



Кафедра конструирования
и производства радиоаппаратуры

УТВЕРЖДАЮ
Заведующий кафедрой КИПР

_____ **В.Н. ТАТАРИНОВ**

“ ___ ” _____ 2012 г.

Программирование с использованием записей

Лабораторная работа по дисциплинам «Информатика» для студентов специальностей 211000.62 «Конструирование и технология электронных средств» (бакалавриат) и 162107.65 «Информатика и информационные технологии» (специалитет)

Разработчик:
Доцент кафедры КИПР

_____ **Ю.П. Кобрин**

Томск 2012

Оглавление

1 Цели работы.....	3
2 Порядок выполнения работы	3
3 Контрольные вопросы.....	3
4 Отчетность	3
5 Применение записей при разработке баз данных	4
5.1 Постановка задачи.....	4
5.2 Записи	4
6 Учебный пример.....	7
6.1 Условия задания.....	7
6.2 Основная программа	7
6.3 Модуль для программы работы с картотекой книг	8
7 Индивидуальные задания	14
7.1 Требования к программе	14
7.2 Варианты заданий	15
Список рекомендуемой литературы	16

1 Цели работы

- Освоение работы с данными типа запись в **Borland Pascal**.
- Получение практических навыков в разработке программ, использующих запись.

2 Порядок выполнения работы

В ходе выполнения этой работы следует:

- 1) Изучить описание лабораторной работы, обратив особое внимание на правила описания и использования записей в **Borland Pascal**. При необходимости использовать дополнительную литературу [1 - 8].
- 2) Ответить письменно на контрольные вопросы.
- 3) Войти в свой личный каталог, и настроить интегрированную среду **Borland Pascal** для последующей работы. Записать файл конфигурации в личный каталог.
- 4) Получить индивидуальное задание и провести разработку алгоритма и программы в соответствии с требованиями модульного и структурного программирования.
- 5) Внимательно проштудировать учебный пример и по возможности максимально использовать его в своей задаче.
- 6) Ввести и отладить разработанную программу.
- 7) Продемонстрировать работоспособность программы для различных вариантов исходных данных.
- 8) Оформить отчет по лабораторной работе и защитить его у преподавателя.

3 Контрольные вопросы

Ответьте письменно на следующие контрольные вопросы:

- 1) Что такое запись?
- 2) В каких случаях целесообразно использовать данные типа запись?
- 3) Как объявляются тип и переменные типа запись?
- 4) Поясните правила объявления типизированных констант записи.
- 5) Для чего используются составные имена и из каких компонентов они состоят?
- 6) Каково назначение и форма оператора присоединения **with**?
- 7) Какова форма обращения к полям записей с оператором и без использования оператора **with**?
- 8) Какие преимущества дает оператор присоединения?
- 9) Какие операции допустимы над полями записей?
- 10) В чем отличие записей от массивов?

4 Отчетность

Отчет должен быть выполнен в соответствии с [9] и состоять из следующих разделов:

- 1) Тема и цель работы.
- 2) Индивидуальное задание.
- 3) Схема алгоритма решения задачи.
- 4) Текст программы и вводимые тестовые исходные данные.
- 5) Откомпилированный текст программы-заготовки (в электронном виде).
- 6) Результаты выполнения программы.

- 7) Ответы на контрольные вопросы.
- 8) Выводы.

При защите отчета по работе для получения зачета студент должен:

- уметь отвечать на контрольные вопросы;
- обосновать структуру выбранного алгоритма и показать его работоспособность;
- уметь пояснять работу программы;
- продемонстрировать *навыки работы в среде Borland Pascal*.

5 Применение записей при разработке баз данных

5.1 Постановка задачи

Большинство современных информационных технологий, задач экономики и управления базируется на концепции **баз данных** [БД]. По определению, БД - это **совокупность взаимосвязанных файлов**, в которых хранятся определенным образом организованные данные, адекватно отражающие соответствующую предметную область (например, информацию о моделях электрорадиоэлементов, материалах, проектных документах и т.д.). Невозможно представить себе САПР РЭС, которая не опиралась бы на БД, хранящую используемые при проектировании данные.

Как правило, файлы БД содержат большое число однотипных записей. Каждая **запись¹ объединяет под одним именем логически связанные данные разного типа, относящиеся к одному объекту²**. Запись состоит из фиксированного числа объектов, называемых ее полями. **Поле** – это переменная определенного типа. Объем памяти, необходимый для хранения записи, складывается из длин полей.

Записи - наиболее общий и гибкий тип данных Паскаля. Следовательно, задача разработки программ, позволяющих работать с записями, представляется весьма жизненной.

5.2 Записи

Определить запись в Паскале можно следующим образом:

```

Типе           {Раздел объявлений нестандартных пользовательских типов данных}
<Имя типа записи> = record           {Заголовок объявления типа записи}
  <идентификатор поля>: <тип поля>;   } {Список определений
  .....                               } элементов записи - полей}
  <идентификатор поля>: <тип поля>   }
end;           {Конец определения записи}

```

Элементами (полями) записи могут быть любые стандартные типы данных, и, кроме того, определенные пользователем множества, файлы, другие записи и массивы.

Экземпляр записи можно создать в разделе описания переменных³

¹ Пример – карточка в некоторой картотеке или каталоге (например, в регистратуре, библиотеке и т.п.).

² Например, записи о владельце автомобиля могут содержать поля: номер, марка машины, год выпуска, год регистрации, фамилия владельца, его адрес и т.п.

³ В несложных случаях записи, как и массивы, допускается определять и непосредственно в разделе описания переменных.

Var

<Идентификатор переменной, ...>: <Имя типа записи>;

Например, в БД Госавтоинспекции возможны записи такого вида:

Type {раздел объявления пользовательских типов данных}

tData = record {заголовок типа данных: запись даты}

Jear: **integer;** {поле № 1, год}

Month: **1..12;** {поле № 2, номер месяца}

Day: **1..31;** {поле № 3, день}

end;

tAuto = record {заголовок типа данных: запись об автомобиле}

GosNomer: **string[11];**{поле № 1, номер государственной регистрации}

Marka: **string[20];**{поле № 2, марка автомобиля}

God, GodReg: **tData;** {одинаковые поля № 3 и №4, записи о датах выпуска и регистрации автомобиля, состоящие из полей **Jear, Month, Day** записи **tData**}

FIO: **string[40];**{поле № 5, фамилия, имя и отчество владельца}

Adres: **string[50];**{поле № 6, адрес владельца}

end; {конец определения записи об автомобиле}

Var {раздел объявления переменных}

Auto: **tAuto;** {экземпляр записи об автомобиле}

TablAuto: **array[1..50] of tAuto;** {таблица (массив) записей об автомобилях}

Обращение к полю записи выполняется с помощью **составного имени**. Первая часть составного имени – это имя записи, вторая часть уточняет имя поля. В качестве разделителя между ними применяется точка. Изменять значения полей можно либо присвоением, либо вводом. Например,

Auto.Marka := 'BA3 21063'; {присвоение полю **Marka** записи **Auto** нового значения}

{ввод нового значения поля **GosNomer** для 5 элемента массива записей **TablAuto**}

Read(TablAuto[5].GosNomer);

Для вложенных полей (подструктур) уточнений имени может быть несколько, например:

Auto.God.Jear := 1999; {присвоение полю **Jear** записи **Auto** нового значения}

Допускается применение оператора присваивания и к записям в целом, если они имеют один и тот же тип. Например, чтобы значения всех полей текущей записи **Auto** запомнить в *i*-м элементе массива записей **TablAuto**, следует записать оператор:

TablAuto[i] := Auto

Для краткого обращения к полям записи удобно использовать оператор присоединения **with**. В операторе **with** к полям одной или более конкретных переменных типа запись можно обращаться, используя только лишь простые идентификаторы полей:

with <Имя записи> do

Оператор; {В этом единственном операторе указывают только имена полей записи, <Имя записи> добавляется к каждому имени поля автоматически}

Для распространения действия оператора присоединения **with** на некоторую группу операторов используют составной оператор (закljučают эту группу операторов в **begin .. end**)

С учетом введенных ранее описаний записи, приведем пример использования оператора **with**:

with Auto.God do {Теперь к полям записи **Auto** достаточно обращаться только по их именам}

```

if Month = 12
  then begin
    Month := 1;
    Year := Year + 1
  end
else Month := Month + 1;

```

Это эквивалентно следующему фрагменту программы:

{Для доступа к полям записи **Auto** используются длинные составные имена}

```

if Auto.God.Month = 12
  then begin
    Auto.God.Month := 1;
    Auto.God.Year := Auto.God.Year + 1
  end
else Auto.God.month := Auto.God.Month + 1;

```

При необходимости, можно использовать более сложный тип записи, содержащий несколько вариантов структур полей [1 - 8]. Любой из вариантов структуры записи может быть активизирован в зависимости от значения поля-переключателя.

В заключение приведем пример еще одной возможности задания начальных значений полей записей – с помощью типизированных констант (переменных с начальными значениями).

Type {Объявления нестандартных типов данных}

```

tPoint = record {запись о координатах точки}
  X, Y: Real
end;

```

```

tVector = array[0..1] of tPoint; {Массив из двух точек}

```

{Перечислимый тип данных - месяцы}

```

tMonth = (Jan, Feb, Mar, Apr, May, Jun, July, Aug, Sep, Oct, Nov, Dec);

```

```

tDate = record {Запись о некоторой дате}

```

```

  Day: 1..31;
  Month: tMonth;
  Year: 1900..2999
end;

```

const {Примеры определения типизированных констант}

```

Origin: tPoint = (X: 0.0; Y: 0.0); {Начальные координаты точки Origin}

```

```

{Координаты начала и конца отрезка прямой Line}

```

```

Line: tVector = ((X: -3.1; Y: 1.5), (X: 5.8; Y: 3.0));

```

```

{Запоминается дата дня рождения SomeDay}

```

SomeDay: tDate = (Day: 11; Month: Oct; Year: 1947);

6 Учебный пример

6.1 Условия задания

Пример 10.1. Разработать программу, работающую с массивом записей (картотекой) книг вида:

Номер книги	Фамилия автора	Название книги
-------------	----------------	----------------

Предусмотреть многократный ввод данных о книгах с помощью простейшего диалога типа «запрос-ответ»; признак окончания ввода данных – ввод пустой строки. Программа должна работать в режимах:

- ввода данных о книгах,
- коррекции введенных ранее данных о книге,
- формирование запроса на поиск и выбор нужных книг;
- удаления записи о книге с заданным номером.

Подпрограммы следует оформить в виде пользовательского модуля.

6.2 Основная программа

Основная программа *Primer10_1* использует подпрограммы и описания данных из специально разработанного для нее пользовательского модуля *Unit10_1*. Она представляет собой бесконечный цикл, последовательно реализующий решение следующих подзадач:

- вывод меню выбора режима работы,
- ввод номера режима работы,
- работа в выбранном режиме.

Program Primer10_1; {Картотека регистрации и поиска книг}

Uses {Подключение библиотечных модулей}

Crt, {Системный модуль с подпрограммами работы с экраном}

Unit10_1; {Пользовательский модуль с подпрограммами и описаниями данных картотеки книг}

begin {Главная программа Primer10_1}

{Устанавливаем цвет фона и текста: желтые символы на голубом фоне}

TextColor(Yellow);

TextBackGround(Blue);

NumbBook := 0; {Начальное количество книг в картотеке}

Repeat {Цикл диалогового выбора и выполнения режимов работы}

ClrScr, {очистка экрана}

{Вывод меню выбора режимов работы}

GoToXY(1, 10);

Writeln(' Картотека регистрации и поиска книг');

Writeln;

Writeln(' Главное меню');

Writeln;

Writeln(' 1: Ввод информации о книгах');

Writeln(' 2: Коррекция записи о книге');

```

Writeln(' 3: Удаление записи о книге');
Writeln(' 4: Выбор и вывод информации о книгах');
Writeln(' 5: Закончить работу');
Writeln;
Write(' Введите номер режима работы: ');
Readln(Otvet);
Writeln;

```

```

    {Выполнение выбранного режима работы}
    {Светло-серые символы на голубом фоне}

```

```

TextColor(LightGray);

```

```

Case Otvet of

```

```

    1: Vvod; {Ввод и запись информации о книгах}
    2: Korr; {Коррекция записи о книге с номером Number}
    3: Udal; {Удаление записи о книге с номером Number}
    4: Out; {Вывод информации обо всех книгах на экран}
    5: Halt {Закончить работу}

```

```

else begin

```

```

    Writeln('Неверно выбран режим работы.', #7);

```

```

    Pause {Временная остановка}

```

```

end

```

```

end; { Case }

```

```

TextColor(Yellow); {Восстанавливаем желтые символы на голубом фоне}

```

```

until false; {Конец "бесконечного" цикла выбора и выполнения режимов работы}

```

```

end. {Конец головной программы}

```

6.3 Модуль для программы работы с картотекой книг

```

{$F+,O+,R+} {Директивы трансляции модуля}

```

```

Unit Unit10_1; {Модуль с подпрограммами и описаниями данных картотеки книг}

```

```

Interface {начало интерфейсной секции}

```

```

{===== Раздел объявлений констант =====}

```

```

Const

```

```

{===== Ограничения =====}

```

```

    MaxBooks = 25; {Максимальная длина картотеки – массива с данными о книгах}

```

```

    LenAutor = 15; {Максимальная длина ФИО автора книги}

```

```

    LenNazv = 15; {Максимальная длина названия книги}

```

```

{===== Раздел объявлений типов нестандартных данных =====}

```

```

type

```

```

    tBook = record {запись о книге}

```

```

        Number: integer; {поле с порядковым номером книги}

```

```

        Autor: string[LenAutor]; {поле с ФИО автора книги}

```

```

        Nazv: string[LenNazv] {поле с названием книги}

```

```

    end;

```

```

{===== Раздел объявлений переменных =====}

```


Var

Books: array [1 .. MaxBooks] of tBook; {библиотечный массив с записями о книгах}
Book, FindBook: tBook; {экземпляры записи о книге}
NumbBook: integer; {количество книг в картотеке}
Error: integer; {код ошибки при работе, 0 - нет ошибки}
Otv: integer; {номер выбранного режима при диалоге}

{===== Раздел объявлений функций и процедур =====}

procedure Pause; {Временная остановка, до нажатия Enter}

{Функция YesNo - возвращает True, если ответ Y, Д, д или Enter, иначе возвращает False}

function YesNo: Boolean;

{Функция VvodBook - Ввод информации о книге с номером N}

{Возвращает True, если ввод успешен, False, если во введенном поле Avtor - пустая строка}

function VvodBook(N: integer): Boolean;

procedure VvodFindBook; {Ввод информации о параметрах поиска книги}

procedure Vvod; {Ввод и запись информации о книгах в картотеку}

procedure Korr; {Коррекция записи о книге с номером Number}

procedure Udal; {Удаление записи о книге с номером Number}

procedure OutBook; {Вывод информации о книге}

procedure Out; {Вывод информации о всех книгах на экран}

{===== Секция реализации функций и процедур =====}

Implementation

Uses {Подключение библиотечных модулей}

Crt; {Системный модуль процедур работы с экраном}

procedure Pause; {Временная остановка}

begin

writeln;

write('Для продолжения нажмите <Enter>', #7); {#7 - звуковой сигнал}

Readln

end; {---Pause---

{--- Функция YesNo ---}

function YesNo: Boolean; {Возвращает True если ответ Y, Д, д, Enter, False если иначе}

var

Otv: char; {Символ с начальной буквой ответа}

begin

Write(' (y/n): '); Otv := ReadKey; {Ввод первого символа ответа}

Otv := UpCase(Otv); {Преобразование к верхнему регистру}

if (Otv = 'Y') or (Otv = 'Д') or (Otv = 'д') or (Otv = #13)

then YesNo := True {если положительный ответ}

else YesNo := False; {если отрицательный ответ}

Writeln

end; {---YesNo---

```

function VvodBook(N: integer): Boolean; {Ввод информации о книге с номером N}
{Возвращает True, если ввод успешен, или False, если во введенном поле Autor - пустая строка}
begin
  with Book do
    begin
      Writeln;
      Writeln('Порядковый номер книги = ', N);
      Writeln;
      Write('Автор книги : '); Readln(Autor);

      if Autor = "
        then begin {информация о книге не введена}
          VvodBook := False;
          Writeln;
          Writeln('Ввод информации завершен.');
          Pause; {Временная остановка}
          Exit {Выход}
        end;

      VvodBook := True; {информация о книге введена}

      Number := N; {Запоминаем порядковый номер книги}
      Write('Название книги: '); Readln(Nazv);
      Writeln
    end
  end; {---VvodBook---}

  {--- процедура VvodFindBook ---}
procedure VvodFindBook; {Ввод информации о параметрах поиска книги}
var
  St: string; {вспомогательная строка для ввода номера книги}
begin
  with FindBook do
    begin
      WriteLn('Найти книги с параметрами: ');
      repeat
        {Error = 0, если порядковый номер книги в допустимых пределах, Error > 0, если нет}
        Error := 0;

        Writeln;
        Write('Порядковый номер книги');
        Write(' (Нажмите Enter, если искать книги с любым номером): ');
        ReadLn(St);
        if St = "
          then Number := -1
          else begin
            Val(St, Number, Error); {Преобразуем текст St в число Number}
            if (Error > 0) or (Number < 0) or (Number > NumbBook)
              then begin
                GoToXY(1, 5);
                writeln(' Недопустимый порядковый номер книги!', #7);
                Error := 1;
                Pause
            end
          end
        until Error = 0;
      end
    end
  end;

```

```

        end
    end
until Error = 0;
Write('Автор книги ');
Write('(Нажмите Enter, если искать книги любого автора): ');
Readln(Autor);
Write('Название книги ');
Write('(Нажмите Enter, если искать книги с любым названием): ');
Readln(Nazv);
Writeln
end
end; {---VvodFindBook---}

{--- процедура Vvod ---}
procedure Vvod; {Ввод и запись информации о книгах в файл}
begin
    ClrScr;
    Repeat {цикл ввода информации по книгам}
        if NumbBook + 1 > MaxBooks
            then begin
                writeln('Картотека книг переполнена!', #7);
                Pause;
                Break {Выход из цикла ввода, если картотека переполнена}
            end;
        if VvodBook(NumbBook + 1)
            then begin
                {Индекс номера книги перемещаем на очередную книгу, добавляемую в конец массива!}
                NumbBook := NumbBook + 1;
                {если запись Book о книге сформирована, сохраняем ее в массиве Books}
                Books[NumbBook] := Book
            end
            else Break {Выход из цикла ввода, если запись о книге не сформирована}
        until False;
    end; {---Vvod ---}

    {--- процедура Korr ---}
procedure Korr; {Коррекция записи о книге введенным номером NBook}
var
    NBook: integer; {Порядковый номер книги в файле}
    flag: Boolean; {флаг ошибки задания номера книги}
begin
    ClrScr;
    if NumbBook = 0
        then begin
            GoToXY(1, 5); Writeln('Список книг пуст..', #7);
            Pause;
            Exit
        end;

    repeat
        Writeln;
        Write('Введите порядковый номер модифицируемой книги: ');
        Readln(NBook);

```

{flag = true, если порядковый номер книги в допустимых пределах, flag = false, если нет}
flag := (NBook >= 0) and (NBook <= NumbBook);

if not flag
then begin
 GoToXY(1,5); writeln(' Недопустимый порядковый номер книги!!', #7);
 Pause
end
until flag;

Book := Books[NBook]; {Читаем из картотеки запись о книге с номером NBook}

writeln;

OutBook; {Выводим на экран информацию о книге с номером NBook}

writeln;

Write('Корректировать данные? ');

if not YesNo

then Exit; {Выход, если не корректировать}

{если запись о книге скорректирована, то сохраняем ее в картотеке вместо старой записи}

if VvodBook(NBook)

then Books[NBook] := Book;

end; {---Korr---}

{--- процедура Udal ---}

procedure Udal; {Удаление записи о книге с номером Number}

var

N, NBook: integer; {Порядковый номер книги в файле}

flag: Boolean; {флаг ошибки задания номера книги}

begin

ClrScr;

if NumbBook = 0

then begin

GoToXY(1, 5); Writeln('Список книг пуст..', #7);

Pause;

Exit

end;

repeat

Writeln;

Write('Введите порядковый номер удаляемой книги: ');

Readln(NBook);

{flag = true, если порядковый номер книги в допустимых пределах, flag = false, если нет}

flag := NBook > 0;

if not flag

then begin

GoToXY(1, 5);

writeln(' Недопустимый порядковый номер книги!!', #7);

Pause

end

until flag;

{Поиск книги с порядковым номером NBook}

```

for N := 1 to NumbBook do
  if Books[N].Number = NBook
    then begin
      Book := Books[N]; {Читаем из картотеки запись о книге с номером NBook}
      flag := false;
      Break
    end;
  if flag
    then begin
      GoToXY(1, 5);
      writeln('Книги с порядковым номером ', NBook, ' в картотеке нет! ',
        #7);
      Pause;
      Exit
    end;

writeln;
OutBook; {Выводим на экран информацию о книге с номером NBook}
writeln;
Write('Удалять данные об этой книге? ');
if not YesNo
  then Exit; {Выход, если не удалять}

  {Цикл удаления записи о книге и «сжатия» массива}
if (NumbBook > 1) and (N < NumbBook)
  then for NBook := N to NumbBook - 1 do
    Books[NBook] := Books[NBook + 1];
NumbBook := NumbBook - 1 {Число записей о книгах уменьшаем на единицу}

end; {---Udal---}

  {--- процедура OutBook ---}
procedure OutBook; {Вывод на экран информации о книге Book}
begin
  with Book do
    writeln(Number:5, ' ', Autor: LenAutor, ' ', Nazv: LenNazv)
end; {---OutBook---}

  {--- процедура Out ---}
procedure Out; {Вывод информации о всех книгах на экран}
var
  NBook: integer; {Порядковый номер книги в файле}
begin
  ClrScr; {Очистка экрана}
  if NumbBook = 0
    then begin
      GoToXY(1, 5); Writeln('Список книг пуст..', #7);
      Pause;
      Exit
    end;

VvodFindBook; {Ввод информации о параметрах поиска книги}

```

```

        {Вывод заголовка списка книг}
Writeln;
writeln('Гомер', 'Автор': LenAutor+2, 'Название книги': LenNazv+2);
Writeln;
for NBook := 1 to NumbBook do {Просматриваем картотеку}
  with Book do
    begin
      Book := Books[NBook]; {чтение информации о книге Book}
      {Ищем книгу с заданным номером}

      if (Number = FindBook.Number) or (FindBook.Number < 0)
        then begin {книга с заданным номером найдена}
          {Выводим на экран данные о книге, если Автор и название произвольные}
          if (FindBook.Autor = "") and (FindBook.Nazv = "")
            then OutBook
              else if FindBook.Autor = ""
                then begin {Ищем книгу с заданным названием}
                  if Pos(FindBook.Nazv, Nazv) <> 0
                    then OutBook
                  end
                else if FindBook.Nazv = ""
                  {Ищем книгу с заданным автором}
                  then if Pos(FindBook.Autor, Autor) <> 0
                    then OutBook
                end
              end
            end
          end;
          Pause {Временная остановка}
        end;{--- Out ---}

    end. {Конец модуля Unit10_1}

```

7 Индивидуальные задания

7.1 Требования к программе

При выполнении задания в качестве заготовки программы разумно максимально использовать рассмотренный выше учебный пример. Для этого следует скопировать файлы *Prim10_1.pas* и *Unit10_1.pas* в свой каталог, переименовать их и взять затем за основу разрабатываемой программы.

Тестовый пример формируется самостоятельно. Он должен позволять оценить работоспособность программы.

При составлении программы по заданию предусмотреть:

- использование структурного и модульного подхода;
- простейший диалог типа «запрос-ответ» при вводе данных;
- многократный ввод данных при исполнении программы, т.е. возможность повторной обработки при иных исходных данных. Признак окончания ввода данных – ввод пустой строки;
- вывод результатов в удобном для пользователя виде;
- освоить методику поиска причин и исправления ошибок, а также трассировки программы «по шагам»⁴ по тестовому примеру.

⁴ Используйте функциональные клавиши F7 или F8.

- подпрограммы, составленные при выполнении задания, должны быть оформлены в виде пользовательского модуля.
- программа должна работать в режимах:
 - ввода исходных данных,
 - формирование запроса на поиск и выбор требуемых элементов массива записей;
 - коррекции введенных ранее записей;
 - удаления записи с заданным номером.

7.2 Варианты заданий

Создать программу, работающую с массивом записей:

1. Самолеты

Наименование типа	Фамилия конструктора	Год выпуска	Количество кресел	Грузоподъемность, т
-------------------	----------------------	-------------	-------------------	---------------------

2. Расчет движения

Наименование воздушной линии	Тип самолета	Количество рейсов	Налет, тыс. км	Пассажирооборот, человеко·км
------------------------------	--------------	-------------------	----------------	------------------------------

3. Перевозки

Тип самолета	Номер борта	Количество рейсов	Налет в часах	Налет, тыс. км
--------------	-------------	-------------------	---------------	----------------

Расписание

Номер рейса	Наименование рейса	Тип самолета	Стоимость билета	Протяженность линии
-------------	--------------------	--------------	------------------	---------------------

4. Сооружения аэропорта

Наименование	Площадь	Этажность	Год сооружения	Стоимость
--------------	---------	-----------	----------------	-----------

5. Ремонт аэродромных сооружений

Наименование	Шифр	Вид ремонта	Стоимость	Наименование подрядчика
--------------	------	-------------	-----------	-------------------------

6. Кассы авиабилетов

Номер кассы	Ф.И.О. кассира	Количество проданных билетов	Суммарная выручка	Дата продаж
-------------	----------------	------------------------------	-------------------	-------------

7. Характеристики персональных компьютеров

Тип процессора	Тактовая частота	Емкость ОП, Мбайт	Емкость ЖМД, Мбайт	Тип монитора
----------------	------------------	-------------------	--------------------	--------------

8. Города

Наименование	Количество жителей	Площадь, кв. км	Год основания	Количество школ
--------------	--------------------	-----------------	---------------	-----------------

9. Московские мосты

Наименование	Высота	Ширина	Количество опор	Протяженность
--------------	--------	--------	-----------------	---------------

10. Линии московского метро

Наименование	Район линии	Год пуска	Протяженность, км	Количество поездов
--------------	-------------	-----------	-------------------	--------------------

11. Легковые автомобили

Марка	Цвет	Стоимость	Изготовитель	Максимальная скорость
-------	------	-----------	--------------	-----------------------

12. Продажа программных продуктов

Наименование	Фирма изготовитель	Стоимость, тыс. руб	Объем, Мбайт	Количество на складе
--------------	--------------------	---------------------	--------------	----------------------

13. Абонентская плата за телефон

Ф.И.О. абонента	Телефон	Год установки	Количество абонентов	Плата за телефон
-----------------	---------	---------------	----------------------	------------------

14. Детские сады

Наименование детского сада	Номер сада	Количество детей	Район города	Плата за месяц
----------------------------	------------	------------------	--------------	----------------

15. Сотрудники

Ф., И., О.	Табельный номер	Дата рождения	Оклад, тыс. руб.	Стаж
------------	-----------------	---------------	------------------	------

16. Ведомость зарплаты за текущий месяц

Ф., И., О.	Номер отдела	Табельный номер	Количество рабочих часов	Размер зарплаты
------------	--------------	-----------------	--------------------------	-----------------

17. Музеи

Наименование	Назначение	Адрес	Время работы	Стоимость билета
--------------	------------	-------	--------------	------------------

18. Экскурсии

Наименование	Страна	Стоимость	Продолжительность	Транспорт
--------------	--------	-----------	-------------------	-----------

19. Кинофильмы

Наименование кинотеатра	Стоимость билета	Время сеансов	Адрес	Количество мест
-------------------------	------------------	---------------	-------	-----------------

20. Книга - почтой

Наименование книги	Ф.И.О. автора	Номер по каталогу	Издательство	Стоимость книги
--------------------	---------------	-------------------	--------------	-----------------

21. Квартиры

Адрес	Площадь, кв. м	Сторона света	Стоимость 1 кв. м	Этаж	Количество комнат
-------	----------------	---------------	-------------------	------	-------------------

22. Склад товаров

Номер магазина	Наименование товара	Артикул товара	Цена единицы товара	Количество товара
----------------	---------------------	----------------	---------------------	-------------------

23. Телевизоры на складе магазина

Наименование	Фирма изготовитель	Стоимость	Размер экрана	Количество на складе
--------------	--------------------	-----------	---------------	----------------------

24. Холодильники на складе магазина

Наименование	Фирма изготовитель	Стоимость	Емкость камеры	Количество на складе
--------------	--------------------	-----------	----------------	----------------------

Список рекомендуемой литературы

1. Д. Ван Тассел. Стилль, разработка, эффективность, отладка и испытание программ: Пер. с англ. – М.: Мир, 1985. – 332 с.
2. Н. Вирт. Алгоритмы и структуры данных. : Пер. с англ. – М.: Мир, 1989. – 360 с.
3. Основы информатики. Учеб. Пособие / Аладьев В.З., Хунт Ю.Я., Шишаков М.Л. - М.: Информационно-издательский дом "Филинь", 1998. - 496 с.
4. Марченко А.И., Марченко Л.А. Программирование в среде Borland Pascal 7.0. – К.: ЮНИОР, 1998. – 480 с.
5. Зуев Е.А. Программирование на языке Турбо Паскаль 6.0, 7.0. - М: Веста, Радио и связь, 1993. - 384 с.
6. Епанешников А., Епанешников В. Программирование в среде Turbo Pascal 7.0. - М.: "ДИАЛОГ-МИФИ", 1993. - 288 с.
7. Фаронов В.В. Turbo Pascal 7.0. Начальный курс. Учебное пособие.- М.: "НОЛИДЖ", 2001. - 576 с.

8. Фаронов В.В. Turbo Pascal 7.0. Практика программирования. Учебное пособие.- М.: "НОЛИДЖ", 1998. - 432 с.
9. ОС ТУСУР 6.1-97. Работы студенческие учебные и выпускные квалификационные. - Томск: ТУСУР, 1999.- 10 с.