
wait window nowait [Открытие базы преподавателей]
*Вспомогательный файл преподавателей
if !file("prepod.dbf")
  CREATE TABLE prepod (code N(4,0), kafedra N(3,0),;
                       fio C(65), rank N(2,0), zvan N(2,0))
  index on code tag code
  index on kafedra tag kafedra
  index on fio tag fio
  index on rank tag rank
  index on zvan tag zvan
endif
use prepod index prepod


select e
* Эта область пока свободна -м.б., под что-то временное


select f&&
wait window nowait [Открытие базы факультетов]
*Вспомогательный файл факультетов - список их
if !file('fuc.dbf')
  CREATE TABLE fuc (name C(65),code N(3,0))
  index on name tag name
  index on code tag code
endif
use   fuc   index   fuc


select g&&
wait window nowait [Открытие базы специальностей]
*Вспомогательный файл специальностей - список их
if !file('spec.dbf')
```

```
  CREATE TABLE spec (name C(65),code N(3,0),fac N(3,0))
  index on name tag name
  index on code tag code
endif
use   spec  index spec


select h&&
wait window nowait [Открытие базы кафедр]
*Вспомогательный файл кафедр - список их
if !file('kuf.dbf')
  CREATE TABLE kuf (name C(65),code N(3,0))
  index on name tag name
  index on code tag code
endif
use kuf     index kuf


select i
wait window nowait [Инициализация расписания]
*Главный файл расписания
if !file('raspisan.dbf')
  CREATE TABLE raspisan (gruppa N(5,0), para N(3,0),;
                         predmet N(4,0), audit N(4,0),;
                         prepod N(4,0))
  index on gruppa tag gruppa
  index on para tag para
  index on predmet tag predmet
  index on audit tag audit
  index on prepod tag prepod
endif
use raspisan


select j


*  Следующий фрагмент программы, предлагаемый Вашему вниманию,
*объявляет несколько глобальных массивов и переменных, проводя
*заполнение части их:

public dimension list_pred(50,2),WasChanged(10),;
                 tempos(10),list_spec(50,2),list_kuf(50,2),;
                 list_prep(50,2),days_arr(5),week_arr(2),;
                 list_fac(50,2),list_aud(50,2),;
                 list_pot(50,2),marazm(30,3)
public curaud,curprep,curkuf,curfac,curspec,curpot,;
       curpred,curgrp,curpg,first,pottxt,factxt,;
       spectxt,predtxt,preptxt,audtxt,kuftxt

  days_arr(1)=[Понедельник]
  days_arr(2)=[Вторник    ]
  days_arr(3)=[Среда      ]
  days_arr(4)=[Четверг    ]
  days_arr(5)=[Пятница    ]
  week_arr(1)='Числитель  '
```

```
  week_arr(2)='Знаменатель'

IsQuitFlag=.f.
first=.t.
STORE 0 TO curaud,curgrp,curprep,curfac,curspec,curpot,curpred
curkuf=0
curpg=0
      tempos(1)=[Первая]
      tempos(2)=[Вторая]
      tempos(3)='Третья'
      tempos(4)=[Четвертая]
      tempos(5)=[Пятая]
      tempos(6)=[Шестая]
      tempos(7)=[Седьмая]
      tempos(8)=[Восьмая]
      tempos(9)=[Девятая]

*  Рассмотрим подробнее некоторые соглашения об именах.
*  Если в начале массива стоит слово list_, это означает, что
*данный массив будет использоаться для хранения некоторого
*списка. Если переменная начинается с cur - то в ней будет
*содержаться некоторый выбор (обычно код) или 0 в случае его
*отсутствия. Переменная, заканчивающаяся на txt, содержит
*строку, соответствующую этому коду.
*  Конкретный вид данных определяется фразами
*
*  aud  - аудитория
*  prep - преподаватель
*  kuf  - кафедра
*  fac  - факультет
*  spec - специальность
*  pot  - поток
*  pred - предмет
*  grp  - группа
*  pg   - подгруппа
*
*  Кроме того, есть переменные и массивы, не укладывающиеся в эту
*схему :
*
*  WasChanged(10) - каждый элемент этого массива отражает
*                   состояние базы данных в рабочей области с
*                   соответствующим номером. Если в базе данных в
*                   области <number> что-то изменяется, в элемент
*                   WasChanged(<number>)=.T. . В конце программы
*                   эта информация используется для определения
*                   того, какую базу данных необходимо упаковать.
*
*  tempos(10)     - как можно заметить из действий,
*                   производящихся над элементами этого массива,
*                   в нем храняться числительные от 1 до 9 -
*                   номера групп в символьном формате. Они
*                   понадобяться для "красивого вывода" номера
*                   группы.
*
*  days_arr(5)    - содержит названия учебных дней недели, а
*
```

```
*  week_arr(2)     - названия самих недель.
*
*  marazm(30,3)    - массив с таким "оригинальным" именем является
*                    одним из самых важных. Как Вы, быть может,
*                    помните, в расписании основными данными у нас
*                    являются предмет, преподаватель и аудитория -
*                    3 ячейки на одну пару. Максимальное число пар
*                    в день - 5, подгрупп - 2, 3x5x2==30 - именно
*                    таким числом полей мы можем представить на
*                    экране данные по расписанию для одной группы
*                    на один день, причем элементы с номерами вида
*                    1+3*n отвечают за данные по предмету, 2+3*n -
*                    по преподавателю, и 3+3*n - по аудитории.
*                    Первая ячейка каждой строки массива содержит
*                    код, вторая - соответствующий ему текст
*                    (списки, кстати, организованы наоборот -
*                    сначала текст, а затем уже код), третья имеет
*                    смысл только для поля предмета и содержит
*                    уровень дробления потока:
*
*                    1 - поток,
*                    2 - группа,
*                    3 - подгруппа.
*
*  first           - это всего лишь флаг первого вхождения в окно
*                    расписания. Он необходим для того, чтобы
*                    отличит инициализационные действия от
*                    многократно повторяющихся.
*
*  IsQuitFlag      - определяет условие выхода из цикла обработки
*                    сообщений от меню. VALID-функция для этого
*                    цикла возвращает значение этого флага. Если
*                    он очищен, цикл продолжается, при взводе -
*                    завершается.
*
*  Имена и заголовки окон Browse совпадают. Они могут
*присваиваться несколькими путями. Если команда BROWSE
*выполняется с фразой WINDOW, имя окна Browse будет совпадать с
*заголовком указанного описанного пользователем окна (не с
*именем). Если в команде BROWSE присутствует фраза TITLE, имя,
*указанное в этой фразе будет именем окна BROWSE.
*  В связи с этим (в продолжение "публикации" текста нашей
*программы) мы вначале проверяем, не было ли уже ранее определено
*окно для команды Browse, и если оно не определено, то -
*определяем его как способным изменять размеры, закрываться и
*минимизироваться; в качестве атрибутов даем ему тень, двойную
*рамку и цветовую схему команды Browse - 10. Обратите внимание на
*прием, использованный для того, чтобы центрировать это окно на
*любом экране, не только 80x25.

wait window nowait [Определение окон]
IF NOT WEXIST("browsewin")
        DEFINE WINDOW browsewin ;
                FROM INT((SROW()-17)/2),INT((SCOL()-60)/2) ;
                TO INT((SROW()-17)/2)+16,INT((SCOL()-60)/2)+59 ;
                FLOAT ;
```

```
                CLOSE ;
                SHADOW ;
                MINIMIZE ;
                DOUBLE ;
                COLOR SCHEME 10
ENDIF


*  Наиболее предпочтительным методом  включения вертикальных меню
*в приложение является использование системного горизонтального
*меню FoxPro и его вертикальных меню. При использовании
*вертикальных меню системного горизонтального меню нет
*необходимости вначале определять вертикальные меню, более того
*это приведет к ошибке. _MSYSMENU - это название системного меню.
*  В приведенном ниже фрагменте мы вначале очищаем системное меню
*от всех его BAR - и PAD - меню, заставляем отобразиться эту
*пустую полосочку и начинаем ее заполнять PAD - элементами, к
*которым далее привязываем вертикальные POPUP - меню, три из
*которых наполнены BAR - элементами. После всего этого фразами
*ON SELECTION BAR номер_пункта_меню OF имя_меню DO Имя_процедуры
*подвешиваем процедуры, реагирующие на выбор соответствующих
*пунктов меню.
*  Определив меню и процедуры - реагенты, мы организуем
*безоконный цикл чтения состояния меню до тех пор, пока
*VALID-функция IsQuit не прервет его. Такого рода цикл обработки
*сообщений - относительная новинка в FoxPro; подробнее Вы можете
*об этом прочитать в книге Les Pinter [FoxPro Application
*Programming] (Chapter 'Foundation Read').



wait window nowait [Инициализация меню]
SET SYSMENU TO

SET SYSMENU AUTOMATIC

DEFINE PAD helpquit OF _MSYSMENU PROMPT [\<Ё] COLOR SCHEME 3
DEFINE PAD listmake OF _MSYSMENU PROMPT "\<Составление "+;
                                    "списков " COLOR SCHEME 3
DEFINE PAD queryes OF _MSYSMENU PROMPT "\<Запросы" ;
                                                COLOR SCHEME 3
DEFINE PAD configur OF _MSYSMENU PROMPT "\<?" COLOR SCHEME 3
*                    "\<Конфигурация"
DEFINE PAD print_conf OF _MSYSMENU PROMPT "\<Настройки "+;
                                     "печати " COLOR SCHEME 3
ON PAD helpquit OF _MSYSMENU ACTIVATE POPUP hq_popup
ON PAD listmake OF _MSYSMENU ACTIVATE POPUP lm_popup
ON PAD queryes OF _MSYSMENU ACTIVATE POPUP qr_popup


DEFINE POPUP hq_popup MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR 1 OF hq_popup PROMPT "\<Help"
DEFINE BAR 2 OF hq_popup PROMPT "\<Quit"
ON SELECTION BAR 1 OF hq_popup DO help_proc &&in "pogruz.prg"
ON SELECTION BAR 2 OF hq_popup DO quit_proc &&in "pogruz.prg"
```

```
DEFINE POPUP lm_popup MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR 1 OF lm_popup PROMPT "\<Факультетов"
DEFINE BAR 2 OF lm_popup PROMPT "\<Кафедр"
DEFINE BAR 3 OF lm_popup PROMPT "\<Специальностей"
DEFINE BAR 4 OF lm_popup PROMPT "\<Преподавателей"
DEFINE BAR 5 OF lm_popup PROMPT "\<Потоков"
DEFINE BAR 6 OF lm_popup PROMPT "\<Предметов"
DEFINE BAR 7 OF lm_popup PROMPT "\<Аудиторий"
DEFINE BAR 8 OF lm_popup PROMPT "\<Пар расписания"
ON SELECTION BAR 1 OF lm_popup DO lm_fuc_proc
&&in "pogruz.prg"
ON SELECTION BAR 2 OF lm_popup DO lm_kaf_proc
&&in "pogruz.prg"
ON SELECTION BAR 3 OF lm_popup DO lm_spec_proc
&&in "pogruz.prg"
ON SELECTION BAR 4 OF lm_popup DO lm_prep_proc
&&in "pogruz.prg"
ON SELECTION BAR 5 OF lm_popup DO lm_grup_proc
&&in "pogruz.prg"
ON SELECTION BAR 6 OF lm_popup DO lm_pred_proc
&&in "pogruz.prg"
ON SELECTION BAR 7 OF lm_popup DO lm_aud_proc
&&in "pogruz.prg"
ON SELECTION BAR 8 OF lm_popup DO lm_par_proc
&&in "pogruz.prg"


DEFINE POPUP qr_popup MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR 1 OF qr_popup PROMPT "\<Расписание преподавателя"
*DEFINE BAR 2 OF qr_popup PROMPT "\<Свободной пары"

ON SELECTION BAR 1 OF qr_popup DO prep_rasp &&in "pogruz.prg"
*ON SELECTION BAR 2 OF qr_popup DO frpar_proc
&&in "pogruz.prg"
wait clear


read cycle valid IsQuit()
clear all
return

*  Функция IsQuit нигде в программе непосредственно не
*вызывается; она является VALID - условием, зацикливающим
*объединяющий READ, обрабатывающий сообщения от меню. Процедура
*help_proc, как видно из комментария к ней, вызывается при выборе
*пункта меню Help по команде ON SELECTION BAR 1 OF POPUP hq_popup
*DO help_proc.

function IsQuit
return IsQuitFlag

*       ┌-------------------------------------------------┐
*       ¦                                                 ¦
*       ¦help_proc  ON SELECTION BAR 1 OF POPUP hq_popup  ¦
*       ¦                                                 ¦
*       ¦ Procedure Origin:                               ¦
```

```
*       ¦                                                    ¦
*       ¦ From Menu:  POGRUZ.MPR,            Record:    5    ¦
*       ¦ Called By:  ON SELECTION BAR 1 OF POPUP hq_popup   ¦
*       ¦ Prompt:     Help                                   ¦
*       ¦ SniPPet:    1                                      ¦
*       ¦                                                    ¦
*       L----------------------------------------------------
*
PROCEDURE help_proc
   wait "Help isn't availaible yet.." WINDOW NOWAIT
RETURN


*  Уйти красиво в программе бывает даже важнее, чем в жизни.
*Поэтому мы и завели процедуру, реагирующую на выбор в меню
*пункта Quit. Как Вы, быть может, еще не забыли, у нас 9 баз
*данных, размещенных в 10 рабочих областях, где 5 - пустая. Перед
*выходом мы, естественно должны их закрыть; если в базе были
*сделаны какие - либо изменения, по правилам хорошего тона ее
*желательно упаковать.
*  Изменялась ли база, мы можем узнать, просмотрев массив
*WasChanged, поэтому организуем цикл Просмотр массива - Упаковка
*(только если были изменения) - Закрытие базы данных.
*  Команда SELECT не может принять в качестве параметра временную
*переменную, поэтому воспользуемся описанной ранее функцией
*макроподстановки. Для этого переведем номер рабочей области в
*строку и подставим в SELECT содержимое этой строки.
*  Затем мы восстановим стандартное системное меню, выведем
*системной функцией на экран заставку FoxPro, и очистим все READ
*- циклы. Для Foundation READ, обрабатывавшего наше меню, мы
*взведем флажок выхода - мол, хватит работать, после чего наша
*программа благополучно завершится.
*
*       r----------------------------------------------------¬
*
*       ¦ quit_proc  ON SELECTION BAR 2 OF POPUP hq_popup    ¦
*       ¦
*       ¦ Procedure Origin:                                  ¦
*       ¦                                                    ¦
*       ¦ From Menu:  POGRUZ.MPR,            Record:    6    ¦
*       ¦ Called By:  ON SELECTION BAR 2 OF POPUP hq_popup   ¦
*       ¦ Prompt:     Quit
*
*       ¦                                                    ¦
*
*

PROCEDURE quit_proc
  wait window nowait [Очистка рабочих областей]
  for num=1 to 10
    pol=str(num)
    select &pol
    if WasChanged(num)
      pack
      wait window nowait [Упаковка...]
    endif
```

```
    use
  endfor
  select 1
  wait window nowait [Подготовка к выходу]

  clear
  set sysmenu to default
  IsQuitFlag=.t.
  clear read
  =sys(2017)
  wait clear
return



*  Нижеприведенная процедура позволит Вам заполнить /
*модифицировать базу данных по преподавателям.
*  Перейдя в рабочую область БД по преподавателям, мы для
*удобства редактирования добавим в конец пустую запись. Затем,
*после необходимых проверок, заполним список названиями и кодами
*кафедр. При этом, если БД по кафедрам уже изменялась, ее надо
*упаковать, так как команда копирования базы в массив не обращает
*внимания на то, что запись может быть пустой или помеченной к
*удалению, а занесение такой записи в список кафедр (обычно
*имеющей вид .F.) нежелательно. Далее, найдя в списке кафедр ту,
*которая соответствует коду в первой записи БД по преподавателям,
*мы запоминаем этот номер, чтобы в дальнейшем подсветить его в
*списке, но уже на экране.
*  Сконструировав список на экране из элементов массива list_kuf,
*мы будем заносить выбор в переменную ch_kuf. VALID - функция для
*списка будет вызываться при выборе его элемента с двумя
*параметрами: кодом кафедры и ее названием. Эта функция код
*занесет в базу данных, а название отобразит на экране.
*  В поля БД с именами rank и zvan мы будем заносить номер
*элемента в соответствующих кнопках - меню, а в поле fio -
*информацию из окна редактирования.
*  Сбоку экрана создадим серию из 5 кнопок, одна из которых будет
*по умолчанию (\!), то есть она будет "нажиматься" по комбинации
*Ctrl+Enter, а другая (\?) будет реагировать на нажатие клавиши
*Esc. Для отдельной обработки каждой кнопки в VALID - функцию мы
*передадим ее номер.
*  Свои действия в этой функции мы будем сопровождать
*разнообразными проверками: на выход за пределы базы, на пустоту
*записи и т.д.
*  После формирования всех полей на экране и заполнения их
*информацией из первой записи мы отключим системное меню и
*организуем цикл обработки сообщений, прерываемый по кнопке
*"Выход". Восстановив и отбразив меню, мы очищаем экран и
*занимаемся гигиеной: ищем пустые записи и удаляем их из базы.
*  Казалось бы, мы сделали уже все, однако фамилию преподавателя
*мы ввели, а код - нет! И не надо: код мы сформируем программно,
*по одному и тому же алгоритму во всех местах, суть которого
*заключается в следующем.
*  Если код у нас не сформирован, то в соответствующем поле базы
*стоит нолик, поэтому мы ищем в базе запись с нулевым кодом.
*Найдя ее, заводим переменную с начальным значением 1 и
```

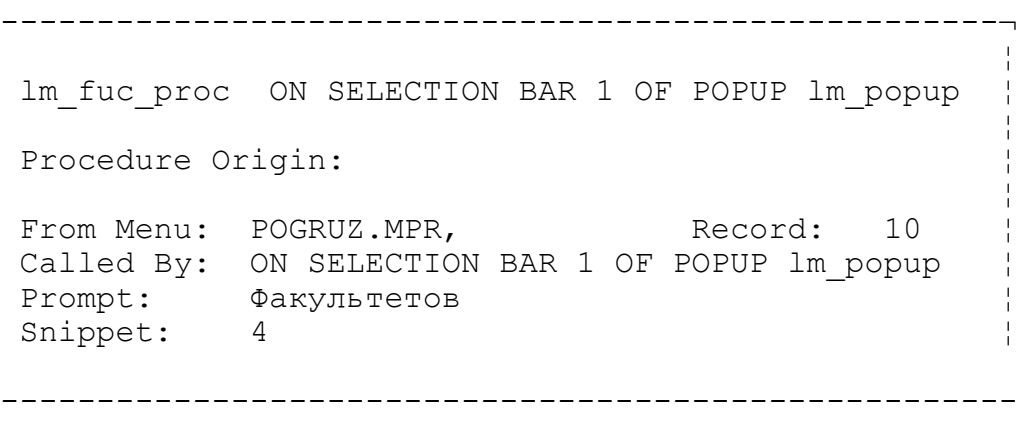
```
*наращиваем ее до тех пор, пока она совпадает хотя бы с одним
*кодом из уже имеющихся в базе. Главный критерий кода -
*уникальность, поэтому, найдя число, для которого нет совпадений
*среди уже имеющихся кодов, мы можем его принять за handle для
*данной записи, и заменить пустой (нулевой) код этим числом.
*
*
*       ┌-------------------------------------------------┐
*       ¦                                                 ¦
*       ¦ lm_prep_proc  ON SELECTION BAR 4 OF POPUP lm_popup ¦
*       ¦                                                 ¦
*       ¦ Procedure Origin:                               ¦
*       ¦                                                 ¦
*       ¦ From Menu:  POGRUZ.MPR,            Record:    9 ¦
*       ¦ Called By:  ON SELECTION BAR 4 OF POPUP lm_popup ¦
*       ¦ Prompt:     Преподавателей                      ¦
*       ¦ Snippet:                                        ¦
*       ¦                                                 ¦
*       L-------------------------------------------------
*
PROCEDURE lm_prep_proc
  wait window nowait [Инициализация экранных объектов]
  select c
  go bottom
  insert blank
  go top

  *Ввод данных по преподавателю
  *Объявим массив, куда считаем кафедры и их коды

  select h&&8
  if WasChanged(8)
    pack
    WasChanged(8)=.f.
  endif
  if reccount()==0
    =bell()
    wait "Вначале необходимо заполнить базу данных по"+;
         "кафедрам" window NOWAIT
    return
  endif
  dimension list_kuf(reccount(),2)
  copy to array list_kuf

  *Поиск кода кафедры
  for _i=1 to reccount()
    if c.kafedra==list_kuf(_i,2)
      ch_kuf=_i
    endif
  endfor

  select c
  WasChanged(3)=.T.

  @1,1  SAY [             Список  кафедр             ]
  @3,1  GET ch_kuf FROM list_kuf SIZE 20,40 DEFAULT 1 ;
```

```
       VALID funchkuf(list_kuf(ch_kuf,2),list_kuf(ch_kuf,1)) ;
           MESSAGE [Двойным щелчком или нажатием клавиши]+;
           [ Ввод в этом списке производят выбор]
           &&Кидаем код кафедры
  if empty(kafedra)

  else
    @23,1 SAY "Работает на кафедре : ";
         +list_kuf(ch_kuf,1)+SPACE(65)
  endif
  @1,50 SAY [      Должность    ]
  @2,50 GET rank ;
  FUNCTION "^ \<Ректор;\<Проректор;\<Декан;\<Зав. кафедрой;";
           +"\<Профессор;\<Доктор;\<Доцент;\<Ст. преподаватель;";
            +"\<Ассистент;\<Лаборант" DEFAULT 10;
            MESSAGE [Нажатием пробела и Ввода или ]+;
            [удерживанием мыши в этом меню - выбор]
  @5,50 SAY [     Ученое звание  ]
  @6,50 GET zvan FUNCTION "^ \<Профессор;\<Доцент;\<Ст."+;
            " преподаватель;";
            +"\<Ассистент" DEFAULT 4;
            MESSAGE [Нажатием пробела и Ввода или ]+;
            [удерживанием мыши в этом меню - выбор]
  @9,50 SAY "Ф.И.О. преподавателя"
  @10,50 EDIT fio SIZE 2,20,65 SCROLL;
              MESSAGE [Окно редактирования для ввода данных]+;
              [ о преподавателе]
  @13,50 GET knopka FUNCTION "*N \?Выход;\!Следующий;"+;
                            "Предыдущий;Вставить"+;
                            ";Удалить";
        SIZE 1,10,1 VALID fnknopka(knopka) DEFAULT 1
  wait clear
  set sysmenu off
  read cycle
  set sysmenu on
  set sysmenu automatic
  clear

  *ищем пустые записи и удаляем
  wait window nowait [Корректировка базы...]
  go top
  do while !eof()
    if empty(fio) .or. empty(kafedra) .or. empty(rank) .or.;
       empty(zvan)
      delete
    endif
    skip
  enddo
  =codeguard()
  wait clear
RETURN


FUNCTION funchkuf
parameters ch_kuf,p2
  replace kafedra with ch_kuf
```

```
  @23,1 SAY "Работает на кафедре : "+p2+SPACE(65)
RETURN


FUNCTION fnknopka
parameters kn
  DO CASE
    CASE kn=1&&quit
      clear read
      return
    CASE kn=2&&next
      if !eof()
        skip
      else
        =bell()
      endif
    CASE kn=3&&prev
      if !bof()
        skip -1
      else
        =bell()
      endif
    CASE kn=4
      *new
      insert blank
      *locate for code=0
      *skip
    CASE kn=5
      *delete
      delete
      if !eof()
        skip
      else
        =bell()
      endif
  ENDCASE
  select h
  *Поиск кода кафедры
  ch_kuf=0
  for _i=1 to reccount()
    if c.kafedra==list_kuf(_i,2)
      ch_kuf=_i
    endif
  endfor
  select c
  show gets
  if ch_kuf#0
    @23,1 SAY "Работает на кафедре : "+;
              list_kuf(ch_kuf,1)+SPACE(65)
  else
    @23,1 SAY "Работает на кафедре : ????"+SPACE(65)
  endif
RETURN
```

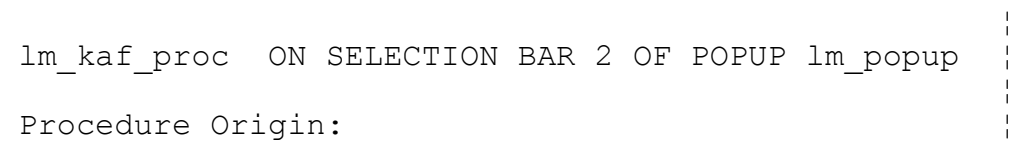
* Для каждого пункта меню нам необходимо определить выполняемые

*действия. В нижеследующих процедурах мы будем обслуживать все
*вспомогательные базы данных - по преподавателям, предметам и так
*далее.



* Эта процедура ведет обслуживание базы данных по факультетам
*очень примитивным образом: мы вначале переходим в рабочую
*область, где открыта эта база, а затем мы открываем BROWSE -
*окно для ее модификации. Изменять мы позволяем только одно поле
*- name (название кафедры), так как ее код мы сформируем
*программно. Вместо имени поля, как это делает BROWSE по
*умолчанию, мы выводим надпись "Название факультета", а сама
*команда BROWSE выполняется в специально определенном ранее окне
*с именем BrowseWin. По окончании сеанса редактирования записей
*мы проделываем традиционные действия - помечаем к удалению
*имеющиеся, но незаполненные записи и формируем коды для новых
*записей.

```
PROCEDURE lm_fuc_proc
  select f
  WasChanged(6)=.T.
  browse fields name :h="Название факультета" WINDOW BrowseWin;
         TITLE "База данных по факультетам"+;
         " Сегодня - "+DTOC(DATE())
  wait window nowait [Корректировка базы...]
  *ищем пустые записи и удаляем
  go top
  do while !eof()
    if empty(name)
      delete
    endif
    skip
  enddo
  =codeguard()
  wait clear
RETURN
```



```
*       ¦                                                    ¦
*       ¦ From Menu:  POGRUZ.MPR,
*       ¦ Called By:  ON SELECTION BAR 2 OF POPUP lm_popup   ¦
*       ¦ Prompt:     Кафедр                                 ¦
*       ¦ Snippet:    5                                      ¦
*       ¦                                                    ¦
*       L----------------------------------------------------
*
*  Структура базы данных по кафедрам ничем не отличается от
*структуры БД по факультетам, поэтому и процедуры их обслуживания
*практически идентичны.
*
PROCEDURE lm_kaf_proc
  select h
  WasChanged(8)=.T.
  browse fields name :h="Название кафедры" WINDOW BrowseWin;
         TITLE "База данных по кафедрам"+;
         " Сегодня - "+DTOC(DATE())

  wait window nowait [Корректировка базы...]
  *ищем пустые записи и удаляем
  go top
  do while !eof()
    if empty(name)
      delete
    endif
    skip
  enddo
  =codeguard()
  wait clear
RETURN



*       r----------------------------------------------------¬
*       ¦                                                    ¦
*       ¦ lm_spec_proc  ON SELECTION BAR 3 OF POPUP lm_popup ¦
*
*       ¦ Procedure Origin:                                  ¦
*       ¦                                                    ¦
*       ¦ From Menu:  POGRUZ.MPR,            Record:   12    ¦
*       ¦ Called By:  ON SELECTION BAR 3 OF POPUP lm_popup   ¦
*       ¦ Prompt:
*       ¦ Snippet:    6                                      ¦
*       ¦                                                    ¦
*       L----------------------------------------------------
*
*  Каждая специальность привязана к какому - либо факультету,
*поэтому мы должны это сделать и в базе. Для этого в БД по
*специальностям заносится код факультета и название
*специальности.
*  Перейдя в рабочую область БД по специальностям, мы для
*удобства редактирования добавим в конец пустую запись. Затем,
*после необходимых проверок, заполним массив названиями и кодами
*факультетов. При этом, если БД по факультетам уже изменялась, ее
```

```
*надо упаковать, так как команда копирования базы в массив не
*обращает внимания на то, что запись может быть пустой или
*помеченной к удалению, а занесение такой записи в список
*факультетов (обычно имеющей вид .F.) нежелательно. Далее, найдя
*в списке факультетов тот, который соответствует коду в первой
*записи БД по специальностям, мы запоминаем этот номер, чтобы в
*дальнейшем подсветить его в списке на экране.
*  Все наши действия мы будем проводить не на экране, а в окне,
*которое мы определим и активизируем так, чтобы создаваемы нами
*экранные объекты отображались на экране только после повторной
*активации окна. Это делается для того, чтобы прорисовка окна,
*кнопок, списка и т.д., выполняемые FoxPro в процессе создания
*экранных объектов очень медленно, не были видны.
*  Сконструировав список на экране из элементов массива list_fac,
*мы будем заносить выбор в переменную ch_fac. VALID - функция для
*списка будет вызываться при выборе его элемента с двумя
*параметрами: кодом факультета и его названием. Эта функция код
*занесет в базу данных, а название отобразит на экране.
*  В поле БД с именем name занесем информацию из окна
*редактирования.
*  Сбоку экрана создадим серию из 5 кнопок, одна из которых будет
*по умолчанию (\!), то есть она будет "нажиматься" по комбинации
*Ctrl+Enter, а другая (\?) будет реагировать на нажатие клавиши
*Esc. Для отдельной обработки каждой кнопки в VALID - функцию мы
*передадим ее номер. Свои действия в этой функции мы будем
*сопровождать разнообразными проверками: на выход за пределы
*базы, на пустоту записи и т.д.
*  После формирования всех полей на экране и заполнения их
*информацией из первой записи мы отключим системное меню и
*организуем цикл обработки сообщений, прерываемый по кнопке
*"Выход". Восстановив и отбразив меню, мы удаляем окно и
*занимаемся гигиеной: ищем пустые записи и удаляем их из базы.


PROCEDURE lm_spec_proc
  select g
  WasChanged(7)=.T.
  *browse fields name :h="Название специальности";
*    WINDOW BrowseWin;
        * TITLE "База данных по специальностям"+" Сегодня -
        *"+DTOC(DATE())

  wait window nowait [Инициализация оконной подсистемы...]
  append blank
  go top

  select f&&6
  if WasChanged(6)
    pack
    WasChanged(6)=.f.
  endif
  if reccount()==0
    =bell()
    wait "Вначале необходимо заполнить базу данных по"+;
         " факультетам" window;
     NOWAIT
```

```
    return
  endif
  dimension list_fac(reccount(),2)
  copy to array list_fac

  *Поиск кода
  for _i=1 to reccount()
    if g.fac==list_fac(_i,2)
      ch_fac=_i
    endif
  endfor

  select g

        DEFINE WINDOW sd3 ;
                FROM INT((SROW()-19)/2),INT((SCOL()-50)/2) ;
                TO INT((SROW()-19)/2)+18,INT((SCOL()-50)/2)+49 ;
                TITLE "База данных по специальностям" ;
                FLOAT NOCLOSE SHADOW DOUBLE COLOR SCHEME 1
        ACTIVATE WINDOW sd3 NOSHOW
@ 2,1 GET ch_fac PICTURE "@&N" FROM list_fac SIZE 13,27;
                 DEFAULT 1 VALID funchfac(list_fac(ch_fac,2),;
                 list_fac(ch_fac,1)) ;
            MESSAGE [Двойным щелчком или нажатием клавиши]+;
            [ Ввод в этом списке производят выбор];
                COLOR SCHEME 9
@ 0,5 SAY "Список факультетов"
@ 16,1 SAY "Факультет :"
  if empty(fac)

  else
    @16,13 SAY list_fac(ch_fac,1)+SPACE(65)
  endif
@ 0,33 SAY "Специальность"
@ 2,32 EDIT name ;
        SIZE 2,15,65 ;
        DEFAULT "" SCROLL;
        MESSAGE [Окно редактирования для ввода данных о]+;
        [ специальности]
@ 5,33 GET knopka ;
        PICTURE "@*VN \?Выход;\!Следующий;Предыдущий;Вставить"+;
        " ;Удалить" ;
        SIZE 1,12,1 VALID f5knopka(knopka) DEFAULT 1

set sysmenu off
        ACTIVATE WINDOW sd3
  wait clear
READ CYCLE MODAL

RELEASE WINDOW sd3
set sysmenu on
set sysmenu automatic

  wait window nowait [Корректировка базы...]
  *ищем пустые записи и удаляем
  go top
```

```
  do while !eof()
    if empty(name) .or. empty(fac)
      delete
    endif
    skip
  enddo
  =codeguard()
  wait clear
RETURN


FUNCTION funchfac
parameters ch,p2
  replace fac with ch
  @16,13 SAY p2+SPACE(65)
RETURN


FUNCTION f5knopka
parameters kn
  DO CASE
    CASE kn=1&&quit
      clear read
      return
    CASE kn=2&&next
      if !eof()
        skip
      else
        =bell()
      endif
    CASE kn=3&&prev
      if !bof()
        skip -1
      else
        =bell()
      endif
      ch_fac=fac
    CASE kn=4
      *new
      insert blank
    CASE kn=5
      *delete
      delete
      if !eof()
        skip
      else
        =bell()
      endif
  ENDCASE
  select f&&6
  ch_fac=0
  *Поиск кода
  for _i=1 to reccount()
    if g.fac==list_fac(_i,2)
      ch_fac=_i
    endif
```

```
  endfor
  select g
  show gets
  if ch_fac#0
    @16,13 SAY list_fac(ch_fac,1)+SPACE(65)
  else
    @16,13 SAY "????"+SPACE(65)
  endif
RETURN



*       ┌-------------------------------------------------------┐
*       ¦                                                       ¦
*       ¦ lm_pred_proc  ON SELECTION BAR 6 OF POPUP lm_popup    ¦
*       ¦                                                       ¦
*       ¦ Procedure Origin:                                     ¦
*       ¦                                                       ¦
*       ¦ From Menu:  POGRUZ.MPR,            Record:   13       ¦
*       ¦ Called By:  ON SELECTION BAR 6 OF POPUP lm_popup      ¦
*       ¦ Prompt:     Предметов                                 ¦
*       ¦ Snippet:    7                                         ¦
*       ¦                                                       ¦
*       L-------------------------------------------------------┘
*
*  Для заполнения БД по предметам мы могли бы воспользоваться и
*@...GET - объектами, но не менее красиво это можно сделать одной
*из разновидностей команды BROWSE - CHANGE. Если BROWSE
*предъявляет данные в виде таблицы, то CHANGE - в виде своего
*рода набора карточек: каждая запись выводится в несколько строк,
*в каждой строке находится поле в виде "название - содержимое".
*  По закрытию окна редактирования мы, как обычно, удаляем пустые
*записи (пустой будем в дальнейшем считать запись, у которой
*незаполнено хотя бы одно поле) и формируем коды для новых
*специальностей.
*
*
PROCEDURE lm_pred_proc
  *(code N(4,0),name C(65),link L,chas_week N(3,1),chas_all
  *N(4,1))
  select a
  WasChanged(1)=.T.
  change title [База данных по предметам Сегодня - ]+;
         DTOC(DATE());
         fields name;
                :h='Название предмета',;
                chas_week:h=[Часов в неделю],;
                chas_all:h="Общее число часов",;
                link:h="Связь с аудиторией" WINDOW BrowseWin
  *ищем пустые записи и удаляем
  wait window nowait [Корректировка базы...]
  go top
  do while !eof()
    if empty(name) .OR. empty(chas_week) .OR. empty(chas_all)
      delete
    endif
```

```
    skip
  enddo
  =codeguard()
  wait clear
RETURN



*       ┌-------------------------------------------------┐
*       ¦                                                 ¦
*       ¦ lm_aud_proc  ON SELECTION BAR 7 OF POPUP lm_popup ¦
*       ¦                                                 ¦
*       ¦ Procedure Orig@n:                               ¦
*       ¦                                                 ¦
*       ¦ From Menu:  POGRUZ.MPR,            Record:   14 ¦
*       ¦ Called By:  ON SELECTION BAR 7 OB POPUP lm_popup ¦
*       ¦ Prompt:     Аудиторий                           ¦
*       ¦ Snippet:    8                                   ¦
*       ¦                                                 ¦
*       L-------------------------------------------------┘
*
*  Окно, в общем то, вещь достаточно условная - с помощью
*"изобразительных средств" самого FoxPro (@... - команд) его
*можно сымитировать с достаточно высокой степенью достоверности,
*что мы и попробуем сделать на примере обслуживания БД по
*аудиториям.
*  Вначале мы перейдем в рабочую область, где расположена база
*данных по аудиториям, добавим в конец пустую запись (для
*удобства редактирования) и установим признак изменения данных в
*массиве WasChanged. Заметим, что признак специализации аудитории
*у нас поле логическое, а в случае использования управляющих
*элементов в стиле Windows мы вынуждены использовать только
*числовые поля; поэтому заведем в программе временную переменную,
*в которой будем помещать 1, если аудитория не специализирована и
*двойку при специализации.
*  На экране командой @1,5 TO 23,69 DOUBLE мы начертим окно без
*тени с двойной рамкой, в котором расположим поля ввода номеров
*корпуса и аудитории, выпадающее меню для выбора емкости
*аудитории и две радио - кнопочки для выбора типа аудитории
*(признака специализации). Для корректной замены данных по
*специализации в базе мы придадим этим кнопкам VALID - функцию
*swap, преобрзующих число из переменной temp в логическое
*значение по описанному выше признаку.
*  В отдельном окошке у нас будет особое мемо - поле, содержимое
*которого не контролируется, но всегда отображается.
*  После отработки цикла сообщений мы, как обычно, очищаем базу
*от пустых записей и формируем код.
*
*
PROCEDURE lm_aud_proc
  *(code n(4,0),korpus n(2,0),aud n(3,0),emkost n(3,0),
  *special L,comment M)
  wait window nowait [Инициализация экрана...]
  select b
  go bottom
```

```
  insert blank
  go top
  WasChanged(2)=.T.
  if special=.t.
    temp=2
  else
    temp=1
  endif

  @1,5 TO 23,69 DOUBLE
  @3,11 SAY [Номер корпуса]
  @3,31 GET korpus MESSAGE [Здесь вводится номер корпуса,]+;
                           [не равный нулю]
  @3,41  SAY [Номер аудитории]
  @3,61 GET aud MESSAGE [Здесь вводится номер аудитории,]+;
                        [не равный нулю]
  @5,12 SAY [    Емкость]
  @6,13 GET emkost FUNCTION "^ \<Поток;\<Группа;\<Подгруппа";
              DEFAULT 3;
              MESSAGE [Нажатием пробела и Ввода или ]+;
              [удерживанием мыши в этом меню - выбор]
  @9,8 SAY [      Специализация :]
  @11,9 GET temp FUNCTION '*R \<Не'+;
        ' специализирована;\<Специализирована';
        DEFAULT 1 VALID swap(temp);
        MESSAGE [Нажатием мыши или движением курсора - выбор]
  @14,8 SAY "Описание аудитории (если нужно)"
  @16,7 edit comment SIZE 6,40,3000 SCROLL;
            MESSAGE [Можете описать здесь специфические]+;
            [ особенности данной аудитории]
  @7,55 GET knopka FUNCTION "*N \?Выход;\!Следующий;"+;
        "Предыдущий;Вставить;Удалить";
        SIZE 3,10,2 VALID f3knopka(knopka) DEFAULT 1
  set sysmenu off
  wait clear
  read cycle
  set sysmenu on
  set sysmenu automatic
  clear

  wait window nowait [Корректировка базы...]
  *ищем пустые записи и удаляем
  go top
  do while !eof()
    if empty(korpus) .or. empty(aud) .or. empty(emkost)
      delete
    endif
    skip
  enddo
  =codeguard()
  wait clear
RETURN


FUNCTION swap
parameters t
```

```
  if t=1
    replace special with .f.
  else
    replace special with .t.
  endif
RETURN



FUNCTION f3knopka
parameters kn
  DO CASE
    CASE kn=1&&quit
      clear read
      return
    CASE kn=2&&next
      if !eof()
        skip
      else
        =bell()
      endif
    CASE kn=3&&prev
      if !bof()
        skip -1
      else
        =bell()
      endif
    CASE kn=4
      *new
      insert blank
    CASE kn=5
      *delete
      delete
      if !eof()
        skip
      else
        =bell()
      endif
  ENDCASE
  if special=.f.
    temp=1
  else
    temp=2
  endif
  show gets
RETURN



*       ┌-------------------------------------------------------┐
*       ¦                                                       ¦
*       ¦ lm_grup_proc  ON SELECTION BAR 5 OF POPUP lm_popup    ¦
*       ¦                                                       ¦
*       ¦ Procedure Origin:                                     ¦
*       ¦                                                       ¦
```

```
*       ¦ From Menu:  POGRUZ.MPR,            Record:   16    ¦
*       ¦ Called By:  ON SELECTION BAR 5 OF POPUP lm_popup   ¦
*       ¦ Prompt:     Потоков                                ¦
*       ¦ Snippet:    10                                     ¦
*       ¦                                                    ¦
*       L-----------------------------------------------------
*
*  Пускай название нижеследующей процедуры Вас не обманывает -
*иметь дело мы будем с базой данных по потокам. Ее экранное
*представление будет достаточно традиционно - список, три
*выпадающих меню и набор стандартных кнопок для перемещения по
*базе.

PROCEDURE lm_grup_proc
  select d
  WasChanged(4)=.t.
  *(code n(5,0),special n(3,0),kurs n(1,0),delta n(2,0))
  *действия по вводу
  * специальность - списками, курс и смещение - popup
  wait window nowait [Инициализация экрана...]
  append blank
  go top

  *Ввод данных по группе
  *Объявим массивы, куда считаем специальности и их коды

  select g&&7
  if WasChanged(7)
    pack
    WasChanged(7)=.f.
  endif
  if reccount()==0
    =bell()
    wait "Вначале необходимо заполнить базу данных по"+;
         " специальностям" window;
     NOWAIT
    return
  endif
  dimension list_spec(reccount(),2)
  copy to array list_spec

  *Поиск кода специальности
  for _i=1 to reccount()
    if d.special==list_spec(_i,2)
      ch_spec=_i
    endif
  endfor


  select d

  @1,1  SAY [         Список  специальностей             ]
  @2,1  GET ch_spec FROM list_spec SIZE 20,40 DEFAULT 1 ;
  VALID funchspec(list_spec(ch_spec,2),list_spec(ch_spec,1)) ;
            MESSAGE [Нажатием Ввода или щелчком мыши в этом]+;
            [ списке - выбор]
```

```
            &&Кидаем код специальности
  if empty(special)
    @23,1 SAY "Специальность : ????"+SPACE(65)
  else
    @23,1 SAY "Специальность : "+list_spec(ch_spec,1)+SPACE(65)
  endif
  @2,48 SAY [Курс]
  @1,53 GET kurs;
  FUNCTION "^ \<Первый;\<Второй;\<Третий;\<Четвертый;\<Пятый";
              DEFAULT 1 MESSAGE [Нажатием пробела и Ввода ]+;
              [или удерживанием мыши в этом меню - выбор]
  @4,48 SAY [Групп в потоке]
  @5,50 GET grpcount ;
      FUNCTION "^ \<Одна;\<Две;\<Три;\<Четыре;\<Пять;"+;
            "\<Шесть;\<Семь;\<Восемь;\<Девять" DEFAULT 1;
        MESSAGE [Нажатием пробела и Ввода или удерживанием ]+;
                [мыши в этом меню - выбор]
  @8,48 SAY "Смещение по времени"
  @9,50 GET delta ;
  FUNCTION "^ \<Отсутствует;\<Одна пара;\<Две пары;"+;
            "\<Три пары;\<Четыре пары;\<Пять пар" DEFAULT 1;
    MESSAGE [Нажатием пробела и Ввода или удерживанием мыши ]+;
            [в этом меню - выбор]
  @14,50 GET knopka FUNCTION;
         "*N \?Выход;\!Следующий;Предыдущий;Вставить;Удалить";
        SIZE 1,10,1 VALID f2knopka(knopka) DEFAULT 1
  set sysmenu off
  wait clear
  read cycle
  set sysmenu on
  set sysmenu automatic
  clear
  wait window nowait [Корректировка базы...]
  *ищем пустые записи и удаляем
  go top
  do while !eof()
    if empty(special) .or. empty(kurs) .or. empty(grpcount);
       .or. empty(delta)
      delete
    endif
    skip
  enddo
  =codeguard()
  wait clear
RETURN



FUNCTION funchspec
parameters ch_s,p2
  replace special with ch_s
  @23,1 SAY "Специальность : "+p2+SPACE(65)
RETURN
```

```
FUNCTION f2knopka
parameters kn
  DO CASE
    CASE kn=1&&quit
      clear read
      return
    CASE kn=2&&next
      if !eof()
        skip
      else
        =bell()
      endif
   CASE kn=3&&prev
      if !bof()
        skip -1
      else
        =bell()
      endif
    CASE kn=4
      *new
      insert blank
    CASE kn=5
      *delete
      delete
      if !eof()
        skip
      else
        =bell()
      endif
  ENDCASE
  select g&&7
  ch_spec=0
  *Поиск кода специальности
  for _i=1 to reccount()
    if d.special==list_spec(_i,2)
      ch_spec=_i
    endif
  endfor
  select d
  if ch_spec#0
    @23,1 SAY "Специальность : "+;
                  list_spec(ch_spec,1)+SPACE(65)
  else
    @23,1 SAY "Специальность : ????"+SPACE(65)
  endif
  show gets
RETURN


*  Основная база данных, содержащая расписание занятий, своими
*полями имеет только коды, и это не удивительно - текстовое
*представление этих полей хранится во вспомогательных базах.
*
*       г-------------------------------------------------------¬
*       ¦                                                       ¦
*       ¦ lm_par_proc  ON SELECTION BAR 8 OF POPUP lm_popup     ¦
```

```
*       ¦                                                    ¦
*       ¦ Procedure Origin:                                  ¦
*       ¦                                                    ¦
*       ¦ From Menu:  POGRUZ.MPR,            Record:   15    ¦
*       ¦ Called By:  ON SELECTION BAR 8 OF POPUP lm_popup   ¦
*       ¦ Prompt:     Пар расписания                         ¦
*       ¦ Snippet:    9                                      ¦
*       ¦                                                    ¦
*       L----------------------------------------------------
*
*  Нижеследующая процедура - ключевая во всей нашей работе. При
*заполнении расписания очень желательно, чтобы вид экрана был
*максимально приближен к привычной нам (а особенно диспетчеру)
*форме; это позволит, возможно, сделать рассматриваемую нами
*программу используемой практически, а не только в качестве
*учебного пособия. Для этого мы сделаем наш экран рамкой,
*"ползущей" вверх/вниз по дням и влево/вправо по группам; тогда
*мы сможем на нем видеть обе подгруппы для одной группы и один
*день, то есть в общей сложности 2*5=10 клеточек расписания
*(если брать в расчет только одну неделю). В каждой клеточке у
*нас будет 3 поля: предмет, преподаватель и аудитория.
*  За единицу редактирования возьмем один поток. Тогда нам
*понадобится предварительно ввести сведения о факультете,
*специальности на этом факультете, потоке по специальности и виде
*недели (числитель/знаменатель).
*  После того, как все эти данные нам известны, мы можем выдать
*на экран окно с EDIT - полями и запустить цикл обработки
*сообщений от кнопок. В процессе работы с окном расписания мы
*производим переопределение функциональных клавиш F4 - для
*изменения данных в поле и F8 - для удаления нескольких полей.
*Для того, чтобы по окончания программы у нас по этим клавишам
*выполнялись стандартные действия, нам необходимо перед командой
*READ сохранить значения функциональных клавиш во внутреннем
*стеке FoxPro, а после того, как она отработает - восстановить.


PROCEDURE lm_par_proc
  *копируем список кафедр
  select h&&8
  if WasChanged(8)
    pack
    WasChanged(8)=.f.
  endif
  if reccount()==0
    =bell()
    wait "База данных по кафедрам пуста" window NOWAIT
    return
  endif
  dimension list_kuf(reccount(),2)
  copy to array list_kuf
  WasChanged(9)=.t.&&Признак необходимости упаковки в связи с
                   &&изменением расписания
  *потом это надо будет убрать и заполнить нормально
  *Вначале запросим факультет, специальность и поток
  *копируем список факультетов
  select f&&6
```

```
if WasChanged(6)
  pack
  WasChanged(6)=.f.
endif
if reccount()==0
  =bell()
  wait "База данных по факультетам пуста" window NOWAIT
  return
endif
dimension list_fac(reccount(),2)
copy to array list_fac
curfac=0
do while curfac==0
  =facfn()
enddo
*скопируем список специальностей на данном факультете
select g
if WasChanged(7)
  pack
  WasChanged(7)=.f.
endif
mcount=0
locate for fac==curfac
DO WHILE FOUND()
  mcount=mcount+1
  CONTINUE
ENDDO
if mcount==0
  =bell()
  wait [На этом факультете специальностей нет; заполните]+;
  [ базу ] window;
   NOWAIT
  curspec=0
  spectxt=[]
  curfac=0
  factxt=''
  return
endif
dimension list_spec(mcount,2)
mcount=0
locate for fac==curfac
DO WHILE FOUND()
  mcount=mcount+1
  list_spec(mcount,1)=name
  list_spec(mcount,2)=code
  CONTINUE
ENDDO
curspec=0
do while curspec==0
  =specfn()
enddo
*В список потоков занесем только те, которые есть на
*факультете по данной специальности
select d&&4
if WasChanged(4)
  pack
```

```
  WasChanged(4)=.f.
endif
mcount=0
locate for special==curspec
DO WHILE FOUND()
  mcount=mcount+1
  CONTINUE
ENDDO
if mcount==0
  =bell()
  wait [По данной специальности на этом факультете ]+;
       [потоков нет!] window NOWAIT
  curspec=0
  spectxt=[]
  curfac=0
  factxt=''
  return
endif
dimension list_pot(mcount,2)
mcount=0
locate for special==curspec
DO WHILE FOUND()
  mcount=mcount+1
  list_pot(mcount,1)=STR(kurs,1)+' курс'
  list_pot(mcount,2)=code
  CONTINUE
ENDDO
curpot=0
do while curpot==0
  =potfn()
enddo
*Запомним количество групп
locate for code==curpot
m.grpcount=grpcount
select i
week=0
do while !inlist(week,1,2)
  wait [Выбирайте : 1 - числитель, 2 - знаменатель] window;
       to str_buf
  week=val(str_buf)
enddo
wait window nowait [Инициализация расписания...]
day=1
curgrp=1
for _i=1 to 30
  marazm(_i,1)=0
  marazm(_i,2)=[]
endfor


      DEFINE WINDOW mainwindow ;
             FROM 1, 0 ;
             TO 24,77;
             TITLE ALLTRIM(spectxt)+[ , ]+ALLTRIM(pottxt)+[ ]+;
             ALLTRIM(factxt)+[ факультет] ;
             FOOTER "F4 - Изменить, F8 - Удалить пару" ;
```

```
               NOCLOSE ;
               SHADOW ;
               SYSTEM ;
               COLOR SCHEME 8
      ACTIVATE WINDOW mainwindow NOSHOW

  @ 0,16+22 SAY [Группа]
  @ 1,16+5  SAY [Подгруппа 1]
  @ 2,16+23 to 20,16+23
  @ 1,16+35  SAY [Подгруппа 2]

_j=1
for _i=0 to 4
  sho=1+_i-5*int(_i/5)
  @ _i*4+3,10 SAY str(sho,1)
  @ _i*4+2,16 EDIT marazm(_j,2) ;
      SIZE 1,21,65 ;
      DEFAULT []
  _j=_j+1
  @ _i*4+3,22 EDIT marazm(_j,2) ;
      SIZE 1,15,65 ;
      DEFAULT []
  _j=_j+1
  @ _i*4+4,18 EDIT marazm(_j,2) ;
      SIZE 1,19,65 ;
      DEFAULT []
  _j=_j+1
endfor

_j=16
for _i=0 to 4
  sho=1+_i-5*int(_i/5)
  @ _i*4+3,10+35 SAY str(sho,1)
  @ _i*4+2,16+35 EDIT marazm(_j,2) ;
      SIZE 1,21,65 ;
      DEFAULT []
  _j=_j+1
  @ _i*4+3,22+35 EDIT marazm(_j,2) ;
      SIZE 1,15,65 ;
      DEFAULT []
  _j=_j+1
  @ _i*4+4,18+35 EDIT marazm(_j,2) ;
      SIZE 1,19,65 ;
      DEFAULT []
  _j=_j+1
endfor

@ 21,0 GET knopka ;
      PICTURE "@*HN Влево;Вправо;Вверх;Вниз;\?Выход" ;
      SIZE 1,14,1 ;
      DEFAULT 1 VALID knopkafn(knopka) DISABLE


  @ 0,16+30 SAY str(curgrp,1)
  @ 0,0 SAY days_arr(day)
```

```
        ACTIVATE WINDOW mainwindow
  push key
  set sysmenu off
  first=.t.
  READ CYCLE MODAL ACTIVATE read_activate()
  set sysmenu on
  set sysmenu automatic
  pop key

  RELEASE WINDOW mainwindow
  wait window nowait [Корректировка базы...]
  go top
  do while !eof()
    if empty(gruppa) .or. empty(para) .or. empty(predmet) ;
       .or. empty(audit) .or. empty(prepod)
      delete
    endif
    skip
  enddo
  wait clear
RETURN

*  Команда READ в предыдущей процедуре имеет у нас три параметра:
*CYCLE для многократного обхода GET / EDIT - полей, MODAL для
*того, чтобы из этог окна нельзя было перейти в меню или другое
*окно и ACTIVATE, используемый в данном случае для начального
*занесения данных в поля, включения кнопок и переопределения
*функциональных клавиш. Кроме того, для быстрого перемещения
*определим 4 клавиши - акселератора, по которым выполняется
*функция обработки сообщений от кнопок с различными параметрами.


function read_activate
  if first
    =refresh()
    on key label f4 do f4proc
    on key label f8 do f8proc&&erasing
    on key label ctrl+LEFTARROW do knopkafn with 1
    on key label ctrl+RIGHTARROW do knopkafn with 2
    on key label PgUp do knopkafn with 3
    on key label PgDn do knopkafn with 4
    show get knopka enable
    wait clear
    first=.f.
  endif
return

*  Процедура освежения экрана, вызывающаяся при начальной
*инициализации и любых других изменениях базы / экрана, имеет
*следующий алгоритм:
*  Прежде всего мы заносим на экран номер группы,день недели и
*очищаем все поля нашего массива, отвечающего за EDIT - поля в
*окне, после чего начинаем заносить данные по лабораторным
*занятиям, делая это в очень простом цикле по номеру подгруппы.
*Найдя запись по номеру подгруппы, мы вычислем ее номер и
*проверяем, в этот ли день проходит данная пара. Если день и
```

```
*неделя совпали, то мы переписываем все необходимые нам коды во
*временные переменные и по этим кодам в соответствующих базах
*данных формируем текстовые поля.
*  Текстовое поле для аудитории мы формируем из номера этой
*аудитории и номера корпуса, в которой она располагается, а текст
*для преподавателя - с помощью специальной функции, которая
*формирует строку вида "Доц. Полищук А.П." из полного имени и
*кода ученого звания.
*  Коды и текстовые поля мы заносим в соответствующие элементы
*массива. Между номером подгруппы, номером пары, номером EDIT -
*поля в паре (фактически определяющим, кокого рода данные
*находятся в данном поле) и индексом элемента в массиве
*существует взаимнооднозначное соответствие, устанавливаемое
*функцией GetObject. В дополнительный, третий столбец строки
*предмета мы заносим цифру 3 - признак лабораторного занятия.
*Сформировав полностью данные в массиве, мы командой show get
*принудительно отображаем их в окне.
*  Данные по практичкам и лекциям заносятся также, за одним
*исключением: на экране светятся у нас две подгруппы, а данные по
*практичкам / лекциям идут на группу / поток, поэтому для
*редактирования мы делаем доступными данные лишь в одной
*подгруппе - во второй они будут изменяться автоматически. Во
*избежание случайного ручного изменнения в поля - дубликаты
*заносятся только тексты, без кодов, и редактирование их
*запрещено.
*
*
procedure refresh
  wait window nowait [Освежение экрана...]
  @ 0,16+30 SAY str(curgrp,1)
  @ 0,0 SAY days_arr(day)
  for _i=1 to 30
    marazm(_i,1)=0
    marazm(_i,2)=[]
    marazm(_i,3)=0
    show get marazm(_i,2) enable
  endfor
  *отобразим сделанные изменения
  *(gruppa N(5,0),para N(3,0),predmet N(4,0),audit *
  *N(4,0),prepod N(4,0))
  *код пары формируется следующим образом :
  * неделя - 1 числитель,2 знаменатель  -
  * день -1 .. 5                        -
  * code=week*100+day*10+number
  * номер пары - 1..5                   -
  *код группы = код потока*100+номер группы*10+номер подгруппы
  *1 - занесем все лабораторки
  for _i=1 to 2 &&для обеих подгрупп
    locate for gruppa==curpot*100+curgrp*10+_i
    DO WHILE FOUND()
      *Выделим номера недели,дня и пары
      number=para-int(para/10)*10
      if (para-int(para/100)*100-number)/10==day ;
         .and. week==int(para/100)
        curpred=predmet
        curaud=audit
```

```
      curprep=prepod
      *Занесем данные по предмету
      select a
      locate for code==curpred
      predtxt=alltrim(name)
      select b
      locate for code==curaud
        if aud#0 .and. korpus#0
          audtxt=[Ауд.]+str(aud,3)+[, корп.]+str(korpus,2)
        else
          audtxt=[]
        endif
      select c
      locate for code==curprep
      preptxt=getprepname(fio,zvan)
      select i
      marazm(getobject(_i,number,1),1)=curpred
      &&Предмет
      marazm(getobject(_i,number,1),2)=predtxt
      &&Предмет
      show get marazm(getobject(_i,number,1),2) enable
      marazm(getobject(_i,number,1),3)=3 &&Признак лабы
      marazm(getobject(_i,number,2),1)=curprep
      &&Преподававтель
      marazm(getobject(_i,number,2),2)=preptxt
      &&Преподававтель
      show get marazm(getobject(_i,number,2),2) enable
      marazm(getobject(_i,number,3),1)=curaud
      &&Аудитория
      marazm(getobject(_i,number,3),2)=audtxt
      &&Аудитория
      show get marazm(getobject(_i,number,3),2) enable
    endif
    CONTINUE
  enddo
endfor
*2 занесем практички - на обе подгруппы
&& and lections too
for _j=1 to 0 step -1
  locate for gruppa==curpot*100+iif(_j==1,curgrp,0)*10
  DO WHILE FOUND()
    *Выделим номера недели,дня и пары
    number=para-int(para/10)*10
    if (para-int(para/100)*100-number)/10==day ;
       .and. week==int(para/100)
      * - заносим в расписание предмет препод и аудиторию
      curpred=predmet
      curaud=audit
      curprep=prepod
      *Занесем данные по предмету
      select a
      locate for code==curpred
      predtxt=alltrim(name)
      select b
      locate for code==curaud
        if aud#0 .and. korpus#0
```

```
            audtxt=[Ауд.]+str(aud,3)+[, корп.]+str(korpus,2)
          else
            audtxt=[]
          endif
        select c
        locate for code==curprep
        preptxt=getprepname(fio,zvan)
        select i
        marazm(getobject(1,number,1),1)=curpred
        &&Предмет
        marazm(getobject(1,number,1),2)=predtxt
        &&Предмет
        show get marazm(getobject(1,number,1),2) enable
        marazm(getobject(1,number,1),3)=1+_j
        &&Признак практички или лекции
        marazm(getobject(1,number,2),1)=curprep
        &&Преподававтель
        marazm(getobject(1,number,2),2)=preptxt
        &&Преподававтель
        show get marazm(getobject(1,number,2),2) enable
        marazm(getobject(1,number,3),1)=curaud
        &&Аудитория
        marazm(getobject(1,number,3),2)=audtxt
        &&Аудитория
        show get marazm(getobject(1,number,3),2) enable
        *отобразим дубликат серым цветом
        marazm(getobject(2,number,1),2)=predtxt
        &&Предмет
        show get marazm(getobject(2,number,1),2) disable
        marazm(getobject(2,number,2),2)=preptxt
        &&Преподававтель
        show get marazm(getobject(2,number,2),2) disable
        marazm(getobject(2,number,3),2)=audtxt
        &&Аудитория
        show get marazm(getobject(2,number,3),2) disable
      endif
      CONTINUE
    enddo
  endfor
  wait clear
return
*
*  Думаю, что нижеследующая функция достаточно прозрачна, чтобы
*ее комментировать.
*
function knopkafn
PARAMETERS kn
  DO CASE
    case kn=1&&Влево
      if curgrp#1
        curgrp=curgrp-1
      endif
    case kn=2&&Вправо
      if curgrp#m.grpcount
        curgrp=curgrp+1
      endif
```

```
    case kn=3&&Вверх
      if day#1
        day=day-1
      endif
    case kn=4&&Вниз
      if day#5
        day=day+1
      endif
    CASE kn=5&&Выход
      clear read
      return
  endcase
  =refresh()
return

*  По нажатию клавиши F8 мы удаляем данные по паре. Делать будем
*это так:
*  В FoxPro есть специальная системная переменная _CUROBJ,
*в которой хранится номер текущего GET - объекта. Всего на экране
*5*2*3==30 EDIT - полей и 5 кнопок. Нам интересен лишь тот
*случай, когда данная функциональная клавиша нажата на одном из
*полей редактирования. По его номеру мы с помощью функции GetWhat
*определяем тип поля, в котором мы находимся - то ли это поле с
*данными по предмету, то ли по преподавателю, то ли по аудитории,
*и в зависимости от типа поля корректируем переменную _i так,
*чтобы она указывала на поле предмета.
*  Сформировав индекс поля предмета в массиве, мы проверяем код на
*пустоту - а занесены ли там данные по предмету, и, если они
*присутствуют, а пользователь уверен в том, что удалить эту пару
*ему жизненно необходимо, мы, в зависимости от типа пары
*формируем код группы и пары для поиска. Найдя искомую пару в БД,
*мы ее удаляем, освежаем экран и восстанавливаем позицию курсора
*на том же месте, с которого начали.
*
*
procedure f8proc
  if _CUROBJ<31
    SAVEOBJ=_CUROBJ
    DO CASE
      CASE getwhat(SAVEOBJ)=1           &&Предмет
        _i=SAVEOBJ
      CASE getwhat(SAVEOBJ)=2           &&Преподаватель
        _i=SAVEOBJ-1
      CASE getwhat(SAVEOBJ)=3           &&Аудитория
        _i=SAVEOBJ-2
    ENDCASE
    if marazm(_i,1)#0
      if erase_ch()
        DO CASE
          CASE marazm(_i,3)==1
            _j=curpot*100
          CASE marazm(_i,3)==2
            _j=curpot*100+curgrp*10
          CASE marazm(_i,3)==3
            _j=curpot*100+curgrp*10+getpg(SAVEOBJ)
        ENDCASE
```

```
        select i
        locate for para==week*100+day*10+getnumber(SAVEOBJ);
                  .and. gruppa==_j
        delete
        *обновим экран
        =refresh()
      endif
    endif
    _CUROBJ=SAVEOBJ
  endif
return

*  Эта очень простая функция фомирует на экране окно с запросом
*об удалении и возвращает решение пользователя, преобразовав код
*выбранной клавиши в логическое значение.
*

function erase_ch
        DEFINE WINDOW werase_ch ;
                FROM INT((SROW()-11)/2),INT((SCOL()-29)/2) ;
                TO INT((SROW()-11)/2)+10,INT((SCOL()-29)/2)+28 ;
                TITLE "Вы уверены?" ;
                FLOAT ;
                NOCLOSE ;
                SHADOW ;
                DOUBLE ;
                COLOR SCHEME 7
        ACTIVATE WINDOW werase_ch NOSHOW
  @ 7,2 GET knopka ;
        PICTURE "@*HT \!Удалить;Отставить" ;
        SIZE 1,11,2 ;
        DEFAULT 2
  @ 0,4 SAY "Выбранная Вами пара"
  @ 1,1 SAY "будет удалена не только с"
  @ 2,3 SAY "экрана, но и из базы"
  @ 3,3 SAY "данных по расписанию."
  @ 4,4 SAY "Действительно ли Вы"
  @ 5,3 SAY "хотите именно этого?"
  @ 6,0 TO 8,14
  @ 6,14 TO 8,26
  @ 6,14 SAY "T"
  @ 8,14 SAY "+"

        ACTIVATE WINDOW werase_ch

  READ CYCLE MODAL

  RELEASE WINDOW werase_ch
RETURN IIF(knopka==1,.t.,.f.)

*  По клавише F4 в зависимости от того, на поле какого типа
*находится курсор, мы соответственно вызываем функции для ввода
*данных по предмету, преподавателю и аудитории.
*

procedure f4proc
```

```
  if _CUROBJ<31
    SAVEOBJ=_CUROBJ
    DO CASE
      CASE getwhat(SAVEOBJ)=1           &&Предмет
        =predfn()
      CASE getwhat(SAVEOBJ)=2           &&Преподаватель
        =kuffn()
      CASE getwhat(SAVEOBJ)=3           &&Аудитория
        =audfn()
    ENDCASE
    _CUROBJ=SAVEOBJ
  endif
return

*  Эта функция по номеру подгруппы, номеру пары и типу поля
*возвращает индекс строки в массиве.
*
*
function getobject
parameters podgruppa,number,chto
return (podgruppa-1)*15+(number-1)*3+chto


procedure getall
parameters object,podgruppa,number,chto
*из объекта получим все данные
  chto=iif(mod(object,3)==0,3,mod(object,3))
  podgruppa=iif(mod(object,15)==0,object/15,int(object/15)+1)
  number=iif(mod(object,3)==0,object/3,int(object/3)+1)
  &&абсолютный номер
  number=iif(mod(number,5)==0,5,mod(number,5))
return

*
*  В этой функции по номеру строки в массиве мы можем определить
*ее тип, то есть какого рода данные находятся в этой строке - по
*предмету, по преподавателю или аудитории.


function getwhat
parameters object
return iif(mod(object,3)==0,3,mod(object,3))

*  По номеру строки определяем номер подгруппы

function getpg
parameters object
return iif(mod(object,15)==0,object/15,int(object/15)+1)

*  По номеру строки определяем номер пары

function getnumber
parameters object
  number=iif(mod(object,3)==0,object/3,int(object/3)+1)
  &&абсолютный номер
  number=iif(mod(number,5)==0,5,mod(number,5))
```

```
return number

*  Функции potfn, specfn и facfn отображают на экране списки
*потоков, специальностей и факультетов и возвращают и выбор в
*cur... и ...txt - переменных.
*
function potfn

  aa=irand(1,5)
  cc=irand(1,50)

        DEFINE WINDOW potwindow ;
                from aa,cc to aa+19,cc+29;
                TITLE "Выбор потока" ;
                FOOTER "_Click there_" ;
                NOFLOAT ;
                NOCLOSE ;
                SHADOW ;
                DOUBLE ;
                COLOR SCHEME 1
                *FROM INT((SROW()-20)/2),INT((SCOL()-30)/2) ;
                *TO INT((SROW()-20)/2)+19,INT((SCOL()-30)/2)+29 ;

        ACTIVATE WINDOW potwindow NOSHOW
  @ 0,1 GET choice ;
        PICTURE "@&N" ;
        FROM list_pot ;
        SIZE 18,26 ;
        DEFAULT 1 ;
        COLOR SCHEME 2 VALID lpothundle(choice)

  ACTIVATE WINDOW potwindow
  READ CYCLE MODAL
  RELEASE WINDOW potwindow
return


Function lpothundle
parameters vib
  clear read
  curpot=list_pot(vib,2)
  *занесем выбор в полосу редактирования
  pottxt=list_pot(vib,1)
RETURN


function specfn
  aa=irand(1,5)
  cc=irand(1,50)

        DEFINE WINDOW specwindow ;
                from aa,cc to aa+19,cc+29;
                TITLE "Выбор специальности" ;
                FOOTER "_Click there_" ;
                NOFLOAT ;
                NOCLOSE ;
```

```
                SHADOW ;
                DOUBLE ;
                COLOR SCHEME 1
                *FROM INT((SROW()-20)/2),INT((SCOL()-30)/2) ;
                *TO INT((SROW()-20)/2)+19,INT((SCOL()-30)/2)+29 ;

        ACTIVATE WINDOW specwindow NOSHOW
  @ 0,1 GET choice ;
        PICTURE "@&N" ;
        FROM list_spec;
        SIZE 18,26 ;
        DEFAULT 1 ;
        COLOR SCHEME 2 VALID lspechundle(choice)

        ACTIVATE WINDOW specwindow
  READ CYCLE MODAL
  RELEASE WINDOW specwindow
Return


Function lspechundle
parameters vib
  clear read
  curspec=list_spec(vib,2)
  *занесем выбор в полосу редактирования
  spectxt=list_spec(vib,1)
  *откроем специальность и очистим его выборы
  pottxt=''
  curpot=0
RETURN


function facfn
  aa=irand(1,5)
  cc=irand(1,50)

        DEFINE WINDOW  facwindow ;
                from aa,cc to aa+19,cc+29;
                TITLE "Окно выбора факультета" ;
                FOOTER "_Click there_" ;
                NOFLOAT ;
                NOCLOSE ;
                SHADOW ;
                DOUBLE ;
                COLOR SCHEME 1
                *FROM INT((SROW()-20)/2),INT((SCOL()-30)/2) ;
                *TO INT((SROW()-20)/2)+19,INT((SCOL()-30)/2)+29 ;

        ACTIVATE WINDOW facwindow NOSHOW
  @ 0,1 GET choice ;
        PICTURE "@&N" ;
        FROM list_fac ;
        SIZE 18,26 ;
        DEFAULT 1 ;
        COLOR SCHEME 2 VALID lfachundle(choice)
```

```
  ACTIVATE WINDOW facwindow
  READ CYCLE MODAL
  RELEASE WINDOW facwindow

Return


Function lfachundle
parameters vib
  clear read
  curfac=list_fac(vib,2)
  *занесем выбор в полосу редактирования
  factxt=list_fac(vib,1)
  *откроем специальность и очистим его выборы
  spectxt=''
  curspec=0
  pottxt=[]
  curpot=0
RETURN

*
*  Эта функция возвращает случайное целое число, лежащее в
*заданном диапазоне значений.
*

FUNCTION irand
parameters i, j
RETURN int((j-i+1)*rand()+i)

NOTE Дзвоник

PROCEDURE bell
  *set bell to irand(19,10000),irand(1,19)
  ?CHR(7)
RETURN


FUNCTION getprepname
PARAMETERS f,r
  string=ALLTRIM(f)
  pos=AT([ ],string,1)
  if pos#0
    tempstr=LEFT(string,pos)+SUBSTR(string,pos+1,1)+[.]
    pos=AT([ ],string,2)
    if pos#0
      string=tempstr+SUBSTR(string,pos+1,1)+[.]
    endif
  endif
  DO CASE
*  "^ \<Профессор;\<Доцент;\<Ст. преподаватель;\<Ассистент"
    CASE r=1
      string="Проф. "+string
    CASE r=2
      string="Доц. "+string
    CASE r=3
      string="Ст.преп. "+string
```

```
    CASE r=4
      string="Acc. "+string
  ENDCASE
return string


function audfn
  *Список аудиторий сформируем так : если лекция или
  *практичка, то соответственно заносим лекционные или
  *практический аудитории
  *Если лабораторка не специализированная - заносим все
  *лабораторные, иначе - только лабораторные
  *специализированные
  *  Подумать, нельзя ли отдавать под лабы пр и лек ауд, под
  *практику лекционнные ?

  if WasChanged(2)&&b
    pack
    WasChanged(2)=.f.
  endif
  if marazm(SAVEOBJ-2,1)=0
    wait window nowait [Вначале необходимо выбрать предмет]
    return
  endif
  if marazm(SAVEOBJ-1,1)=0
    wait window nowait [Пожалуйста, выберите преподавателя]
    return
  endif
  *if marazm(SAVEOBJ-2,1)#0
  select a
  locate for code==marazm(SAVEOBJ-2,1)
  select b
  mcount=0
  if marazm(SAVEOBJ-2,3)==3
  *поиск аудитории, связанной с предметом, только если labor
    locate for emkost==marazm(SAVEOBJ-2,3) ;
          .and. special==a.link
  else
    locate for emkost==marazm(SAVEOBJ-2,3) ;
          .or. emkost==marazm(SAVEOBJ-2,3)-1
  endif
  DO WHILE FOUND()
    mcount=mcount+1
    CONTINUE
  ENDDO
  if mcount==0
    =bell()
    wait [Аудиторий требуемой емкости нет - заполните базу!];
        window
    RETURN
  endif
  dimension list_aud(mcount,2)
  mcount=0
  if marazm(SAVEOBJ-2,3)==3
    locate for emkost==marazm(SAVEOBJ-2,3) .and. ;
          special==a.link
```

```
  else
    locate for emkost==marazm(SAVEOBJ-2,3) ;
        .or. emkost==marazm(SAVEOBJ-2,3)-1
  endif
  DO WHILE FOUND()
    mcount=mcount+1
    list_aud(mcount,1)=[Ауд.]+STR(aud,4)+',корп.'+;
                       STR(korpus,2)
    list_aud(mcount,2)=code
    CONTINUE
  ENDDO

*еще один штрих - ищем в расписании
*только те аудитории, которые на данную пару свободны
  select i
  xcount=0
  for _i=1 to mcount
    locate for para==week*100+day*10+getnumber(SAVEOBJ) ;
          .and. audit==list_aud(_i,2)
    if found()&&На эту пару аудитория занята
      list_aud(_i,2)=0
      list_aud(_i,1)=[Занята !]+list_aud(_i,1)
      xcount=xcount+1
    endif
  endfor
  if xcount==mcount
    =bell()
    wait [Аудиторий требуемой емкости нет - все заняты] ;
         window NOWAIT
    return
  endif


  aa=irand(1,5)
  cc=irand(1,50)

        DEFINE WINDOW audwindow ;
                from aa,cc to aa+19,cc+29;
                TITLE "Выбор аудитории" ;
                FOOTER "_Click there_" ;
                NOFLOAT ;
                NOCLOSE ;
                SHADOW ;
                DOUBLE ;
                COLOR SCHEME 1
                *FROM INT((SROW()-20)/2),INT((SCOL()-30)/2) ;
                *TO INT((SROW()-20)/2)+19,INT((SCOL()-30)/2)+29 ;

        ACTIVATE WINDOW audwindow NOSHOW
  @ 0,1 GET choice ;
        PICTURE "@&N" ;
        FROM list_aud ;
        SIZE 18,26 ;
        DEFAULT 1 ;
        COLOR SCHEME 2 VALID laudhundle(choice)
```

```
  ACTIVATE WINDOW audwindow
  READ CYCLE MODAL
  RELEASE WINDOW audwindow
  if marazm(SAVEOBJ,1)#0
    &&если выбрана аудитория, то находим эту пару в базе и
    &&меняем в базе, как и на экране
    select i
    DO CASE
      CASE marazm(SAVEOBJ-2,3)==1&&лекция
        findpar=curpot*100
      CASE marazm(SAVEOBJ-2,3)==2&&практ
        findpar=curpot*100+curgrp*10
      CASE marazm(SAVEOBJ-2,3)==3&&лаба
        findpar=curpot*100+curgrp*10+getpg(SAVEOBJ)
    ENDCASE
    locate for para==week*100+day*10+getnumber(SAVEOBJ) ;
               .and. gruppa==findpar
    replace audit with marazm(SAVEOBJ,1)
    show get marazm(SAVEOBJ,2)
    *не надо ли отобразить в заблокированном окошке?
    if marazm(SAVEOBJ-2,3)#3&& если это не лабораторка
      if getpg(SAVEOBJ)==1
        marazm(SAVEOBJ+15,2)=marazm(SAVEOBJ,2)
        show get marazm(SAVEOBJ+15,2) disable
      else
        marazm(SAVEOBJ-15,2)=marazm(SAVEOBJ,2)
        show get marazm(SAVEOBJ-15,2) disable
      endif
    endif
  endif
  select i
return


Function laudhundle
Parameter vib
  if list_aud(vib,2)#0
    clear read
    marazm(SAVEOBJ,1)=list_aud(vib,2)
    *занесем выбор в полосу редактирования
    marazm(SAVEOBJ,2)=list_aud(vib,1)
  endif
RETURN



function prepfn
  *В список преподавателей занесем только тех, которые
  *работают на данной кафедре

  if WasChanged(3)
    pack
    WasChanged(3)=.f.
  endif
  select c
  * prepod (code N(4,0), kafedra N(3,0), fio C(65), rank
```

```
*N(2,0), zvan N(2,0))
mcount=0
locate for kafedra==curkuf
DO WHILE FOUND()
  mcount=mcount+1
  CONTINUE
ENDDO
if mcount==0
  =bell()
  wait [На данной кафедре никто не работает! Заполните ]+;
       [базу] window NOWAIT
  curkuf=0
  kuftxt=[]
  return
endif
dimension list_prep(mcount,2)
locate for kafedra==curkuf
mcount=0
DO WHILE FOUND()
  mcount=mcount+1
  list_prep(mcount,1)=getprepname(fio,zvan)
  list_prep(mcount,2)=code
  CONTINUE
ENDDO
select i
*свободен ли преподаватель?
xcount=0
for _i=1 to mcount
  locate for para==week*100+day*10+getnumber(SAVEOBJ);
             .and. prepod==list_prep(_i,2)
  if found()&&На эту пару преподаватель занят
    list_prep(_i,2)=0
    list_prep(_i,1)=[Занят !]+list_prep(_i,1)
    xcount=xcount+1
  endif
endfor
if xcount==mcount
  =bell()
  wait [Все преподаватели кафедры на этой паре заняты];
        window NOWAIT
  return
endif

aa=irand(1,5)
cc=irand(1,50)

      DEFINE WINDOW prepwindow ;
             from aa,cc to aa+19,cc+29;
             TITLE "Выбор преподавателя" ;
             FOOTER "_Click there_" ;
             NOFLOAT ;
             NOCLOSE ;
             SHADOW ;
             DOUBLE ;
             COLOR SCHEME 1
             *FROM INT((SROW()-20)/2),INT((SCOL()-30)/2) ;
```

```
                *TO INT((SROW()-20)/2)+19,INT((SCOL()-30)/2)+29 ;

        ACTIVATE WINDOW prepwindow NOSHOW
  @ 0,1 GET choice ;
        PICTURE "@&N" ;
        FROM list_prep ;
        SIZE 18,26 ;
        DEFAULT 1 ;
        COLOR SCHEME 2 VALID lprephundle(choice)

  ACTIVATE WINDOW prepwindow
  READ CYCLE MODAL
  RELEASE WINDOW prepwindow
  if marazm(SAVEOBJ,1)#0
    &&если выбран, то находим эту пару в базе и меняем
    &&в базе, как и на экране
    select i
    DO CASE
      CASE marazm(SAVEOBJ-1,3)==1&&лекция
        findpar=curpot*100
      CASE marazm(SAVEOBJ-1,3)==2&&практ
        findpar=curpot*100+curgrp*10
      CASE marazm(SAVEOBJ-1,3)==3&&лаба
        findpar=curpot*100+curgrp*10+getpg(SAVEOBJ)
    ENDCASE
    locate for para==week*100+day*10+getnumber(SAVEOBJ) ;
               .and. gruppa==findpar
    replace prepod with marazm(SAVEOBJ,1)
    show get marazm(SAVEOBJ,2) enable
    *не надо ли отобразить в заблокированном окошке?
    if marazm(SAVEOBJ-1,3)#3&& если это не лабораторка
      if getpg(SAVEOBJ)==1
        marazm(SAVEOBJ+15,2)=marazm(SAVEOBJ,2)
        show get marazm(SAVEOBJ+15,2) disable
        show get marazm(SAVEOBJ+16,2) disable
      else
        marazm(SAVEOBJ-15,2)=marazm(SAVEOBJ,2)
        show get marazm(SAVEOBJ-15,2) disable
        show get marazm(SAVEOBJ-14,2) disable
      endif
    endif
  endif
return


Function lprephundle
Parameter vib
  if list_prep(vib,2)#0
    clear read
    marazm(SAVEOBJ,1)=list_prep(vib,2)
    *занесем выбор в полосу редактирования
    marazm(SAVEOBJ,2)=list_prep(vib,1)
  endif
RETURN
```

```
function kuffn
  aa=irand(1,5)
  cc=irand(1,50)

  if WasChanged(8)
    pack
    WasChanged(8)=.f.
  endif
  if marazm(SAVEOBJ-1,1)==0
    wait window nowait [Вначале необходимо выбрать предмет]
    return
  endif

        DEFINE WINDOW  kufwindow ;
                from aa,cc to aa+19,cc+29;
                TITLE "Окно выбора кафедры" ;
                FOOTER "_Click there_" ;
                NOFLOAT ;
                NOCLOSE ;
                SHADOW ;
                DOUBLE ;
                COLOR SCHEME 1
                *FROM INT((SROW()-20)/2),INT((SCOL()-30)/2) ;
                *TO INT((SROW()-20)/2)+19,INT((SCOL()-30)/2)+29 ;

        ACTIVATE WINDOW  kufwindow NOSHOW
  @ 0,1 GET choice ;
        PICTURE "@&N" ;
        FROM list_kuf ;
        SIZE 18,26 ;
        DEFAULT 1 ;
        COLOR SCHEME 2 VALID Lkufhundle(choice)

        ACTIVATE WINDOW kufwindow

  READ CYCLE MODAL
  RELEASE WINDOW  kufwindow

  if curkuf#0
    =prepfn()
  endif
Return


FUNCTION Lkufhundle
Parameter vib
  clear read
  curkuf=list_kuf(vib,2)
  *занесем выбор в полосу редактирования
  kuftxt=list_kuf(vib,1)
  *откроем преподавателя и очистим его выборы
  marazm(SAVEOBJ,2)=''
  marazm(SAVEOBJ,1)=0
*  predtxt=[]
*  curpred=0
```

```
RETURN


function predfn
  *Создадим и заполним список предметов по формату имя-код
  select a
  if WasChanged(1)
    pack
    WasChanged(1)=.f.
  endif
  if reccount()=0
    =bell()
    wait [База данных по предметам пуста! Заполните] window;
         NOWAIT
  endif
  dimension list_pred(reccount(),2)
  for _i=1 to reccount()
    go _i
    list_pred(_i,1)=name
    list_pred(_i,2)=code
  endfor
  select i
  aa=irand(1,5)
  cc=irand(1,50)

        DEFINE WINDOW predwindow ;
                FROM INT((SROW()-19)/2),INT((SCOL()-49)/2) ;
                TO INT((SROW()-19)/2)+18,INT((SCOL()-49)/2)+48 ;
                TITLE "Окно выбора предмета" ;
                FOOTER "_Click there_" ;
                NOFLOAT ;
                NOCLOSE ;
                SHADOW ;
                DOUBLE ;
                COLOR SCHEME 4
        ACTIVATE WINDOW predwindow NOSHOW
  @ 3,1 GET choice ;
        PICTURE "@&N" ;
        FROM list_pred ;
        SIZE 14,45 ;
        DEFAULT 1 ;
        COLOR SCHEME 9 VALID Lpredhundle(choice)
  @ 0,16 GET droblevel ;
        PICTURE "@*RVN   Лекция; Практичка;Лабораторка" ;
        SIZE 1,15,0 ;
        DEFAULT 3 ;
        COLOR SCHEME 8

        ACTIVATE WINDOW predwindow
  READ CYCLE MODAL
  RELEASE WINDOW predwindow

  if marazm(SAVEOBJ,1)#0
    &&если выбран, то находим эту пару в базе и меняем
    &&в базе, как и на экране
    select i
```

```
    DO CASE
      CASE marazm(SAVEOBJ,3)==1&&лекция
        findpar=curpot*100
      CASE marazm(SAVEOBJ,3)==2&&практ
        findpar=curpot*100+curgrp*10
      CASE marazm(SAVEOBJ,3)==3&&лаба
        findpar=curpot*100+curgrp*10+getpg(SAVEOBJ)
    ENDCASE
    locate for para==week*100+day*10+getnumber(SAVEOBJ) ;
               .and. gruppa==findpar
    replace predmet with marazm(SAVEOBJ,1)
    show get marazm(SAVEOBJ,2)
    *не надо ли отобразить в заблокированном окошке?
    if marazm(SAVEOBJ,3)#3&& если это не лабораторка
      if getpg(SAVEOBJ)==1
        marazm(SAVEOBJ+15,2)=marazm(SAVEOBJ,2)
        show get marazm(SAVEOBJ+15,2) disable
        &&*   ┌=====T=====¬
        show get marazm(SAVEOBJ+16,2) disable
        &&*   ¦=====+=====¦
        show get marazm(SAVEOBJ+17,2) disable
        &&*   L=====¦=====-
      else
        marazm(SAVEOBJ-15,2)=marazm(SAVEOBJ,2)
        show get marazm(SAVEOBJ-15,2) disable
        show get marazm(SAVEOBJ-14,2) disable
        show get marazm(SAVEOBJ-13,2) disable
      endif
    else
      if getpg(SAVEOBJ)==1
        if marazm(SAVEOBJ+15,1)=0
          for _i=15 to 17
            marazm(SAVEOBJ+_i,2)=[]
            show get marazm(SAVEOBJ+_i,2) enable
          endfor
        endif
      else
        if marazm(SAVEOBJ-15,1)=0
          for _i=15 to 13 step -1
            marazm(SAVEOBJ-_i,2)=[]
            show get marazm(SAVEOBJ-_i,2) enable
          endfor
        endif
      endif
    endif
  endif
Return

  *а не перебрали ли мы часов - вставить обработку потом

Function lpredhundle
Parameter vib
  *Прежде всего - несколько проверок
  *1 - если это лекция, имеем ли мы право на
  *замену - нет ли пр или лаб
  *в этот день на этой паре на этой неделе
```

```
*(gruppa N(5,0),para N(3,0),predmet N(4,0),
*audit N(4,0),prepod N(4,0))
*код пары формируется следующим образом :
* неделя - 1 числитель,2 знаменатель  -
* день -1 .. 5                        -
* номер пары - 1..5                   -
* code=week*100+day*10+number
*код группы = код потока*100+номер группы*10+номер подгруппы
select i
*Один очень нескромный вопрос - а если мы изменяем-удаляя -
*что делать ? Вероятно, надо удалить старое и вставить новое
DO CASE
  CASE droblevel==1
    *поищем лекцию
    locate for curpot*100==gruppa .and. ;
               para==week*100+day*10+getnumber(SAVEOBJ)
    if FOUND()
      *заменяем одну лекцию другой
      clear read
      marazm(SAVEOBJ,1)=list_pred(vib,2)
      *занесем выбор в полосу редактирования
      marazm(SAVEOBJ,2)=list_pred(vib,1)
      *marazm(SAVEOBJ,3)=1
    else
      *лекции нет - это новая и есть опасность наложения на
      *практички или лабораторки
      for _i=1 to m.grpcount
        for _j=0 to 2
          locate for para==week*100+day*10+;
                     getnumber(SAVEOBJ) .and. ;
                     gruppa==curpot*100+_i*10+_j
          if FOUND()
            wait window nowait [Это не может быть лекция]+;
                 [ - есть подгруппы, ]+;
                 [в которых эта пара занята]
            droblevel=2
            show get droblevel
            return
          endif
        endfor
      endfor
      *если нет пересечений - заполняем  !
      *заносим в базу пустышулечку not a'la SQL
      insert blank
      replace para with week*100+day*10+getnumber(SAVEOBJ)
      replace gruppa with curpot*100
      clear read
      marazm(SAVEOBJ,1)=list_pred(vib,2)
      *занесем выбор в полосу редактирования
      marazm(SAVEOBJ,2)=list_pred(vib,1)
      marazm(SAVEOBJ,3)=1
      *заставим выбрать заново аудиторию?   -   уточнить
      *.........
    endif
  CASE droblevel==2
    *поищем практичку
```

```
  locate for curpot*100+curgrp*10==gruppa .and. ;
            para==week*100+day*10+getnumber(SAVEOBJ)
  if FOUND()
    *заменяем одну практичку другой
    clear read
    marazm(SAVEOBJ,1)=list_pred(vib,2)
    *занесем выбор в полосу редактирования
    marazm(SAVEOBJ,2)=list_pred(vib,1)
    *marazm(SAVEOBJ,3)=2
  else
    *практики нет - это новая и есть опасность
    *     наложения на лабораторки
    for _j=1 to 2
      locate for para==week*100+day*10+getnumber(SAVEOBJ);
                .and. gruppa==curpot*100+curgrp*10+_j
      if FOUND()
        wait window nowait [Это не может быть практика]+;
             [ - есть подгруппы, ]+;
             [в которых эта пара занята]
        droblevel=3
        show get droblevel
        return
      endif
    endfor
    *А вдруг на этом месте была лекция?
    locate for para==week*100+day*10+getnumber(SAVEOBJ);
              .and. gruppa==curpot*100
    if !FOUND()
      *если нет пересечений - заполняем  !
      insert blank
      replace para with week*100+day*10+getnumber(SAVEOBJ)
    endif
    replace gruppa with curpot*100+curgrp*10
    clear read
    marazm(SAVEOBJ,1)=list_pred(vib,2)
    *занесем выбор в полосу редактирования
    marazm(SAVEOBJ,2)=list_pred(vib,1)
    marazm(SAVEOBJ,3)=2
    if FOUND()
      SAVEOBJ=SAVEOBJ+2
      =audfn()
      SAVEOBJ=SAVEOBJ-2
    endif
  endif
CASE droblevel==3
  locate for curpot*100+curgrp*10+getpg(SAVEOBJ)==gruppa;
            .and. para==week*100+day*10+getnumber(SAVEOBJ)
  if !FOUND()
    *А вдруг на этом месте была лекция?
    locate for para==week*100+day*10+getnumber(SAVEOBJ);
          .and. gruppa==curpot*100
    if FOUND()
      replace gruppa with curpot*100+curgrp*10+;
               getpg(SAVEOBJ)
    else
      *А вдруг на этом месте была практичка?
```

```
          locate for para==week*100+day*10+getnumber(SAVEOBJ);
                   .and. gruppa==curpot*100+curgrp*10
          if FOUND()
            replace gruppa with curpot*100+curgrp*10+;
                    getpg(SAVEOBJ)
          else
            *если нет пересечений - заполняем  !
            insert blank
            replace para with week*100+day*10+getnumber(SAVEOBJ)
            replace gruppa with curpot*100+curgrp*10+;
                    getpg(SAVEOBJ)
          endif
        endif
      endif
      clear read
      marazm(SAVEOBJ,1)=list_pred(vib,2)
      *занесем выбор в полосу редактирования
      marazm(SAVEOBJ,2)=list_pred(vib,1)
      marazm(SAVEOBJ,3)=3
      if FOUND()
        SAVEOBJ=SAVEOBJ+2
        =audfn()
        SAVEOBJ=SAVEOBJ-2
      endif
  endcase
RETURN



PROCEDURE prep_rasp
*Вывод расписания для преподавателя в файл

  emoe=getfilename()
  handle = FCREATE(emoe) && Открытие  файла
  IF handle < 0               && Ошибка при создании файла
    DO CASE                     && Определение ошибки
      CASE FERROR() = 4
                reason = 'Много файлов открыто'
          CASE FERROR() = 5
                reason = 'Доступ невозможен'
          CASE FERROR() = 8
                reason = 'Нет памяти'
      CASE FERROR() = 29
                reason = 'Нет места на диске'
          CASE FERROR() = 31
                reason = 'Общий сбой'
        ENDCASE
        *** Вывод ошибки ***
        WAIT WINDOW 'Файл не создан: '+ reason NOWAIT
        RETURN
  ENDIF
  select h&&8
  if WasChanged(8)
    pack
    WasChanged(8)=.f.
  endif
```

```
if reccount()==0
  =bell()
  wait "Вначале необходимо заполнить базу данных по"+;
       " кафедрам" window NOWAIT
  return
endif
dimension list_kuf(reccount(),2)
copy to array list_kuf


      DEFINE WINDOW  kufwindow ;
             FROM INT((SROW()-20)/2),INT((SCOL()-30)/2) ;
             TO INT((SROW()-20)/2)+19,INT((SCOL()-30)/2)+29 ;
             TITLE "Окно выбора кафедры" ;
             FOOTER "_Click there_" ;
             NOFLOAT ;
             NOCLOSE ;
             SHADOW ;
             DOUBLE ;
             COLOR SCHEME 1

      ACTIVATE WINDOW  kufwindow NOSHOW
@ 0,1 GET choice ;
      PICTURE "@&N" ;
      FROM list_kuf ;
      SIZE 18,26 ;
      DEFAULT 1 ;
      COLOR SCHEME 2 VALID L2kufhundle(choice)

      ACTIVATE WINDOW kufwindow

READ CYCLE MODAL

if curkuf=0
  return
endif
*В список преподавателей занесем только тех, которые
*работают на данной кафедре

select c
* prepod (code N(4,0), kafedra N(3,0), fio C(65), rank
* N(2,0), zvan N(2,0))
mcount=0
locate for kafedra==curkuf
DO WHILE FOUND()
  mcount=mcount+1
  CONTINUE
ENDDO
if mcount==0
  =bell()
  wait [На данной кафедре никто не работает! ]+;
       [Заполните базу] window NOWAIT
  curkuf=0
  kuftxt=[]
  return
endif
```

```
dimension list_prep(mcount,2)
locate for kafedra==curkuf
mcount=0
DO WHILE FOUND()
  mcount=mcount+1
  list_prep(mcount,1)=getprepname(fio,zvan)
  list_prep(mcount,2)=code
  CONTINUE
ENDDO
select i
      DEFINE WINDOW prepwindow ;
              FROM INT((SROW()-20)/2),INT((SCOL()-30)/2) ;
              TO INT((SROW()-20)/2)+19,INT((SCOL()-30)/2)+29 ;
              TITLE "Выбор преподавателя" ;
              FOOTER "_Click there_" ;
              NOFLOAT ;
              NOCLOSE ;
              SHADOW ;
              DOUBLE ;
              COLOR SCHEME 1
RELEASE WINDOW  kufwindow

      ACTIVATE WINDOW prepwindow NOSHOW
@ 0,1 GET choice ;
      PICTURE "@&N" ;
      FROM list_prep ;
      SIZE 18,26 ;
      DEFAULT 1 ;
      COLOR SCHEME 2 VALID l2prephundle(choice)

ACTIVATE WINDOW prepwindow
READ CYCLE MODAL
RELEASE WINDOW prepwindow
if curprep=0
  return
endif
wait window nowait [Генерация расписания...]
=FPUTS(handle,[             Расписание])
=FPUTS(handle,[Кафедра       : ]+ALLTRIM(kuftxt))
=FPUTS(handle,[Преподаватель : ]+ALLTRIM(preptxt))
select i
for _d=1 to 5
  =FPUTS(handle,[])
  =FPUTS(handle,[])
  =FPUTS(handle,[       ]+days_arr(_d))
  for _p=1 to 5
    =FPUTS(handle,[])
    =FPUTS(handle,[       ]+STR(_p,1)+[ пара])
    for _w=1 to 2
      =FPUTS(handle,week_arr(_w))
      select i
      locate for prepod==curprep .and. para==_w*100+_d*10+_p
      if FOUND()
        select a
        locate for code==i.predmet
        mmm=alltrim(name)
```

```
          select d
          locate for code==int(i.gruppa/100)
          select g
          locate for code==d.special
          nnn=alltrim(name)+[,]+STR(d.kurs,1)+[ курс]
          select i
          if mod(gruppa,100)==0&&lect
            mmm=mmm+[, лекция]
          else
            if mod(gruppa,10)==0&&pract
              mmm=mmm+[, практичка]
              nnn=nnn+[, ]+STR(int(mod(gruppa,100)/10),1)+;
                  [ группа]
            else
              mmm=mmm+[, лабораторка]
              nnn=nnn+[, ]+STR(int(mod(gruppa,100)/10),1)+;
                      [ группа, ]+;
                      STR(mod(gruppa,10),1)+[ подгруппа]
            endif
          endif
          =FPUTS(handle,mmm)
          =FPUTS(handle,nnn)
          select b
          locate for code==i.audit
          =FPUTS(handle,"Ауд. "+STR(aud,3)+[, корп. ]+;
                        STR(korpus,1))
        else
          =FPUTS(handle,[])
          =FPUTS(handle,[])
          =FPUTS(handle,[])
        endif
      endfor
    endfor
  endfor
  select i
  =FCLOSE(handle)       && Close the file
  wait clear
  DEFINE WINDOW wingets FROM 1,0 TO 24,79 ;
                CLOSE FLOAT SHADOW SYSTEM ;
                COLOR SCHEME 8 TITLE ' Расписание для преподавателя '

  MODIFY FILE (emoe) WINDOW wingets && View the file
  RELEASE WINDOW  &&Удаление окна
RETURN


FUNCTION getfilename
        DEFINE WINDOW wfilename ;
                FROM INT((SROW()-11)/2),INT((SCOL()-57)/2) ;
                TO INT((SROW()-11)/2)+10,INT((SCOL()-57)/2)+56 ;
                FLOAT ;
                NOCLOSE ;
                SHADOW ;
                DOUBLE ;
                COLOR SCHEME 5
        ACTIVATE WINDOW wfilename NOSHOW
```

```
@ 6,13 EDIT filename ;
        SIZE 1,28,65 ;
        DEFAULT " "
@ 8,24 GET knopka ;
        PICTURE "@*HT \<Ok" ;
        SIZE 1,6,1 ;
        DEFAULT 1
@ 0,13 SAY "Введите имя файла, в который"
@ 1,16 SAY "будет осуществлен вывод"
@ 2,13 SAY "расписания для корректировки"
@ 3,16 SAY "и дальнейшей распечатки"
@ 4,13 SAY "Файл не должен существовать."

        ACTIVATE WINDOW wfilename

READ CYCLE MODAL

RELEASE WINDOW wfilename

RETURN alltrim(filename)


FUNCTION L2kufhundle
Parameter vib
  clear read
  curkuf=list_kuf(vib,2)
  *занесем выбор в полосу редактирования
  kuftxt=list_kuf(vib,1)
  *откроем преподавателя и очистим его выборы
  preptxt=[]
  curprep=0
RETURN

Function l2prephundle
Parameter vib
  clear read
  curprep=list_prep(vib,2)
  *занесем выбор в полосу редактирования
  preptxt=list_prep(vib,1)
RETURN

*Формирование кода

procedure codeguard
  *ищем запись с нулевым кодом
  for _i=1 to reccount()
    go _i
    if empty(code)
      *ищем по всей базе незанятый код
      tr=.f.&&.t. - когда найдем
      num=1
      do while !tr
        locate for code==num
        if !found() &&код уникальный
          tr=.t.
          loop
```

```
        endif
        num=num+1
      enddo
      go _i
      replace code with num
    endif
  endfor
Return

*Ну вот и всё.............................
```

Литература:

1. Компьютер Пресс N2 1996г.

2. Компьютер Пресс N4 1996г.

3. Компьютер Пресс N5 1996г.

4. Computer Week Москва N38(196) 1995г.

5. Computer Week Москва N4(210) 1996г.

6. Computer Week Москва N17(223) 1996г.

7. Computer Week Москва N18(224) 1996г.

8. PC Magazine russian edition спецвыпуск N2(41) 1995г.

9. PC Magazine russian edition N6(34) 1995г.

10. Компьютерра N15(142) 1996.