

Тип ЗАПИСЬ

Структурный тип

Запись (*англ. record*) - это сложный тип данных, позволяющие объединить данные разных типов. Запись можно назвать наиболее общим сложным типом данных. Название "запись" появилось из тех соображений, что данные разного типа можно встретить в таблицах: в каждой строке записаны сразу несколько разных значений. Таким образом, одна запись соответствует одной строке данных: она имеет несколько **полей**, каждое из которых хранит своё **значение**.

Описание записи

Чтобы работать в программе с записями, нужно сначала описать соответствующий тип данных. Запись описывается следующим образом:

```
type
```

```
    имя_типа_записи = record
```

```
        {поля записи}
```

```
end;
```

В блоке полей описание идёт точно так же, как описание переменных: указывается имя поля, далее через двоеточие тип данных этого поля. Сами поля разделяются точкой с запятой.

Для примера:

```
type TForeignLanguages = (flEnglish, flGerman,  
flFrench);
```

```
type
```

```
TPerson = record
```

```
  Name: String;
```

```
  Age: Byte;
```

```
  Education, PC: Boolean;
```

```
  Foreign: set of TForeignLanguages;
```

```
end;
```

Примеры других записей

TPoint - запись с двумя полями - *X* и *Y* - служит для хранения координат точки.

```
TPoint = record
```

```
  X: Longint;
```

```
  Y: Longint;
```

```
end;
```

TRect - позволяет хранить данные о прямоугольной области. Свойства *Left*, *Top*, *Right* и *Bottom* отвечают за координаты сторон прямоугольника, а координаты *TopLeft* и *BottomRight* типа *TPoint* - указывают на левую верхнюю и правую нижнюю вершины соответственно (эти вершины позволяют однозначно определить прямоугольник на плоскости, стороны которого параллельны осям координат). Однако в один момент времени используется только один набор из этих свойств - либо 4 координаты, либо 2 точки.

Хранение даты.

```
TDate = record
```

```
  Day: 1..31;
```

```
  Month: 1..12;
```

```
  Year: 1900..2100;
```

```
End;
```

Запись с константами

```
type TMyRecord = record
```

```
myRecord1 : string;
```

```
myRecord2 : integer;
```

```
myRecord3 : Currency;
```

```
end;
```

```
const MyRecordConst : TMyRecord =
```

```
( myRecord1 : 'myString';           myRecord2 : 125;
```

```
  myRecord3 : 100000 );
```

Работа с записями

имя_переменной.название_поля

```
var Person: TPerson;
```

Пример

Вводим анкеты - интерфейс

- ФИО - TEdit (*Edit1*)
- Возраст - TEdit (*Edit2*)
- Высшее образование - TCheckBox (*CheckBox1*)
- Владение компьютером - TCheckBox (*CheckBox2*)
- Владение иностранными языками - TCheckListBox ером
- TCheckBox (*CheckBox2*)

Компонент **TCheckListBox** расположен на странице *Additional*.

На странице **Win32** есть компонент **TUpDown** - это две кнопки ("вверх-вниз" или "влево-вправо").

Если в свойстве **Associate** выбрать поле ввода *Edit2*, то кнопки автоматически разместятся рядом с полем и нажатие на них будет автоматически изменять число в поле.

The screenshot shows a window titled "Анкеты" with a standard Windows interface. It contains a form with several input fields: a text box for "ФИО:", a spin box for "Возраст:" showing the value 25, two checkboxes for "Высшее образование" and "Владение компьютером", and a check list box for "Владение языками:" with three items: "Английский", "Немецкий", and "Французский". A "Добавить" button is located at the bottom of the form.

```
procedure TForm1.Button1Click(Sender: TObject);
var Person: TPerson;
begin
  Person.Name:=Edit1.Text;
  Person.Age:=UpDown1.Position;
  Person.Education:=CheckBox1.Checked;
  Person.PC:=CheckBox2.Checked;
  Person.Foreign:=[];
  if CheckListBox1.Checked[0] then
    Person.Foreign:=Person.Foreign + [flEnglish];
  if CheckListBox1.Checked[1] then
    Person.Foreign:=Person.Foreign + [flGerman];
  if CheckListBox1.Checked[2] then
    Person.Foreign:=Person.Foreign + [flFrench];
end;
```

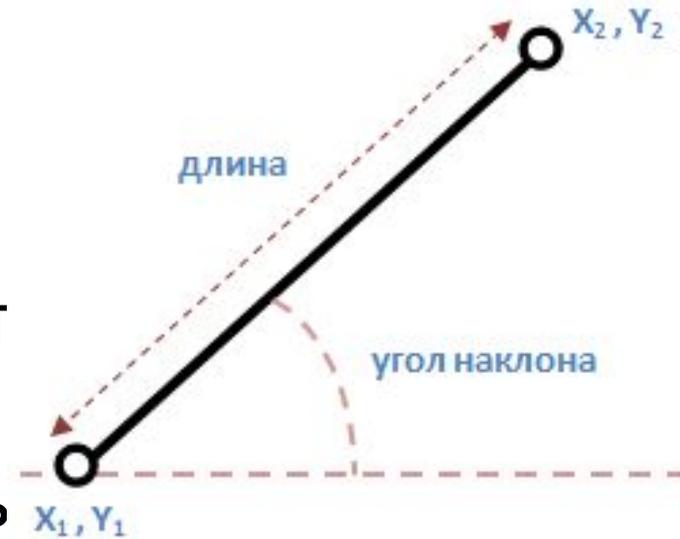
Оптимизация

```
procedure TForm1.Button1Click(Sender: TObject);
var Person: TPerson;
begin
  with Person do
  begin
    Name:=Edit1.Text;
    Age:=UpDown1.Position;
    Education:=CheckBox1.Checked;
    PC:=CheckBox2.Checked;
    Foreign:=[];
    if CheckListBox1.Checked[0] then
      Foreign:=Foreign + [flEnglish];
    if CheckListBox1.Checked[1] then
      Foreign:=Foreign + [flGerman];
    if CheckListBox1.Checked[2] then
      Foreign:=Foreign + [flFrench];
  end;
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
var Person: TPerson; I: Byte;
begin
  with Person do
  begin
    Name:=Edit1.Text;
    Age:=UpDown1.Position;
    Education:=CheckBox1.Checked;
    PC:=CheckBox2.Checked;
    Foreign:=[];
    for I := 0 to CheckListBox1.Items.Count-1 do
      if CheckListBox1.Checked[I] then
        Foreign:=Foreign + [TForeignLanguages(I)];
  end;
end;
```

Записи с вариантами

Записи с вариантами - такой тип записей, у которых создано несколько наборов полей, а используемый набор определяет специальным полем-селектором. При этом часть полей могут быть общими для всех наборов.



Например, информация об отрезке прямой на плоскости может быть представлена двумя способами:

- Двумя точками, каждая из которых имеет координаты X и Y (т.е. X_1, Y_1, X_2, Y_2).
- Одной точкой (X, Y) , длиной отрезка и углом между ним и какой-либо осью (напр-р, осью X).

- type TLineSegment = record
 X1,Y1: Real; X2,Y2: Real; end;
- type TLineSegment = record
 X,Y: Real; //Один из концов отрезка
 Angle: Real; //Угол наклона
 Length: Real; //Длина отрезка end;
- TLineSegment = record
case LType:
 TLineSegmentType of IsPoints: //Здесь
нужно описать первый набор полей...
 IsPolar: //...а здесь второй end;
end;

```
TLineSegment = record
case LType: TLineSegmentType of
  IsPoints: ( X1,Y1: Real;
              X2,Y2: Real;);
  IsPolar: ( X,Y: Real;
            Angle, Length: Real;);
end;
```

```
type TLineSegmentType =  
(IsPoints,IsPolar);  
TLineSegment = record  
X,Y: Real; //Один из концов отрезка  
case LType: TLineSegmentType of  
  IsPoints: (  
    X2,Y2: Real; ); //Второй конец отрезка  
  IsPolar: (  
    Angle: Real; //Угол наклона  
    Length: Real; );//Длина  
end;
```



Form1



Ввод отрезка

Координаты:

Декартовы

Полярные

X1:

Y1:

X:

Y:

X2:

Y2:

Угол:

Длина:

Сохранить

```
procedure TForm1.SaveButtonClick(Sender: TObject);
var L: TLineSegment;
begin
  if PointsRadio.Checked then  L.LType:=IsPoints
    else  L.LType:=IsPolar;
  case L.LType  of  IsPoints:
  begin
    L.X:=StrToFloat(X1Edit.Text);
    L.Y:=StrToFloat(Y1Edit.Text);
    L.X2:=StrToFloat(X2Edit.Text);
    L.Y2:=StrToFloat(Y2Edit.Text);
  end;
  IsPolar:
  begin
    L.X:=StrToFloat(XEdit.Text);
    L.Y:=StrToFloat(YEdit.Text);
    L.Angle:=StrToFloat(AngleEdit.Text);
    L.Length:=StrToFloat(LengthEdit.Text);
  end;
end;
end;end;
```