Информатика

Лекция 11

Агрегированный тип

(типы и структуры данных)

Понятие агрегата

- Агрегат данных – совокупность необязательно однородных объектов, составляющих в определенном смысле единое целое
- Агрегат данных обычно оперативно доступен как в целом, так и по частям
- Агрегированный тип: составной, комбинированный
- Пример – записи, таблицы, фреймы

Математическая модель

$$A \times B = \{ (a, b) \mid a \in A \text{ и } b \in B \}$$

- Прямое (Декартово) произведение множеств для произвольного количества сомножителей

  $D_1 \times D_2 \times \ldots \times D_k = \{ (d_1, d_2, \ldots, d_k) \mid d_1 \in D_1, d_2 \in D_2, \ldots, d_k \in D_k \}$
- Метод индукции:

Записи

- *Запись* – составной тип данных, значение которого состоят из поименованных компонент разных типов
- *Поле* – элемент данных (участок памяти) – данные, рассматриваемые как нерасчленяемое целое при обмене и хранении
- В базах данных *запись* – поименованный агрегат данных, а *поле* – минимально именуемая порция данных

Сравнение

- Запись, как и массив – это сложная переменная с несколькими компонентами, которые могут иметь разные типы, но доступ к ним осуществляется по имени, а не по индексу
- Запись и массив имеют общее свойство: обе являются структурами «со случайным доступом»
- Массив предоставляет большие возможности, поскольку селекторы его компонент могут вычисляться, тогда как селекторы компонент записи – это фиксированные имена, задаваемые в описании типа

Составной тип (1)

- Самый общий метод получения составных типов – это объединение компонент, принадлежащих произвольным, возможно, также составным типам, в один комбинированный (составной) тип
- Примеры:
  - комплексные числа, состоящие из двух вещественных пар
  - координаты пространства в зависимости от его размерности
  - информация о студентах в базе данных университета и т. п.

Составной тип (2)

- Для представления разнородной, но логически связанной информации удобно использовать комбинированный (составной) тип
- Объект – человек – может описываться следующими характеристиками:
  - *Фамилия, имя, отчество* (строки)
  - *Пол* (перечислимый тип из двух значений)
  - *Индекс специальности* (целое)
  - *Номер группы* (строка) и т. д.
- Для обозначения компонент используются идентификаторы (составные имена)

Конструктор записи

- Описание комбинированного типа представляет собой список описаний его элементов (которые также называются полями записи), причем каждое описание похоже на описание простой переменной
- Список открывается служебным словом **record** и должен завершаться служебным словом **end**
- type Person = record
  Name, SecName, SurName: string [20];
  Gender: (Male, Female);
  Speciality: word;
  end;

Селектор записи

- Структура запись состоит из фиксированного числа полей (компонент), каждое из которых имеет собственное (*уникальное в пределах одной записи*) имя и произвольный тип
- Идентификаторы полей могут совпадать с другими идентификаторами текущего блока, а также с идентификаторами полей других записей при этом *не возникает* конфликта имен, поскольку идентификатор поля всегда выступает в паре с переменной – записью
- Доступ к полям записей имеет следующий общий вид: **R.F**
  где **R** – переменная комбинированного типа,
  F – идентификатор поля

Пример

```
type Person = record
                Name, SecName, SurName: string [20];
                Gender: (Male, Female);
                Speciality: word;
              end;
 var Sasha, Masha: Person;
   begin
          Sasha.Name := ‘Александр’;
          Masha.Name := ‘’;
          Masha.Gender := ‘Female’;
          Sasha.Gender := ‘Male”;
          Masha.Speciality := Sasha.Speciality;
    end.
```

Комбинированные типы

- Комбинированные типы используются для построения более сложных структур, например, массивов, состоящих из записей, в состав которых, в свою очередь, входят записи
-
```
type Person = record
            Name, SecName, SurName: string [20];
            Gender: (Male, Female);
            Speciality: word;
          end;
  var
          Group: array [1..25] of Person;
          Database: file of Person;
     begin
       …
          Group [i].Gender := ‘Female’;
  if Group [j].Name=‘Борис’ then writeln(Group [j].SurName);
       …
     end.
```

Запись - дата рождения

- Тип дата может использоваться в описании других типов и переменных
- type

```
        Date = record
             Day: 1 .. 31;
             Month: (Jan, Feb, Mar, Apr, May, June,
                        July, Aug, Sept, Oct, Nov, Dec);
             Year: 1900 .. 2050
         end;
        Person = record
            Name, SecName, SurName: string [20];
            Gender: (Male, Female);
            Speciality: word;
            Birthday: Date
        end;
begin
…
        Sasha.Birthday.Year := 1970;
        Masha.Birthday.Month := ‘Feb’;
…
end.
```

Вариантные записи

- Запись может иметь вариантную часть, изменяющуюся при разных реализациях (она размещается в конце описания записи, начинаясь со служебного слова **case** , за которым следует переменная выбора варианта с указанием ее типа, далее указываются константы, значение которых может принимать переменная, а в круглых скобках поля данного варианта и их типы) *круглые скобки необходимы и в том случае, когда в варианте поле отсутствует*
- Поле, значение которого задают варианты, иногда называют *дискриминантом* записи

Пример

```
type
        figure = (square, triangle, circle);
        param = record
              x, y: real;         {точка привязки}
             case fig: figure of     {параметры фигуры}
                    square: (side: real);
                    triangle: (side1, side2, angle: real);
                    circle: (radius: real)
         end;
var  mysquare, mycircle, mytriangle: param;
begin
…
          fig := square; mysquare.side := 5.0;
          fig := circle; mycircle.radius := 7.5;
…
end.
```

Замечания

1. Все имена полей должны быть различными, в том числе в разных вариантах одной записи
2. Если вариант пустой, т. е. поля нет, то он записывается в виде: V();
3. Любой список полей может иметь только одну вариантную часть, которая должна следовать за фиксированной частью
4. Каждый вариант может содержать в себе вариантную часть (вложенные варианты)

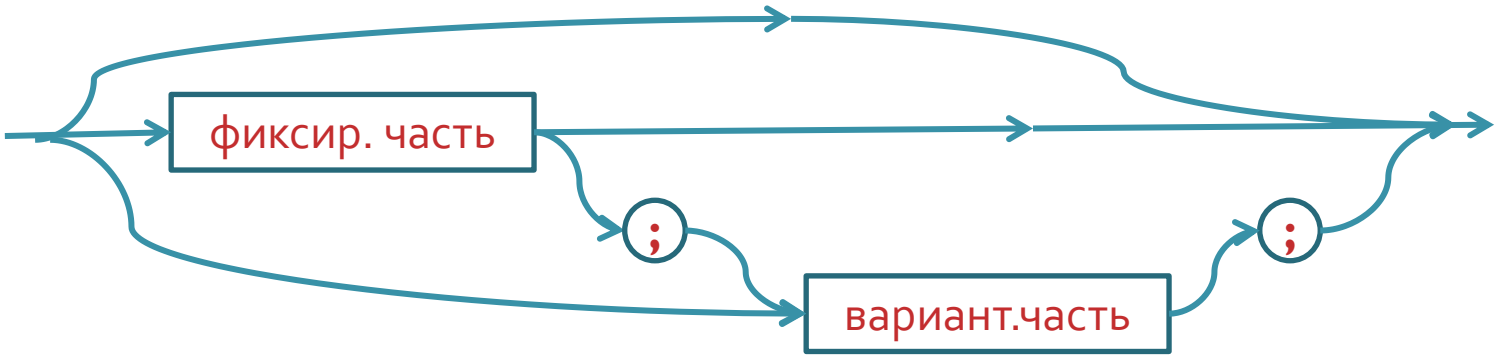
Записи с вариантами

- Вариантные записи имеют дополнительные поля, зависящие от значений других полей
- Вариантная часть позволяет задавать определение нескольких вариантов структуры (иерархия), содержащих список атрибутов, перед которым стоит заголовок (дискриминант) с указанием общего типа этих атрибутов

Синтаксические диаграммы

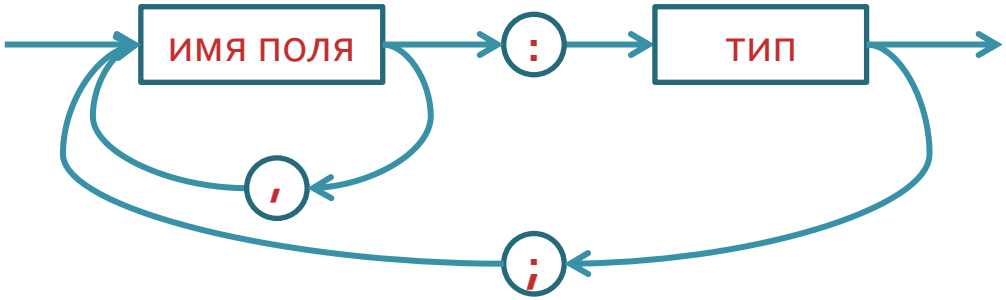
- Запись

- Список полей

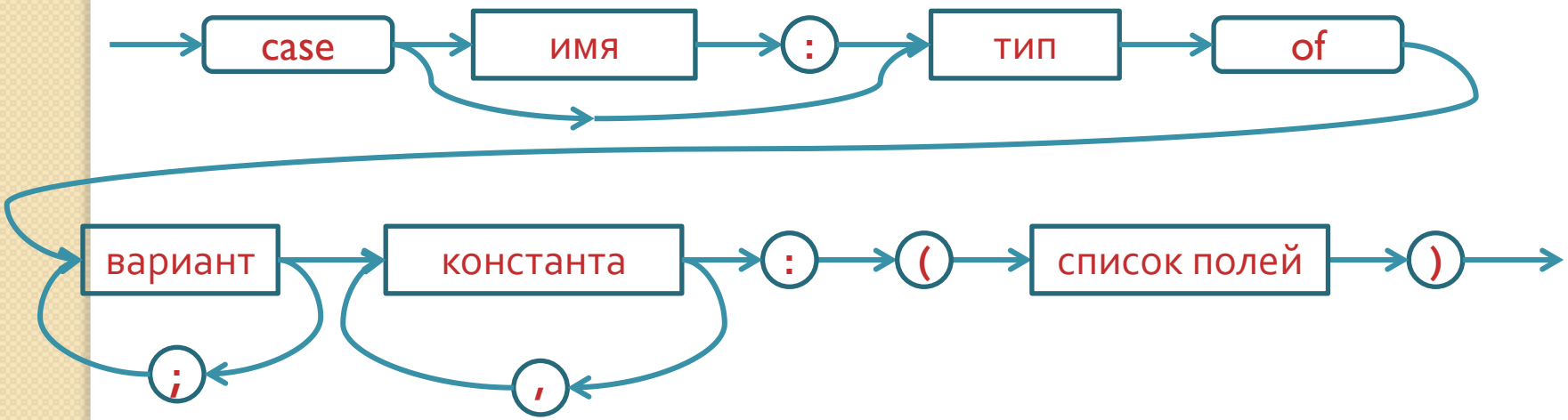


Синтаксические диаграммы

- Фиксированная часть

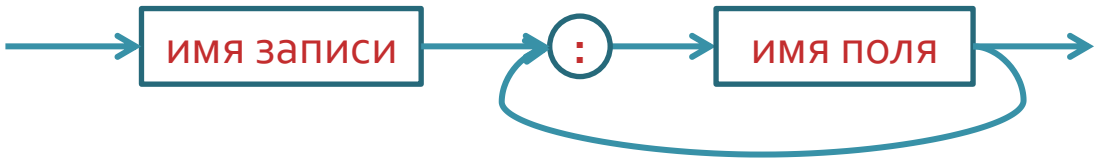


- Вариантная часть



Селектор записи

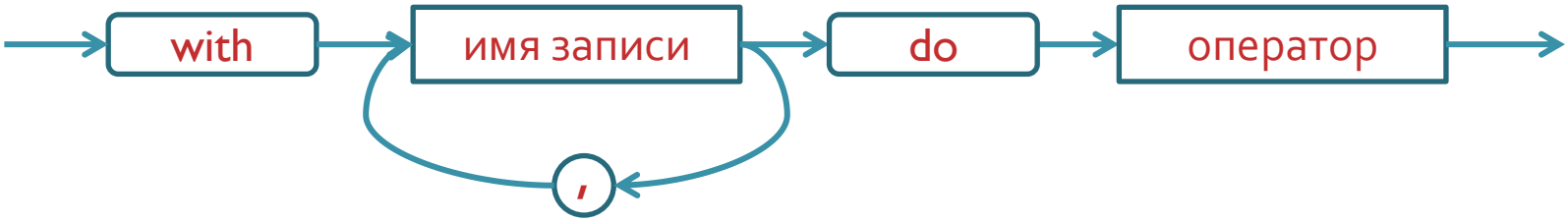
- Для переменных одного комбинированного типа можно использовать оператор присваивания A := B;
  Обращение к конкретному полю выполняет конструкция R.F, причем имя поля в отличие от индекса вычислить нельзя



возможны обращения вида:
F[i+1].D; G[2].P[k].M; F[i+1].D.E;

Оператор присоединения

- При многократном обращении к нескольким компонентам одной и той же записи можно использовать присоединение имени записи



Внутри оператора используют только имена полей

Текст программы (Pascal)

```
program Marry; const n=125;
 type complex = record
              Re: real; Im: real       end;
          person = record
             Name: string [30];
             Gender: (Male, Female);
             Age: 1 .. 150;
             Marriage: boolean       end;
var x, y: complex; a: array [1..n] of person;  count: 0..27;
   begin count := 0;
              for i := 1 to n do
              if (a[i].Gender = Female) and (a[i].Marriage) then
                   count=count + 1;
   end.
```

Цикл лекций подготовлен в 2013/2014уч. году Кузнецовым Игорем Ростиславовичем, доцентом кафедры радиоэлектронных средств Санкт-Петербургского Государственного электротехнического университета «ЛЭТИ»

Прочитан в дисциплине
«Информатика»