




Записи (RECORD)



Запис - це структурований тип даних, що складається з фіксованого числа компонентів, які називають полями. В одному полі дані мають той самий тип, а в різних полях можуть мати різні типи.

Загальний вигляд опису типу **record**:

type *<назва типу запису>* = **record**

<назва поля 1> : *<тип поля 1>*;

<назва поля 2> : *<тип поля 2>*;

.....

<назва поля N> : *<тип поля N>*;

end;



Доступ до конкретного поля запису дає
складене ім'я вигляду

<назва запису> . <назва поля>

Приклад 1. Дати яких-небудь подій
можна описати в такий спосіб:

```
type date=record
```

```
  month: 1..12;
```

```
  day: 1..31;
```

```
  year: integer;
```

```
end;
```

```
var D: date;
```

Поле **day** змінної **D** записується як **D.day**.

Наприклад, щоб заслати в **D** дату 12.01.1985, треба виконати наступні оператори:


D.day:=12;

D.month:=1;

D.year:=1985;

Приклад 2. Тип який описує оцінки з трьох предметів (геодезії, математики, інформатики і рейтинг студента, як середнє з цих трьох оцінок).

```
type reiting=record  
  geod,math,inform:integer;  
  reit:real;  
  end;  
var C:reiting;
```



Запис може бути вкладеним в інший запис.

Приклад 3. Запис про анкетні дані студентів: прізвище, ім'я, дату народження та середній бал, можна описати наступним чином:



```
grupa=record  
name, surname:string[20];  
birthday:record  
year:integer;  
month:1..12;  
day:1..31;  
end;  
sbal:real;  
end;
```

Запис може бути компонентом інших структур.


Приклад 4. Дано оцінки студентів по геодезії, математиці й інформатиці в групі (наприклад із трьох чоловік)

$X[1].geod=4$, $X[1].math=3$, $X[1].inform=5$,

$X[2].geod=3$, $X[2].math=4$, $X[2].inform=3$,

$X[3].geod=5$, $X[3].math=4$, $X[3].inform=4$.

Обчислити рейтинг кожного студента, як середній бал з цих трьох оцінок.




Відповідний фрагмент програми виглядає так:

```
Var X: array [1..3] of reiting; {див. приклад 2}  
i:word;  
begin  
  X[1].geod=4; X[1].math=3; X[1].inform=5;  
  X[2].geod=3; X[2].math=4; X[2].inform=3;  
  X[3].geod=5; X[3].math=4; X[3].inform=4;  
  For i:=1 to 3 do  
    X[i].reit:=(X[i].geod+X[i].math+X[i].inform)/3;  
  .....  
end;
```




Оператор приєднання WITH



Оператор **With** використовується для зручності роботи зі змінними типу **record** (запис). Він дає змогу звертатися до поля запису вказуючи лише ім'я поля.

Загальний вигляд оператора **with**:

with <ім'я змінної типу record> do <оператор>;



В операторі що слідує після **do** при посиланнях на компоненти запису ім'я запису можна опускати.

Зауваження. В операторі **with** після **do** виконується лише один оператор. Якщо потрібно щоб виконувались декілька операторів, то їх необхідно взяти в операторні дужки (**begin ... end**).

Приклад 5. Для занесення оцінок першого студента в попередньому прикладі достатньо виконати оператори:

```
with X[1] do
```


```
begin
```

```
geod:=4;
```


```
math:=3;
```

```
inform:=5;
```

```
end;
```



Тепер використовуючи оператор приєднання **with** і оператор циклу з параметром програму для вище наведеного прикладу 4 можна записати в такому вигляді:



```
Program Zapysy;  
Const n=3;  
type reiting=record  
  geod,math,inform:integer;  
  reit:real;  
  end;  
Var X: array [1..n] of reiting;  
begin
```

```
for i:=1 to n do
  with X[i] do
    begin
      writeln('Enter ocinku z geodezii',i,
        'studenta');
      readln(geod);
      writeln('Enter ocinku z matematuki',i,
        'studenta');
      readln(math);
      writeln('Enter ocinku z informatuki ',i,
        'studenta');
      readln(inform);
    end;
```



```
For i:=1 to n do
```

```
  With X[i] do
```

```
    begin
```

```
      reit:=(geod+math+inform)/3;
```

```
      writeln('Reiting',i,'studenta=',X[i]:5:2);
```

```
    end;
```

```
end.
```

Змінюючи константу **n** можна обчислювати рейтинги для різної кількості студентів.

Приклад 6. Дано $z_1 = 2 + 3 * i$, $z_2 = 4 - i$,

обчислити $z_3 = \frac{z_1 + \bar{z}_2}{z_2}$. Нехай в загальному випадку

$$z_1 = a_1 + b_1 * i, \quad z_2 = a_2 + b_2 * i, \quad z_3 = a_3 + b_3 * i.$$

Тоді сума $z_3 = z_1 + z_2 = a_1 + a_2 + (b_1 + b_2) * i$.

Комплексно спряжене $\overline{a + b * i} = a - b * i$.

Ділення комплексних чисел

$$z_3 = \frac{z_1}{z_2} = \frac{a_1 a_2 + b_1 b_2}{a_2^2 + b_2^2} + \frac{a_2 b_1 - a_1 b_2}{a_2^2 + b_2^2} i.$$

Напишемо програму для виконання дій над комплексними числами із застосуванням процедур. Для опису типу комплексного числа використаємо запис.

```
Program Complexni_Chysla;  
Type complex=record  
  re,im:real;  
  end;  
Var z1,z2,z3,r1,r2:complex;  
procedure spr(c:complex; Var r:complex);  
  begin  
    r.re:=c.re;  
    r.im:=-c.im;  
  end;  
procedure suma(c1,c2:complex; Var r:complex);  
  begin  
    r.re:=c1.re+c2.re;  
    r.im:=c1.im+c2.im;  
  end;
```

```
procedure dil(c1,c2:complex; Var
  r:complex);
Var d:real;
begin
  d:=sqr(c2.re)+sqr(c2.im);
  if d<>0 then
    begin
      r.re:=(c1.re*c2.re+c1.im*c2.im)/d;
      r.im:=(c1.im*c2.re-c1.re*c2.im)/d;
    end;
end;
```

```
{-----Osnovna programa-----}  
begin  
writeln('Enter z1.re,z1.im');  
readln(z1.re,z1.im);  
writeln('Enter z2.re,z2.im');  
readln(z2.re,z2.im);  
spr(z2,r1);  
suma(z1,r1,r2);  
dil(r2,z2,z3);  
writeln('z3.re=',z3.re,' z3.im=',z3.im);  
end.
```