

Лабораторна робота № 7

Використання даних типу запис (record)

Мета роботи: оволодіти практичними навиками застосування комбінованих типів даних в TurboPascal (FreePascal); засвоїти прийоми створення елементарних баз даних.

Теоретичні відомості

Комбіновані типи, як і регулярні типи (масиви), є правильно сформовані складні типи. На відміну від масивів, записи дозволяють об'єднувати у одній структурі значення різних типів і тому вважаються найбільш гнучким механізмом побудови баз даних.

Розповсюдженим випадком є необхідність описати характеристики деякого об'єкта, що обробляється програмою. Таким об'єктом може бути людина, одиниця обладнання, документ чи інший об'єкт, інформація про який не вичерпується одним числом. В усіх таких випадках властивості об'єкта представляються значеннями різних типів і тому для їх опису неможливо використовувати масиви. Так для опису об'єкта "студент" можуть знадобитися, наприклад, такі характеристики:

- прізвище, ім'я та по-батькові – в одній змінній або в трьох окремих (тип String);
- факультет (String);
- спеціальність (шифр та/або назва; String);
- група (String);
- стать (Чоловіча, Жіноча);
- спортивні або навчальні результати (числові або булеві типи).

Для зберігання такої різноманітної, але логічно зв'язаної інформації зручно використовувати комбінований тип. Необхідно відзначити, що в даному випадку окремі компоненти комбінованого типу, оскільки вони мають різну природу, не можуть бути ідентифіковані порядковими номерами

(індексами), як в масивах; тому для їх позначення використовують ідентифікатори. Таким чином, опис комбінованого типу є список описів його елементів (які називаються полями запису); кожен опис схожий на опис простої змінної. Список полів починається службовим словом **record** і має завершуватись службовим словом **end**. Наприклад, варіант втілення наведеного вище опису комбінованого типу може мати вигляд запису Person:

```
type
  Person = record
    Name, SecondName, SurName : string;
    Faculty : string;
    Speciality : word;      {використовуємо шифр
спеціальності}
    Group : String;
    Sex : (Male, Female);
    Speciality : word;
    LastTestPassed : Boolean;
    Result1000m : real
  end;
```

Таким чином, структура запису складається з фіксованої кількості компонентів (полів), кожне з яких має власне (унікальне в межах запису) ім'я і довільний тип. Унікальність імен полів необхідна тільки усередині запису – інший запис може мати поля з таким самим іменем.

Описавши такий запис, в програмі можна використовувати змінні даного типу, наприклад:

```
Var Sasha, Masha : Person;
```

Доступ до елементів (полів) запису проводиться за допомогою конструкції, яка називається селектором запису. Вона має такий вигляд:

```
R . F,
```

де **R** - змінна комбінованого типу, **F** - ідентифікатор поля. Для змінних, описаних вище, допустимими є такі конструкції:

```
Sasha.Name := 'Олександр';
```

```

Masha.Name := 'Марія';
Sasha.Group := 'ФК-43';
Masha.Sex := Female;
Masha.Speciality := Sasha.Speciality;
Sasha.Result1000m:=200.

```

Комбіновані типи можна застосовувати для побудови інших, більш складних структур, наприклад, масивів, що складаються з записів, або записів, у склад яких, в свою чергу, входять записи. Перший випадок можна продемонструвати такими прикладами:

```

Var Group : array[1..25] of Persons;
  GroupeBase : file of Person;

```

Для змінної **Group** доступ до полів записів, що складають цей масив, проводиться таким чином:

```

Group[j].Group:= 'ФК-43' ;
if Group[j].Name='Борис' then
WriteLn(Group[j].SurName) ;

```

Розглянемо тепер випадок, коли у складі запису містяться поля, які також мають тип запис. Нехай для комбінованого типу **Person** необхідно зберігати інформацію про дату народження людини. Цю інформацію можна представити у вигляді трьох полів у складі типу **Person**. Однак, логічніше для цього визначити окремий тип, так як у цьому випадку він може використовуватись у описі інших типів та змінних:

```

Type
  Date = record
Month:
  (Jan, Feb, Mar, Apr, May, June, July, Aug, Sept, Oct, Nov, Decem) ;
Day : 1..31;
Year : 1900..2010 end;

```

Тепер введений таким чином тип можна використовувати в описі запису **Person**:

```

Type Person = record
  Name, SecondName, SurName - string [20] ;

```

```
Speciality : word;  
BirthDay : Date end;
```

Доступ до полів з елемента **BirthDay** проводиться за загальними правилами, наприклад:

```
Sasha.BirthDay.Year := 1970;  
Masha.BirthDay.Month := Feb;
```

Таким чином, при записі селектора необхідно пам'ятати, що зліва від символу "крапка" завжди повинна знаходитися змінна типу запис, а справа - ідентифікатор поля цього запису.

Для більш компактного запису селекторів Pascal має спеціальний оператор приєднування, що дозволяє в ряді випадків пропускати ліві частини селекторів.

```
With Sasha do begin  
  With BirthDay do begin  
    Year:= 1977;  
    Month:=Jul;  
    Day:=7  
  end;  
End;
```

Після опису в програмі деякий комбінований тип може використовуватись так само, як і інші типи – скалярні чи складні. Крім утворення з його участю інших типів, комбінований тип може використовуватись для специфікації параметрів підпрограм. Наприклад, можна описати спеціальний тип для представлення комплексних чисел як пари дійсних змінних (дійсна та уявна частини комплексного числа):

```
Type Complex = record  
  Re, Im : real  
end;
```

Далі можна за допомогою процедур визначити операції над комплексними числами (додавання, множення, ділення):

```
procedure AddC ( C1, C2 : Complex; var R : Complex);  
procedure MultC ( C1, C2 : Complex; var R : Complex);
```

```
procedure DivC ( C1, C2 : Complex; var R : Complex);
```

Записи з варіантами

Поняття запису в Pascal передбачає одну специфічну можливість. Досить часто буває необхідним в межах одного запису мати різну інформацію в залежності від конкретного значення деякого поля.

Розглянемо ще раз тип `Person`, що містить інформацію про людину. Якщо в цьому типі поле **Sex** (Стать) має значення **male** (чоловіча), то необхідно передбачити такі поля: результат бігу на 3000 м (**Result3000m**) та результат стрибка в довжину (**ResultLongJump**); якщо ж поле має значення **female** (жіноча), то необхідно передбачити результати бігу на 1000 м (**Result1000m**) та на 100 м (**Result100m**).

Таке завдання можна вирішити, якщо застосувати записи з варіантами.

Будь-який комбінований тип, окрім фіксованого списку полів, може містити варіантну частину, яка визначає декілька варіантів структури такого типу. Це означає, що різні змінні, хоч і відносяться до одного й того ж типу, можуть мати різну структуру. Різниця може стосуватись як кількості компонентів, так і їх типу.

Варіантна частина містить декілька альтернатив, в кожній з яких, в круглих дужках задається список полів, що присутні в даному варіанті. Перед списком ставиться мітка, що є конкретним значенням поля, яке буде критерієм вибору варіантів. Перелік альтернатив починається з визначення цього поля.

В описаній вище ситуації з типом `Person` запис з варіантами може мати такий вигляд:

```
type
  StudentSex = ( Male, Female ) ;
  Person = record
    Name, SecondName, SurName : string[20];
    Speciality : word; Birthday : Date;
    case Sex : StudentSex of
```

```
Male: (Result3000m: real; ResultLongJump: real);
Female : (Result1000m: real; Result1000m:real)
end;
```

Варто звернути увагу на наступні особливості:

1) Початок варіантної частини відзначається службовим словом `case`; після визначення поля-ознаки вибору варіантів записується службове слово `of`. Варіантна частина завершується службовим словом `end` разом з завершенням усього запису. Таким чином, у описі комбінованого типу може бути тільки одна варіантна частина і її потрібно задавати в кінці опису запису.

2) Альтернативи варіантної частини позначаються допустимими значеннями поля `Sex`; визначення типу цього поля винесено з фіксованої частини та розміщено у заголовок варіантної частини. Поле, значення якого задають варіанти, називають дискримінантом запису.

3) Для того, щоб поле `Sex` могло служити дискримінантом, його тип має задаватися ідентифікатором. Тому перелічувальний тип для цього поля нам довелося описати окремо, а у заголовку варіантної частини використовувати ідентифікатор `StudentSex`.

4) Ідентифікатори полів в усіх варіантах повинні бути різними і відрізнятися від ідентифікаторів полів фіксованої частини.

5) Для деяких значень поля-дискримінанта варіант може бути відсутній. Тоді після двокрапки може знаходитися порожній список виду `()`.

Записи з варіантами є гнучким інструментом для представлення інформації, яка частково відрізняється структурою.

Додатково необхідно відзначити, що будь-який варіант, в свою чергу, може мати свою варіантну частину, яка також розміщуватиметься у кінці списку полів даного варіанта.

Однак, при використанні варіантних записів необхідно враховувати такі особливості:

1. Для розміщення змінної комбінованого типу завжди відводиться фіксований об'єм пам'яті, причому, якщо в записі є варіанти, то цей об'єм визначається за максимальним варіантом. Таким чином, різні варіанти одного запису ніби "накладаються" один на одного в пам'яті - займають одну й ту ж область.

2. Pascal не містить ніяких засобів контролю за правильністю роботи з варіантами записів. Тому, створивши запис, як показано вище, в програмі можна застосувати:

```
Masha.Result3000m := 725.3;
```

Причому програма виконає таке присвоєння і не згенерує помилки. Таким чином, реальні результати можуть бути спотвореними.

Завдання

Розробити, скласти та відлагодити програму за вказівкою викладача з використанням даних типу запис, наприклад, відомість академічної групи з аналізом успішності.

Зробити висновки.

Рекомендована література: [3], [5], [9].