

Комбинированный тип данных (Record)

Решение задач

Комбинированный тип данных (Record)

При работе с массивами основное ограничение заключается в том, что каждый элемент должен иметь один и тот же тип. Однако при решении многих задач возникает необходимость хранить и обрабатывать совокупности данных различного типа как единое целое. Для этого применяется запись.

Запись – это составной тип данных, содержащий набор элементов разных типов. Составляющие запись элементы называются ее полями. В записи каждое поле имеет собственное имя. Чтобы описать запись, необходимо указать имя записи и имена объектов, составляющих запись и их типы.

Общий вид такой:

```
Типе <имя записи> =  
    <поле 1>:<тип 1>;  
    <поле 2>:<тип 2>;  
    ...  
    <поле n>:<тип n>  
End;
```

Примеры объявления типа данных Record

Пример 1.

Объявление записи в разделе Var

Запись в записной книжке

Var

address: record

**Famil:string[25];
Name: string[15];
City: string[20];
Street: string[40];
Home:integer;
Telephone:integer
End;**

Пример 2.

Объявление записи в разделе Type

Запись в записной книжке

Type address=record

**Famil:string[25];
Name: string[15];
City: string[20];
Street: string[40];
Home:integer;
Telephone:integer
End;**

Var

friend: array[1..15] of address:

Примеры объявления типа данных Record

Пример 3.

```
Type year=1583..3000;  
Month=1..12;  
Day=1..31;  
date= record  
    dyear: year;  
    dmonth: Month;  
    dday: Day;  
End;  
Var mas: array[1..15] of  
date;
```

Пример 4.

Type

```
date= record
```

```
    year: 1900..2100;
```

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

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

```
    End;
```

```
pupil= record
```

```
    Famil, Name, Otch: string[25];
```

```
    Birthday: date;
```

```
    Class: 1..11;
```

```
    End;
```

```
Var spisok: array[1..1000] of  
pupil;
```

Оператор With

Обращения к полям записи могут быть громоздки, для упрощения работы с записями предназначен оператор **With**:

With <Имя записи>do<оператор>;

Один раз указав переменную типа запись в операторе **With**, можно работать с именами полей как с обычными переменными, т.е. без указания переменной типа запись

```
For i:=1 to 15 do
Begin
WriteLn('Введите фамилию');
ReadLn(friend[i]. famil);
WriteLn('Введите имя');
ReadLn(friend[i]. Name);
....
Clrscr
End;
```

```
For i:=1 to 15 do
With friend[i] do
Begin
WriteLn('Введите фамилию');
ReadLn(famil);
WriteLn('Введите имя');
ReadLn(Name);
....
Clrscr
End;
```

Решение задач

Задача1: В классе 30 учеников. Введите фамилии и оценки по 8 предметам для каждого ученика, найдите средний балл и выведите на печать фамилию и средний балл лучшего ученика.

```
Uses crt;
Type sved = record
    Famil:string[25];
    P1,p2,p3,p4,p5,p6,p7,p8:1..5;
    Ocsr:real
End;

Var
Spisok:array[1..30] of sved;
Z:sved; l,n:integer;    Champ:real;
Begin
For i:=1 to 30 do
    With spisok[i] do Begin
        Writeln('Введите фамилию ученика');
        Readln(famil);
        Writeln('Введите оценки по 8 предметам');
        Readln(P1,p2,p3,p4,p5,p6,p7,p8);
        Clrscr
    End;
```

**Описание и ввод
данных: фамилии
и оценок по 8
предметам**

продолжение

Решение задач

Задача 1: В классе 30 учеников. Введите фамилии и оценки по 8 предметам для каждого ученика, найдите средний балл и выведите на печать фамилию и средний балл лучшего ученика.

```
For i:=1 to 30 do
  With spisok[i] do
    Begin
      Ocsr:=( P1+p2+p3+p4+p5+p6+p7+p8)/8;
    End;
  Champ:=0;
  For i:=1 to 30 do
    If spisok[i]. Ocsr>= Champ then Champ:= spisok[i]. Ocsr;
  For i:=1 to 30 do
  If spisok[i]. Ocsr= Champ then
  With spisok[i] do
  Writeln(Famil:30,'---',ocsr:8:2);
  Readln;
  End.
```

**Подсчет среднего балла
каждого ученика, поиск
максимального (Champ)
и вывод на печать**

Задача 2. 10 абитуриентов поступают на факультет информатики. Введите фамилии и оценки по 3 экзаменам (математика, информатика, физика) для каждого абитуриента, найдите средний балл и выведите на печать фамилии абитуриентов, имеющих средний балл, больший или равный четырем.

Решение задач

Задача 3 В некотором вузе абитуриенты проходят предварительное тестирование, по результатам которого могут быть допущены к сдаче вступительных экзаменов в первом потоке. Тестирование проводится по трем предметам, по каждому предмету абитуриент может набрать от 0 до 100 баллов. При этом к сдаче экзаменов в первом потоке допускаются абитуриенты, набравшие по результатам тестирования не менее 30 баллов по каждому из трех предметов. На вход программы подаются сведения о результатах предварительного тестирования. Известно, что общее количество участников тестирования не превосходит 300.

В первой строке вводится количество абитуриентов, принимавших участие в тестировании, **N**. Далее следуют N строк, имеющих следующий формат:

<Фамилия> <Имя> <Баллы>,

Здесь **<Фамилия>** - строка, состоящая не более чем из 20 символов;

<Имя> - строка, состоящая не более чем из 15 символов;

<Баллы> - строка, содержащая три целых числа, разделенных пробелом, соответствующих баллам, полученным при тестировании по каждому из трех предметов.

продолжение

Решение задач

Задача 3 (продолжение условия)

При этом **<Фамилия>** **<Имя>** , **<Имя>** и **<Баллы>** разделены одним пробелом.

Примеры входных строк:

Петренко Наталья 58 66 38

Антипов Сергей 99 50 72

Напишите программу, которая будет выводить на экран фамилии и имена абитуриентов, допущенных к сдаче экзаменов в первом потоке. При этом фамилии и имена абитуриентов можно выводить в произвольном порядке.

```
Var a: array[1..300] of record
    name: string;
    sum: integer;
end;
C:char;  i,j,N,s,m:integer;
```

**Описание
переменных**

РЕШЕНИЕ

```
begin
Readln(N); {считано количество абитуриентов}
for i:=1 to N do
  begin
    a[i].name:=“”;
    repeat
      read(c);
      a[i].name:= a[i].name+c;
    until c=‘ ‘; {считана фамилия}
    Repeat
      read(c);
      a[i].name:= a[i].name+c;
    until c=‘ ‘; {считано имя}
    a[i].sum:=0;
    for j:=1 to 3 do
      begin
        read(m);
        if m>= 30 then a[i].sum:= a[i].sum+1;
      end;
    end;
  end;
for i:=1 to N do
  if a[i].sum=3 then writeln(a[i].name);
readln;
end.
```

**абитуриенты,
набравшие по
результатам
тестирования не
менее 30 баллов
по каждому из
трех предметов
выводятся на
печать.**

Задачи повышенной сложности

На вход программы подаются сведения о сдаче экзаменов учениками 9-х классов некоторой средней школы. В первой строке сообщается количество учеников N , которое не меньше 10, но не превосходит 100, каждая из следующих N , строк имеет следующий формат:

<Фамилия><Имя><оценки>,

где <Фамилия> - строка, состоящая не более чем из 20 символов,
<Имя> - строка, состоящая не более из 15 символов,
<оценки> - три целых числа, соответствующие оценкам по пятибалльной системе.

<Фамилия> и <Имя>, а также <Имя> и <оценки> разделены одним пробелом.

Пример входной строки:

Иванов Петр 4 2 4

Требуется написать программу, которая будет выводить на экран фамилии и имена неуспевающих учеников (имеющихся по результатам экзаменов хотя бы одну двойку), располагая их в порядке уменьшения числа двоек.

Решение

Uses crt;

```
var p:array[1..100] of record
    name:string;
    sum: integer;
end;
N,i,j,m:integer;
c:char;
begin
    readln(N);
    for i:=1 to N do
    begin
        p[i].name:='';
        repeat
            read(c);
            p[i].name:= p[i].name+c;
        until c=' '; {считана фамилия}
        Repeat
            read(c);
            p[i].name:= p[i].name+c;
        until c=' '; {считано имя}
```

```
        p[i].sum:=0;
        for i:=1 to 3 do
        begin
            read(m);
            if m= 2 then
                p[i].sum:=p[i].sum+1;
            end; {подсчитано
                количество двоек}
        readln;
        end;
    For i:=1 to N do
    If p[i].sum=3 then writeln(p[i].name);
    { вывод учащихся с 3 двойками}
    For i:=1 to N do
    If p[i].sum=2 then writeln(p[i].name);
    { вывод учащихся с 2 двойками}
    For i:=1 to N do
    If p[i].sum=1 then writeln(p[i].name);
    { вывод учащихся с 1 двойкой}
    end.
```

Задачи повышенной сложности

На вход программы подаются сведения о багаже пассажиров некоторого поезда. В первой строке сообщается количество пассажиров N , которое не может быть отрицательным числом, каждая из следующих N , строк имеет следующий формат: **<Фамилия><Имя><количество вещей>**. Здесь **<Фамилия>** - строка, состоящая не более чем из 20 символов, **<Имя>** - строка, состоящая не более из 15 символов, **<количество вещей>** - целое число, в диапазоне от 0 до 9. **<Фамилия>** и **<Имя>**, а также **<Имя>** и **<количество вещей>** разделены одним пробелом.

Пример входной строки:

Емельянова Екатерина 4

Требуется написать программу, которая будет выводить на экран фамилии и имена пассажиров, имеющих более 2 вещей. Строки данных выводить на экран упорядоченными по фамилии.

Пример вывода:

Емельянова Екатерина 4

Сергеев Григорий 3

Якушев Антон 4

Uses crt;

Type person=record

fam:string[20];

name:string[15];

col:1..9

end;

var mas:array[1..50] of person;

n,k,i,j,w:integer; v:real;

z:person;

s:string;

begin

write(' n=');readln(n);

For i:=1 to n do

begin

writeln('family:'); readln(s);

k:=pos(' ',s);

mas[i].fam:=copy(s,1,k-1);

delete(s,1,k);

k:=pos(' ',s);

mas[i].name:=copy(s,1,k-1);

delete(s,1,k);

val(s,v,w);

mas[i].col:=trunc(v); clrscr;

end;

Решение

for j:=1 to n-1 do

for i:=1 to n-j do

if copy(mas[i].fam,1,1)>copy(mas[i+1].fam,1,1)

then begin

z.fam:=mas[i].fam; z.name:=mas[i].name;

z.col:=mas[i].col;

mas[i].fam:=mas[i+1].fam;

mas[i].name:=mas[i+1].name;

mas[i].col:=mas[i+1].col;

mas[i+1].fam:=z.fam;

mas[i+1].name:=z.name; mas[i+1].col:=z.col;

end;

For i:=1 to n do

with mas[i] do

begin

if col>2 then begin write(fam,' ');

write(name,' ');

writeln(col);

end;

end;

end.