

Лектор: А.Д.Хапугин

Основы программного конструирования

Лекция 8-2. Модуля-2, конструирование типов

Материалы доступны в Интернете по адресу: <http://www.excelsior.ru/afti/>

Типы данных

VAR x: CARDINAL;

VAR x: ИмяТипа | ОписаниеТипа;

TYPE ИмяТипа = ОписаниеТипа;
VAR x: ИмяТипа;

Какие можно использовать описания? И зачем? И что это значит?

Тип данных:

- имя или конструкция
- что можно присвоить
- какие операции
- как устроено
- зачем это нужно
- тонкости, приемы, и т.п.

Перечислимый

```
TYPE T = (A,B,C,D,E,F);  
VAR x,y: T;
```

Что присвоить?

x:=y; x:=A; x:=B; и т.п.

Операции?

а) Сравнения:

=, #, >, <, >=, <=

б) INC(x); DEC(x);

в) MIN(T), MAX(T)

г) ORD(x) -> CARDINAL

Как устроено?

При описании компилятор просто задает числовые значения для каждой из констант.

Размер памяти — не больше чем у целых и ни чего другого нельзя предполагать.

Размер может быть меньше, если компилятор сочтет это выгодным.

Перечислимый - продолжение

Зачем?

Сравните:

```
TYPE WeekDay = (Monday, Tuesday, Wednesday, Thursday, Friday,  
Sarurday, Sunday);  
VAR w: WeekDay;
```

или так:

```
CONST  
    Monday = 0;    Tuesday = 1;  
    Wednesday = 2; Thursday = 3;  
    Friday = 4;    Sarurday = 5;  
    Sunday = 6;  
VAR x: CARDINAL;
```

Что можно присвоить в w а что — в x?

Интересно, что:

```
BOOLEAN = (FALSE, TRUE);
```

Но из-за того, что BOOLEAN — встроен, есть существенная разница. Какая?

Диапазон

```
TYPE T = [0..153];  
VAR x,y: T;
```

Что присвоить?

```
x:=y;  
x:=число или выражение;
```

Есть «базовый тип».

Для $T = [-153..153]$ — это INTEGER;

Для $T = [0..153]$ — CARDINAL;

Для $T = \text{INTEGER}[0..153]$ - INTEGER;

Операции?

а) арифметика — как у целых (знаковых или беззнаковых). В выражении можно смешивать со значениями и своего базового типа.

а) Сравнения:

=, #, >, <, >=, <=

б) INC(x); DEC(x);

в) MIN(T), MAX(T)

Диапазон - продолжение

Как устроено?

Так же как соответствующий базовый тип. Разница только в том, что перед присваиванием выполняется проверка значения.

Зачем?

Сравните:

```
TYPE Month = [1..12];  
VAR m: Month;  
    m1: CARDINAL;
```

Что можно присвоить в m а что — в x?

Тонкости:

```
TYPE WorkDay = [Monday..Friday];  
Базовый тип — перечисление, последствия?
```

```
TYPE Capitals = [«A» .. «Z»];  
Базовый тип — CHAR. Последствия?
```

Массив

```
TYPE T = ARRAY [0..153] OF CHAR;  
VAR x,y: T;
```

Что присвоить?

```
x:=y;
```

Можно ли так?

```
VAR  
  x: ARRAY [0..153] OF CARDINAL;  
  y: ARRAY [0..153] OF CARDINAL;  
  . . .  
  x:=y;
```

Можно ли так?

```
TYPE  
  T = ARRAY [0..153] OF CARDINAL; T1 = ARRAY [0..153] OF CARDINAL;
```

```
VAR x: T; y: T1;
```

```
BEGIN
```

```
  x:=y;
```

Массив - 1

В общем виде

TYPE T = ARRAY ТипИндекса OF ТипЭлемента;

ТипИндекса — какой?

Любой, где осмыслено отношение порядка. То есть: а) диапазон, б)перечислимый, в) CHAR.

ARRAY [WeekDay] OF CARDINAL
ARRAY CHAR OF CHAR;

Можно ли просто CARDINAL?

ТипЭлемента — какой?

Любой.

А то если тип элемента - массив?

T2 = ARRAY [1..20] OF ARRAY [1..20] OF CARDINAL;

T3 = ARRAY [1..20],[1..20] OF CARDINAL;

Как обозначить элемент?

x[i],

какого типа i?

Массив - 2

Операции?

Никаких.

Операции над элементом — те, которые допустимы для ТипаЭлемента.

Как устроено?

Сплошной кусок памяти, по возможности компактно, но возможно и выравнивание.

Доступ:

вычисление адреса по базовому адресу и индексу.

Выравнивание?

TYPE

```
S = ARRAY [1..10] OF CHAR;
```

```
T = ARRAY [1..20] OF S;
```

«Хвостики» по 2 байта после каждого S будут не заняты.

Зачем?

На примере сбора статистики — или придумать 153 имени и потом мучиться с сопоставлением имен и кодировок, или имя одно а выбор элемента для доступа — мало текста и быстро работает.

Множества

TYPE

T = SET OF ТипЭлемента

ТипЭлемента — какой?

Любой, где осмыслено отношение порядка, то есть а)диапазон, б)перечислимый, в)CHAR.

Можно ли CARDINAL?

Что присвоить?

x:=y; x:={«A» .. «Z»}

TYPE

W = (sun,mon,tue,wed,thu,fri,sat); WD = [mon..fri];

SD = SET OF W; SW = SET OF WD;

VAR w: SD; wd: SW;

BEGIN

w:=SD{tue..thu};

wd:=SW{sun..thu}; - ошибка

END sets.

Множества - 1

Операции?

Те же, что и на BITSET.

а) действия +, -, *, / б) сравнения =, #, >=, <=, в) доступ к битам INCL, EXCL, г) проверка бита IN

Как реализовано?

Как массив из битсетов, а для операций пересчет.

Сколько занимает памяти?

Как минимум — столько бит, сколько значений в базовом типе.

Отсюда и ограничения на размер. Но SET OF CHAR — точно должен быть возможен.

Зачем?

На примере SET OF CHAR — признаки символов — строчная, прописная, цифра, пробел

Тонкости?

а) BITSET = SET OF [0..31];

Но есть особенность, связанная с битовым представлением числе в тех же словах.

б) отменено в более поздних языках.

Конец лекции.