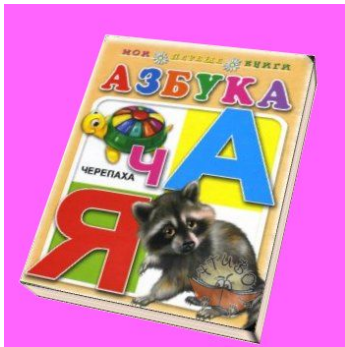


# Тема 7.

# Записи

# Записи



## Свойства:

- автор (*строка*)
- название (*строка*)
- год издания (*целое число*)
- количество страниц (*целое число*)

**Задача:** объединить эти данные в единое целое

**Запись** – это тип данных, который может включать в себя несколько *полей* – элементов разных типов (в том числе и другие структуры).

## Размещение в памяти

автор	название	год издания	количество страниц
40 символов	80 символов	целое	целое

# Одна запись

## Объявление (выделение памяти):

название

запись

поля

```
var Book: record
  author: string[40]; // автор, строка
  title:  string[80]; // название, строка
  year:   integer;   // год издания, целое
  pages:  integer;   // кол-во страниц, целое
end;
```

## Обращение к полям:

```
readln(Book.author); // ввод
readln(Book.title);
Book.year := 1998; // присваивание
if Book.pages > 200 then // сравнение
  writeln(Book.author, '.', Book.title); // вывод
```



Для обращения к полю записи используется точка!

# Обращение к записи

---

Обращение к записи в целом допускается только в операторах присваивания, где слева и справа от знака присваивания используются имена записей одинакового типа.

Во всех остальных случаях оперируют отдельными полями записей.

Чтобы обратиться к отдельной компоненте записи, необходимо задать имя записи и через точку указать имя нужного поля, например:

***str.fio, str.tel***

Такое ***ИМЯ*** называется ***СОСТАВНЫМ***.

# Записи

---

Запись может входить в состав данных более сложной структуры.

Можно говорить, например, о массивах и файлах, состоящих из записей.

Запись может быть полем другой записи.

# Оператор присоединения

---

Обращение к компонентам записей можно упростить, если воспользоваться оператором присоединения

*With.*

Он позволяет заменить составные имена, характеризующие каждое поле, просто на имена полей, а имя записи определить в операторе присоединения.

## Формат оператора:

```
With    <переменная-запись>  
    {, <переменная-запись>}  do  
        <оператор> ;
```

# Оператор присоединения

---

Обращение к компонентам записей можно упростить, если воспользоваться оператором присоединения *With*.

Он позволяет заменить составные имена, характеризующие каждое поле, просто на имена полей, а имя записи определить в операторе присоединения.

## Формат оператора:

```
With    <переменная-запись>  
        {, <переменная-запись>}  do  
                <оператор> ;
```

Заголовок этого оператора открывает область действия "внутренних" имен полей записи, которые могут быть использованы как имена переменных.

# Оператор присоединения

---

Оператор вида

*With r1, ..., rn do <оператор>*

эквивалентен оператору

*With r1 do*

*with r2 ...*

*with rn do*

*<оператор> .*



# Оператор присоединения: пример

---

```
Type Student = Record
    F1, F2, F3 : Name;
    Day : 1..31;
    Month : 1..12;
    Year : integer;
    StudDoc : integer
end;
Var A : Student;
```

Значение переменной **A** можно изменить операторами:

```
A.F1 := 'Иванов ';
A.F2 := 'Илья ';
A.F3 := 'Иннокентьевич ';
A.Day := 14;
A.Month := 9;
A.Year := 1976;
A.StudDoc := 123;
```

# Оператор присоединения: пример

---

Предпочтительнее использовать следующий вариант:

```
with   A   do   begin  
    F1 := 'Иванов ' ;  
    F2 := 'Илья ' ;  
    F3 := 'Иннокентьевич ' ;  
    Day := 14 ;  
    Month := 9 ;  
    Year := 1976 ;  
    StudDoc := 123 ;  
end ;           { оператора with }
```



# Массив записей

---

## Обращение к полям:

```
for i:=1 to N do begin
  readln(aBooks[i].author);
  readln(aBooks[i].title);
  ...
end;
for i:=1 to N do
  if aBooks[i].pages > 200 then
    writeln(aBooks[i].author, '.',
            aBooks[i].title);
```



`aBooks[i].author` – обращение к полю `author` записи `aBooks[i]`

# Новый тип данных – запись

## Объявление типа:



Память не выделяется!

```
type TBook = record
  author: string[40]; // автор, строка
  title:  string[80]; // название, строка
  year:  integer; // год издания, целое
  pages : integer; // кол-во страниц, целое
end;
```

TBook – *Type Book* («тип книга») – удобно!

## Объявление переменных и массивов:

```
const N = 10;
var Book: TBook; // одна запись
    aBooks: array[1..N] of TBook; // массив
```

# Записи в процедурах и функциях

---

## Процедура:

```
procedure ShowAuthor ( b: TBook );  
begin  
    writeln ( b.author );  
end;
```

## Функция:

```
function IsOld( b: TBook ): boolean;  
begin  
    IsOld := b.year < 1900;  
end;
```

## Основная программа:

```
Book.author := 'А.С. Пушкин';  
ShowAuthor ( Book );  
Book.year := 1800;  
writeln( IsOld(Book) );
```

# Файлы записей

---

## Объявление указателя на файл:

```
var F: file of TBook;
```

## Запись в файл:

```
Assign(F, 'books.dat'); { связать с указателем }  
Rewrite(F);           { открыть файл для запись }  
writeln(F, Book);    { запись }  
for i:=1 to 5 do  
    writeln(aBook[i]); { запись }  
Close(F);            { закрыть файл }
```

# Чтение из файла

## Известное число записей:

```
Assign(F, 'books.dat'); { связать с указателем }
Reset(F);                { открыть для чтения }
Read(F, Book);           { чтение }
for i:=1 to 5 do
  Read(F, aBook[i]); { чтение }
Close(F);                { закрыть файл }
```

## «Пока не кончатся»:

```
count := 0;
while not eof(F) do begin
  count := count + 1;    { счетчик }
  Read(F, aBook[count]); { чтение }
end;
```

пока не дошли до конца файла F  
**EOF** = *end of file*



# Пример программы

**Задача:** в файле `books.dat` записаны данные о книгах в виде массива структур типа `TBook` (не более 100). Установить для всех 2008 год издания и записать обратно в тот же файл.

```
type TBook ... ;
```

полное описание  
структуры

```
const MAX = 100;
```

```
var aBooks: array[1..MAX] of TBook;
```

```
    i, N: integer;
```

```
    F: file of TBook;
```

```
begin
```

```
    { прочитать записи из файла, N - количество }
```

```
    for i:=1 to N do
```

```
        aBooks[i].year := 2008;
```

```
    { сохранить в файле }
```

```
end.
```

# Пример программы

## Чтение «пока не кончатся»:

```
Assign(f, 'books.dat');  
Reset(f);  
N := 0;  
while not eof(F) and (N < MAX) do begin  
    N := N + 1;  
    read(F, aBooks[N]);  
end;  
Close(f);
```

чтобы не выйти за пределы массива

## Сохранение:

```
Assign(f, 'books.dat'); { можно без этого }  
Rewrite(f);  
for i:=1 to N do write(F, aBooks[i]);  
Close(f);
```

# Выделение памяти под запись

```
var pB: ^TBook;
```

переменная-  
указатель на TBook

```
begin
```

```
New (pB) ;
```

выделить память под запись,  
записать адрес в pB

```
pB^.author := 'А.С. Пушкин' ;
```

```
pB^.title := 'Полтава' ;
```

```
pB^.year := 1990 ;
```

```
pB^.pages := 129 ;
```

```
Dispose (pB) ;
```



Для обращения  
к полю записи по  
адресу  
используется  
знак ^

```
end.
```

освободить  
память

# Сортировка массива записей

---

**Ключ (ключевое поле)** – это поле записи (или комбинация полей), по которому выполняется сортировка.

```
const N = 100;  
var aBooks: array[1..N] of TBook;  
    i, j, N: integer;  
    temp: TBook; { для обмена }  
begin  
    { заполнить массив aBooks }  
    { отсортировать = переставить }  
    for i:=1 to N do  
        writeln(aBooks[i].title,  
                aBooks[i].year:5);  
end.
```

# Сортировка массива записей

```
for i:=1 to N-1 do
  for j:=N-1 downto i do
    if aBooks[j].year > aBooks[j+1].year
    then begin
      temp := aBooks[j];
      aBooks[j] := aBooks[j+1];
      aBooks[j+1] := temp;
    end;
```

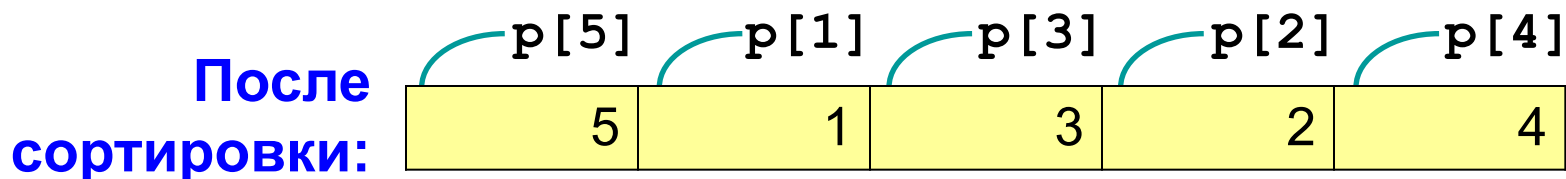
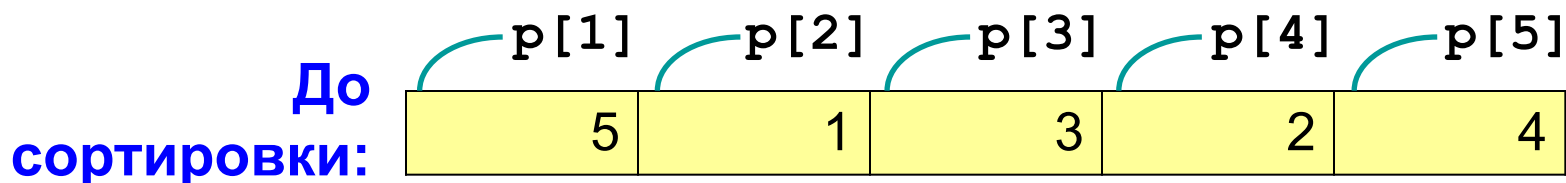
# Сортировка массива записей

## Проблема:

как избежать копирования записи при сортировке?

## Решение:

использовать вспомогательный массив указателей, при сортировке переставлять указатели.



## Вывод результата:

```
for i:=1 to N do
  writeln(p[i]^ .title, p[i]^ .year:5);
```

# Реализация в программе

```
type PBook = ^TBook; { новый тип данных }
```

```
var p: array[1..N] of PBook;
```

```
begin
```

```
  { заполнение массива записей}
```

```
  for i:=1 to N do
```

```
    p[i] := @aBooks[i];
```

вспомогательные  
указатели

начальная  
расстановка

```
  for i:=1 to N-1 do
```

```
    for j:=N-1 downto i do
```

```
      if p[j]^year > p[j+1]^year then begin
```

```
        temp := p[j];
```

```
        p[j] := p[j+1];
```

```
        p[j+1] := temp;
```

```
      end;
```

меняем только  
указатели, записи  
остаются на местах

```
  for i:=1 to N do
```

```
    writeln(p[i]^title, p[i]^year:5);
```

```
end.
```