

ЛАБОРАТОРНАЯ РАБОТА

Тема: "Комбинированный тип данных - записи"

Цель работы

1. Получить навыки в организации ввода и вывода значений комбинированных типов данных.
2. Получить навыки программирования задач с использованием записей.

Краткие сведения из теории

10.1. Объявление данных типа записи

Комбинированный тип характеризует объекты, называемые записями. Синонимом понятия "комбинированный тип" является понятие "структурный тип". Запись (структура) - это сложная переменная с несколькими компонентами. При определении комбинированного типа задаются имя всей записи, имя и тип каждой компоненты. Описание комбинированного типа начинается со служебного слова RECORD и заканчивается словом END. Записи, как и другие данные, объявляются в разделе описаний и используются в разделе операторов.

Записи можно объявлять в разделе TYPE либо VAR.

Объявление записи в разделе VAR имеет следующий вид:

```
VAR
    < имя записи > : RECORD
        < имя компоненты 1: тип >;
        < имя компоненты 2: тип >;
        . . .
        < имя компоненты N: тип >
    END;
```

Здесь служебное слово RECORD (запись) выполняет роль открывающей операторной скобки, END - закрывающей. Внутри операторных скобок описываются компоненты записи. Допускается вместо имени записи указывать список имен, т.е. имена записей, разделенные запятыми. Компоненты записи вместе с их описанием называются полями записи.

Более универсальной формой объявления записи является описание с использованием раздела TYPE, которое имеет вид:

```
TYPE
    < имя типа > = RECORD
        < имя компоненты 1 >: тип;
        < имя компоненты 2 >: тип;
        . . .
        < имя компоненты N >: тип
    END;
```

```
VAR
    < имя записи >: имя типа;
```

Пример. Дана ведомость списка студентов с их оценками (рис.1) Для представленной ведомости объявление записи в разделе переменных выглядит следующим образом:

```
VAR
    vedom : RECORD
        n      : integer;
        fio    : string[15];
        progr,fizika : integer
    END;
```

№ п/п	Фамилия имя, отчество	Оценка	
		программирование	физика
1	Бадмаев И.П.	5	4
2	Иванов А.Р.	4	4
3	Павликова Ю.Т.	5	3

Рис.1. Ведомость успеваемости.

В данном примере фамилия имеет тип STRING, состоящий из 15 элементов, порядковый номер и оценки по предметам - тип INTEGER.

Длина записи Vedom равна 21 байту.

Объявление ведомости с использованием раздела типов имеет вид:

```

TYPE
  vedom = RECORD
    n      : integer;
    fio    : string[15];
    progr,fizika : integer
  END;
VAR
  v : vedom;

```

Здесь сначала введен тип с именем VEDOM, а затем указана переменная V, имеющая тип записи.

Поле записи используется в программе так же, как обычная переменная. Таким образом, поле записи можно указывать как в левой части оператора присваивания, так и в выражениях. Над полем записи можно выполнять действия, допустимые для данных его типа. Если тип поля записи - INTEGER, то выполняются все операции, допустимые для целых данных.

10.2. Доступ к полям записи

Доступ к полям записи производится с помощью селектора записи, имеющего следующий вид:

NAME_Z.NAME_P,

где NAME_Z - имя переменной комбинированного типа (всей записи);

NAME_P - имя поля.

В практическом программировании такая запись называется уточненным именем данного.

Для переменных, введенных выше, допустимы следующие конструкции:

Vedom.n := 5;

Vedom.fio := 'Иванов А.Р.';

или

V.n := 35;

V.fio := 'Павликова Ю.Т.';

Комбинированные типы можно использовать для построения более сложных структур: массивов; файлов; вложенных структур с одним или более полей, которые, в свою очередь, могут быть записью. Например:

```

VAR
  group  : array[1..10] of vedom;
  database : file of vedom;

```

Для переменных GROUP доступ к полям записей, составляющих этот массив, производится следующим образом:

```

. . .
Group[i].fio := 'Бадмаев И.П. ';
If group[i].fio = 'Бадмаев И.П. ' then
  WriteLn (group[i].progr)
Else writeln ('Нет такой фамилии!');

```

Рассмотрим случай, когда в составе записи содержатся поля, имеющие тип записи. Пусть для комбинированного типа VEDOM необходимо хранить информацию о дате сдачи экзамена. Эту информацию можно представить в виде трех полей: месяц, день, год, дополняющих предыдущий состав типа VEDOM. Однако, логичнее дату сдачи экзамена определить как отдельный тип. Это позволит использовать тип DATE в описании других типов и переменных:

TYPE

```
Date = RECORD
    Mounth : (jan,feb,mar,apr,may,jun,jul,aug,sep,oct,nov,dec);
    Day    : 1..31;
    Year   : 1900..2000
END;
```

Теперь тип DATE можно использовать в записи VEDOM:

TYPE

```
Vedom = RECORD
    N      : integer;
    Fio    : string[15];
    Progr,Fizika : integer;
    D_exem : date
END;
```

Доступ к полям D_EXEM осуществляется по общим правилам, т.е. при записи селектора слева от символа 'точка' всегда должна находиться переменная типа запись, а справа идентификатор поля этой записи, например:

```
V.D_exem.Mounth := jan;
V.D_exem.Day := 25;
```

Комбинированный тип может употребляться для спецификации параметров подпрограмм. Например, можно определить специальный тип для представления комплексных чисел как пары вещественных переменных (действительную и мнимую части комплексного числа):

TYPE

```
Complex = RECORD
    Re,Im : real
END;
```

Далее можно с помощью процедур определить операции над комплексными числами (сложение, умножение, деление):

```
Procedure Addc( c1,c2: complex; var R: complex);
Procedure Mulc( c1,c2: complex; var R: complex);
Procedure Divc( c1,c2: complex; var R: complex);
```

10.3. Записи с вариантами

Часто в зависимости от конкретного значения некоторого поля возникает необходимость в пределах одной записи иметь различную информацию. В таких случаях используются записи с вариантами.

Рассмотрим тип PERSON, содержащий информацию о человеке.

Если поле POL имеет значение M (мужской), то пусть необходимо предусмотреть такие поля:

- служил в армии или нет;
- если служил, то дату последних военных сборов.

Если поле POL имеет значение W (женский), то необходима информация о цвете глаз. Запись с вариантами типа PERSON имеет вид:

Type

```
Date = Record
    Mounth : (jan,feb,mar,apr,may,jun,jul,aug,
```

```

        sep,oct,nov,dec);
    Day : 1..31;
    Year : 1900..2000
    End;
Person = Record
    Fio : string[20];
    Special : word;
    Birthday : date;
    PersonPol: (M,W);
    Case pol : PersonPol of
        M: (Army : boolean;
            D_Army : date);
        W: (EyesColor : (blue,brown,
            gray,green))
    End;

```

Записи с вариантами имеют фиксированную и вариантную части. Изменяющаяся часть записи называется вариантом. Вариант всегда располагается в конце записи. Поле (в данном случае POL), позволяющее различать варианты, называется полем признака.

Вариантная часть содержит несколько альтернатив (в данном примере - M и W), в каждой из которых в круглых скобках задается список полей, присущих данному варианту (ARMY и D_ARMY -> M, EYESCOLOR -> W). Списку полей предшествует метка, являющаяся конкретным значением признака POL. Метка служит критерием выбора вариантов. Перечисление альтернатив начинается с определения признака POL.

Началом вариантной части является служебное слово CASE; после признака выбора вариантов записывается служебное слово OF. Вариантная часть завершается служебным словом END вместе с завершением всей записи. В определении комбинированного типа может быть только одна вариантная часть и она должна быть задана в конце записи.

Альтернативы вариантной части помечаются допустимыми значениями поля POL, которое определено в фиксированной части. Иногда поле, значения которого задают варианты, называют дискриминантом записи.

Идентификаторы полей во всех вариантах должны быть различными и отличаться от идентификаторов полей фиксированной части. В этом случае после метки, соответствующей этим значениям может стоять пустой список вида ().

Любой вариант, в свою очередь, может иметь свою вариантную часть, которая должна располагаться в конце списка полей данного варианта.

При использовании вариантных записей необходимо учитывать следующие особенности:

1. Для размещения переменной комбинированного типа всегда отводится фиксированный объем памяти, причем если в записи есть варианты, то объем определяется по самому большому варианту. Различные варианты одной записи как бы накладываются " друг на друга" в памяти, занимая одну и ту же область.
2. Система Турбо-Паскаль не содержит никаких средств контроля за правильностью работы с вариантами записей. За соответствием текущего значения дискриминанты и доступа к полям записи должен следить программист.

10.4. Оператор присоединения

Оператор присоединения предназначен для более наглядной и эффективной организации работы с данными комбинированного типа и используется для доступа к полям записи. Оператор присоединения начинается со служебного слова WITH, далее следует имя записи и служебное слово DO. Операторы, содержащие имена полей записи, заключаются в операторные скобки:

```

WITH < имя записи > DO
BEGIN

```

```

    < операторы, содержащие имена полей записи >

```

END;

Например: для рассмотренной записи (списка студентов) операции присваивания, суммирования и ввода можно объединить в один оператор:

```
With v do
Begin
    N := 4;
    SUM := progr + fizika;
    Read (N)
End.
```

Контрольные вопросы

1. Что понимается под записью в языке Паскаль?
2. Как объявляются записи?
3. Какие операции допустимы над полями записи?
4. Как организовать ввод и вывод данных типа записи?
5. Как осуществляется доступ к полям записи?
6. Можно ли использовать в записи поля одного типа?
7. Чем отличается запись от массива?
8. Каково назначение оператора присоединения?

Задание к работе

Выполнить индивидуальное задание

Содержание отчета

1. Титульный лист.
2. Словесная постановка задачи.
3. Графический или текстуальный алгоритм решения задачи.
4. Листинг программы.
5. Контрольный тест и результаты тестирования программы.
6. Инструкция по эксплуатации программы.
7. Ответы на контрольные вопросы.

Методические указания

1. При выполнении работы использовать массив записей.
2. Разработать алгоритмы и программы для решения задач заданий.
3. Скомпилировать программы.
4. Составить контрольные тесты и протестировать программы.
5. Составить отчет и представить его к защите.

Варианты индивидуальных заданий

1. Дан список учебной группы, включающий 20 человек. Для каждого студента известны: фамилия, имя, дата рождения, оценки по всем дисциплинам за последний семестр. Составить программу, которая обеспечивает ввод информации и отображение ее на экран в виде таблицы. Отобразить на экран анкетные данные студентов-отличников в виде таблицы. Отобразить на экран фамилию и имя студентов, родившихся зимой и весной.
2. Сведения об экзамене содержат следующие данные: дисциплину (программирование, алгебра, история, геометрия), дату сдачи экзамена (год, месяц, день), сведения о студенте (факультет, курс, группа, номер в журнале) и экзаменационную оценку. Задан набор сведений об экзаменах, сданных студентами за последние два года; в них факультет и предмет кодируются первыми буквами названия. Определить количество неуспевающих по программированию на экономическом факультете среди студентов первого курса, сдававших экзамены зимой 1995 года, вывести на экран их группу и номер в журнале.
3. Сведения об экзамене содержат следующие данные: дисциплину (программирование, социология, иностранный язык, физика), дату сдачи экзамена (год, месяц, день), сведения

- о студенте (фамилия, факультет, курс, группа) и экзаменационную оценку. Задан набор сведений об экзаменах, сданных студентами за последние несколько лет; в них факультет и предмет кодируются первыми буквами названия. Определить количество отличников по программированию на технологическом факультете среди студентов первого курса, сдававших экзамены летом 1995 года, вывести на экран их фамилии и группу.
4. Сведения об экзамене содержат следующие данные: дисциплину (программирование, вычислительная техника, информатика), дату сдачи экзамена (год, месяц, день), сведения о студенте (факультет, курс, группа, номер в журнале) и экзаменационную оценку. Задан набор сведений об экзаменах, сданных студентами за последние несколько лет; в них факультет и предмет кодируются первыми буквами названия. Определить, на каком факультете самый высокий средний балл по программированию среди студентов первого и второго курсов, сдававших экзамены зимой 1995 года.
 5. Сведения об экзамене содержат следующие данные: дисциплину (программирование, вычислительная техника, информатика), дату сдачи экзамена (год, месяц, день), сведения о студенте (факультет, курс, группа, номер в журнале) и экзаменационную оценку. Задан набор сведений об экзаменах, сданных студентами за последние несколько лет; в них факультет и предмет кодируются первыми буквами названия. Определить, на каком факультете самый высокий показатель качества успеваемости по информатике (то есть самый высокий процент отличников и хорошистов) среди студентов первого курса, сдававших экзамены зимой 1995 года или летом 1996 года.
 6. Справка о междугороднем телефонном разговоре содержит: номер телефона абонента (6 цифр), дату (год, месяц, день), время (час, минута), код города (3 цифры), номер телефона в другом городе (7 цифр), продолжительность разговора (в минутах), категорию (срочный, обычный) и тариф (плата в рублях за минуту). Определить дату такого телефонного разговора, которой является максимальным по продолжительности среди срочных разговоров за указанный месяц.
 7. Справка о междугороднем телефонном разговоре содержит: номер телефона абонента (6 цифр), дату (год, месяц, день), время (час, минута), код города (3 цифры), номер телефона в другом городе (7 цифр), продолжительность разговора (в минутах), категорию (срочный, обычный) и тариф (плата в рублях за минуту). Вывести на экран код города и номер телефона в другом городе для телефонных разговоров, состоявшихся с телефона 235678 8 марта 1996 года.
 8. Справка о междугороднем телефонном разговоре содержит: номер телефона абонента (6 цифр), дату (год, месяц, день), время (час, минута), код города (3 цифры), номер телефона в другом городе (7 цифр), продолжительность разговора (в минутах), категорию (срочный, обычный) и тариф (плата в рублях за минуту). Вывести на экран номер телефона абонента, код города и номер телефона в другом городе для срочных телефонных разговоров, состоявшихся между 15 марта и 12 апреля 1996 года.
 9. Деталь автомобиля описывается инвентарным номером (положительное целое число), весом (в килограммах), ценой и стоимостью (в рублях), датой начала производства (год, месяц, день), статусом (имеет или не имеет знак качества) и объемом производства (в штуках за смену). В заданной последовательности сведений о деталях найти инвентарные номера деталей с наибольшей датой начала производства среди всех заданных деталей. Вывести на экран инвентарный номер, объем производства, цену и стоимость деталей со знаком качества.
 10. Деталь автомобиля описывается инвентарным номером (положительное целое число), весом (в килограммах), ценой и стоимостью (в рублях), датой начала производства (год, месяц, день), статусом (имеет или не имеет знак качества) и объемом производства (в штуках за смену). В заданной последовательности сведений о деталях найти инвентарные номера деталей с минимальным весом среди деталей без знака качества. Вывести на экран инвентарный номер, объем производства, цену и стоимость деталей, выпускаемых с февраля 1977 года.