

A sunset scene over a body of water with a large, stylized text overlay. The sky is filled with dark, dramatic clouds, and the sun is low on the horizon, creating a bright glow. The water reflects the colors of the sky. The text is in a light blue, outlined font.

Типизированные файлы  
Тип данных Record

## Назначение записей.

**Тип данных Запись (Record) используется в тех случаях, когда необходимо обрабатывать структурированные данные, которые описывают несколько различных свойств объекта.**

**Например, нам надо использовать следующие данные про наших друзей:**

- 1. Фамилия**
- 2. Имя**
- 3. Адрес**
- 4. Телефон**

**Эти данные имеют разный тип. Но из них можно составить структурированный тип данных – запись.**

# Описание типа данных Record

```
type имя записи = record
    имя поля 1 : тип поля1;
    - - -
    имя поля n : тип поля n ;
end;
```

Например:

## Структура Друзья

```
Фамилия : строка [ 12 ]
Имя      : строка [ 12 ]
Адрес    : строка [ 25 ]
Телефон  : строка [ 9 ]
```

```
type friends = record
    Fam      : string [ 12 ];
    Name     : string [ 12 ];
    Adress   : string [ 25 ];
    Telef    : string [ 9 ];
end;
```

## Составные имена полей

С полями, входящими в запись, можно выполнять те же действия, что и с обычными переменными соответствующего типа.

Для обращения к полям записи используют составные имена, части которых разделены точкой:

**ИМЯ ЗАПИСИ.ИМЯ ПОЛЯ**

**Friends.Fam** - фамилия друга

**Friends.Telef** - телефон друга

Составные имена могут участвовать в выражениях как обычные переменные:

**Friends.Telef:=‘123-45-67’;**

## Работа с элементами записи:

# Использование команды присоединения **With**

Составные имена довольно громоздки.

Чтобы иметь возможность обращаться непосредственно к самому полю в записи, используют команду **With**

**Например:**

```
With имя записи do  
begin  
    действия с полями  
end;
```

```
With drug do  
begin  
    writeln ( 'фамилия' );  
    readln ( fam );  
    writeln ( 'имя' );  
    readln ( name );  
    tel := '276-90-90'  
end;
```

## Использование записей в типизированных файлах.

- 1. В разделе описания типов задать тип данных Record**
- 2. В разделе описания переменных объявить переменную этого типа и файл такого типа.**
- 3. В программе сформировать запись в заданную переменную и потом целиком записать ее в файл.**
- 4. При чтении из файла информацию считывать в эту переменную.**

# Задание № 1

**Написать программу, которая создает типизированный файл содержащий информацию о друзьях, и потом выводит эту информацию на экран из файла. Количество друзей должно запрашиваться в начале программы.**

```
Program Druzya;  
uses crt;  
type friends = record  
    Fam, Name : string[ 12 ];  
    Date : string[ 8 ];  
    Adress : string[ 50 ];  
    Tel : string[ 9 ]  
end;  
Var drug : friends;  
    f : file of friends;  
    i,n : integer;  
Begin  
Clrscr;  
assign (f, 'friends.pas');  
{ Создание файла }  
Rewrite ( f );  
write (‘Укажите количество друзей ’);  
readln (n);
```

```
for i:=1 to n do
  begin
    with drug do
      begin
        writeln ('Введите фамилию ',i,'-го друга ');
        readln (Fam);
        writeln ('Введите имя ',i,'-го друга ');
        readln (name);
        writeln 'Введите адрес ',i,'-го друга ';
        readln (Adress);
        writeln ('Введите телефон ',i,'-го друга ');
        readln (Tel);
      end;
    write (f, drug);
  end;
close ( f );
```

```
writeln('Вывод из файла: ');  
reset ( f );  
while not eof(f) do  
  begin  
    read (f, drug);  
    with drug do  
      begin  
        writeln (Fam);  
        writeln (Name);  
        writeln (Adress);  
        writeln (Tel);  
      end;  
    writeln;  
  end;  
end;  
close ( f ); readln; end.
```

## Задачи для самостоятельной работы:

- 1) Написать программу, которая при первом запуске создает файл F записей, а при повторных запусках эта программа дописывает в этот файл следующую информацию:  
фамилия, имя, год рождения, рост и вес.
  
- 2, а) Написать программу, которая считывает файл F созданный в предыдущей программе и выводит на экран данные по фамилиям, первые буквы которых А, Б или В.
  
- 2, б) Написать программу, которая сортирует записи в файле F по алфавиту по фамилиям.
  
- 2, в) Написать программу, которая в файле находит самого высокого человека и выводит его данные.