

Лектор: А.Д.Хапугин

Основы программного конструирования

Лекция 10-2. Модуля-2, конструирование типов

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

Тип - запись

Зачем?

Дата — это (день, месяц, год), где
день — [1..31],
месяц — [1..12],
год — CARDINAL.

Студент — это (ФИО, факультет, группа, номер зачетки), где:
ФИО — строка
факультет — (ФФ,ММФ,ФИТ,ЭФ,ФЕН,ГФ)
группа — CADINAL
номер зачетки — CARDINAL

Хочется в одной переменной объединять данные разных типов, например, чтобы сортировать массив студентов по ФИО или по номеру зачетки...

Запись: описание

TYPE

DATE = RECORD

day: [1..31];

month: [1..12];

year: INTEGER; (* до нашей эры тоже было время *)

END;

Student = RECORD

fiо: ARRAY [0..99] OF CHAR;

group: CARDINAL;

END;

VAR x,y: DATE;

1. Видимость имен day,month,year?

VAR year, group: INTEGER; - можно?

Запись: что присвоить?

Что присвоить?

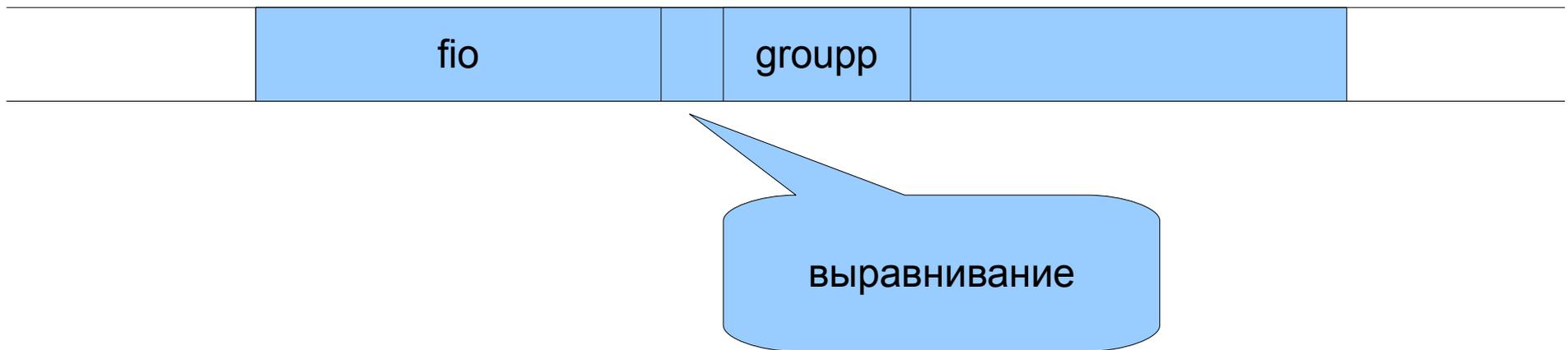
`x:=y;` -- больше ничего.

Для переменных-записей набор операций определяется полями.

`x.day:=31; x.year:=2008;`

Запись: как устроено?

Для хранения переменных выделяется сплошной кусок памяти.



$X.groupp$ — вычисляется так:

- базовый адрес переменной X плюс статическое (то есть известное во время компиляции) смещение

Запись с вариантами

Зачем?

Например, делаем графический редактор.

Есть несколько типов фигур:

TYPE

```
FigType = (Line, Rectangle, Ellipse, Triangle);
```

```
Fig = RECORD
```

```
  x,y: CARDINAL;
```

```
  CASE type: FigType OF
```

```
    |Rectangle, Ellipse,Line: w,h: CARDINAL;
```

```
    |Triangle: x1,y1,x2,y2: CARDINAL;
```

```
  END;
```

```
END;
```

```
VAR x: Fig;
```

Ограничения:

- нельзя одинаковые имена в вариантах

- нет никакого динамического контроля, то есть:

f.type:=Ellipse; f.x1:=0; f.y1:=0; - такое вполне допускается компилятором

Запись с вариантами: как устроено

На примере Fig.

| | |
|-----------|----------------------------|
| x,y, type | Либо w,h, либо x1,y1,x2,y2 |
|-----------|----------------------------|

Получается, что w может оказаться там же где и x1.

Это и есть самая главная слабость такой конструкции.

В более поздних языках это заменено на классы и наследование.

Запись с вариантами: частный случай

```
TYPE
  Something = RECORD
    CASE : BOOLEAN OF
      |FALSE: card: CARDINAL;
      |TRUE:  set: BITSET;
    END;
  END;
```

Это способ «наложить» 2 разных типа друг на друга.

Указатели

Цитата:

Язык Модуля-2 предлагает базовое средство для построения произвольных структур: указательный тип или указатель.

Указатели используются для установления связей между структурированными элементами. Указатели связывают элементы или указывают на элементы. Очевидно, что различные указатели могут указывать на один и тот же элемент, тем самым обеспечивая возможность создавать сколь угодно сложные структуры и в то же время открывая массу возможностей для совершения трудноуловимых ошибок. Работа с указателями требует чрезвычайной тщательности и аккуратности.

Указатели в Модуле не могут указывать на произвольные переменные...

Понятно?

Контрольные вопросы:

Кому указатели указывают?

Что такое этот самый элемент, на который указатели указывают?

Что значит - «указывают»?

Давайте по порядку...

Вспомним, что вообще такое типы, и зачем их придумали?

И как это работает? (компилятор - ограничения)

Указатели: зачем?

UNDO в редакторе?

Стек для quicksort'a.

И там и там вопросы: сколько элементов завести?

Статический — это такой, который определяется полностью до исполнения программы. Пример: `VAR x: ARRAY...` - размер виден из описания, сразу известно, что для самой переменной `X` надо завести память и это будет ровно один объект.

А что если мы не знаем наперед, сколько объектов нам понадобится?

Примеров — сколько угодно.

Указатели: описание

TYPE

P = POINTER TO BaseType;

VAR x,y: P;

Что присвоить?

x:=y;

x:=NIL;

Сравнения = и #

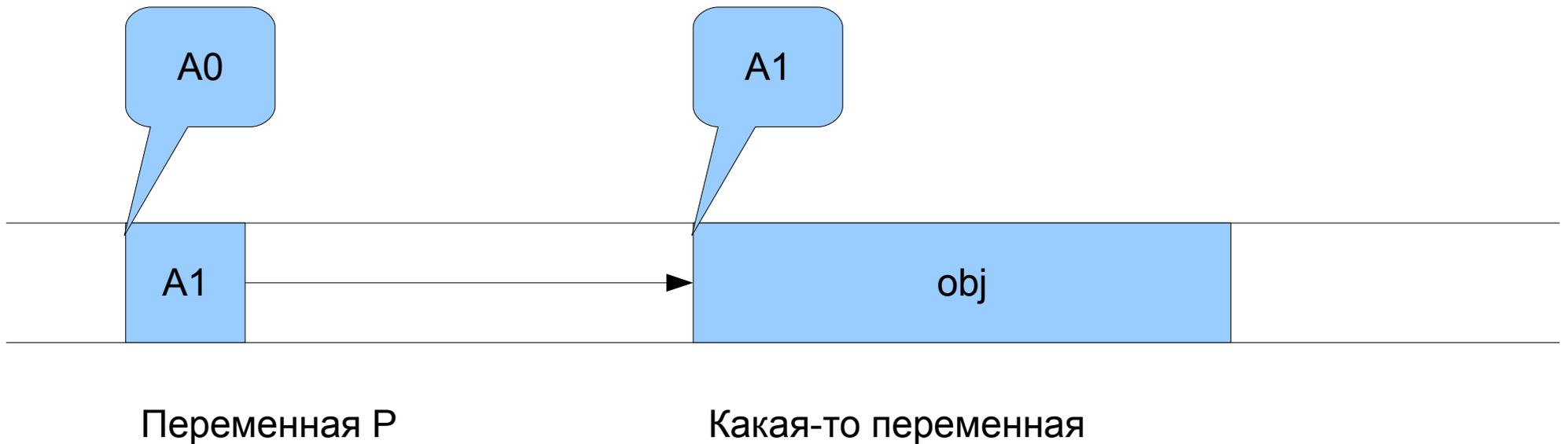
SYSTEM.ADDRESS

SYSTEM.ADR(любой объект)

PROCEDURE ALLOCATE (VAR addr: SYSTEM.ADDRESS; amount: CARDINAL);

PROCEDURE DEALLOCATE (VAR addr: SYSTEM.ADDRESS; amount: CARDINAL);

Указатели: как устроено



Слово в «p» содержит адрес переменной obj.

P^{\wedge} - это просто obj

Указатели: тонкости

NIL — это одно значение, а «неправильных» адресов — много.

Ошибки работы с динамической памятью:

- использование невыделенной памяти
- сохранение указателя на освобожденную память

Типы: промежуточные итоги

Обсудили:

- CARDINAL
- INTEGER
- CHAR
- BOOLEAN
- BITSET
- REAL/LONGREAL
- SET OF
- перечислимые типы
- отрезки (диапазоны)
- записи
- указатели

Для полноты картины остались только процедурные типы, которые отложим ненадолго.