



# Информатика

Лекция 6

## Типы и структуры данных

# Определения

- «Хотя операции и данные – эти активные и пассивные компоненты играют совершенно различные роли в каждой конкретной программе, совершенно невозможно рассматривать одни в отрыве от других»

*Турский В.*

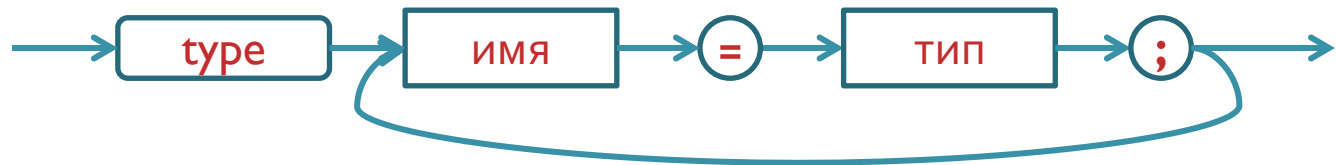
- Данные – это объекты, обладающие *именем* и *значением*
- Структура – совокупность устойчивых связей между частями объекта, обеспечивающих его целостность, т.е. сохранение его свойств при внешних и внутренних изменениях
- Структура данных – способ объединения нескольких элементов данных в один (вектор, матрица, множество, список, файл и т.п.)
- Структуры строятся из значений компонент с помощью *конструкторов*, а значение компоненты извлекается с помощью *селекторов*
- Каждому методу структурирования данных соответствует своя пара <конструктор - селектор>, обозначения которых четко различаются

# Концепция типа

- Тип – определенное множество объектов
- Тип данных – множество значений, которые может принимать переменная или выражение в совокупности со множеством операций над этими значениями, определенными так, что:
  - каждое значение принадлежит одному и только одному типу
  - тип значения константы, переменной или выражения можно вывести из свойств программы
  - каждой операции соответствует определенный тип операндов и определенный тип результата

# Синтаксическая диаграмма

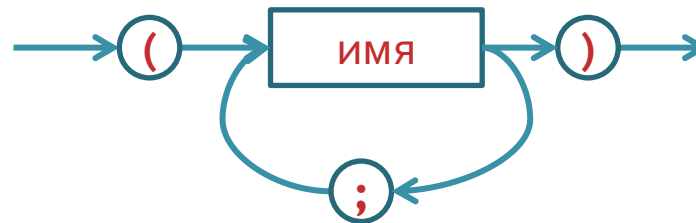
- Задание типа



- имя – название типа, обозначение множества значений
- тип – задание множества значений данного типа (способ задания всех констант)

# Перечислимый (номинальный) тип

- Синтаксическая диаграмма



- перечислимый тип – список значений, которые могут принимать переменные данного типа
- значения упорядочены и для них определены стандартные процедуры:
  - $\text{ord}(\text{имя})$  – порядковый номер значения
  - $\text{succ}(\text{имя})$  – последующее значение
  - $\text{pred}(\text{имя})$  – предшествующее значение

# Перечислимый (номинальный) тип

- Совокупность значений определяется правилом  
 $(c_i < c_j) \equiv (i < j)$
- Так как значения упорядочены, то доступны операции отношения `| < | > | ≤ | ≥ | = | ≠ |`
- Специальных операций над переменными перечислимого типа нет, но можно использовать операторы присваивания  
< переменная перечислимого типа > := <одно из значений>
  - {type t1=(one, two, three);  
    day=(mo, tu, we, th, fr, sa, su);  
    color=(red, blue, green);  
    var x : t1; y, z : day; a, b : color;  
    x:=two; z:=mo; a:=red}
  - type figure=(прямоугольник, квадрат, круг, эллипс);  
    structure=(файл, массив, запись, множество);

# Ограниченный (интервальный) тип

- Синтаксическая диаграмма



- Интервальный тип – наложение ограничений на тип значений, который является упорядоченным (константы указывают диапазон значений или границы интервала, причем первая должна быть меньше второй)
- Базовые типы – **integer, boolean, char**, а также перечисляемые типы, заданные в программе
- Значения ограниченного типа могут использоваться везде, где могут использоваться значения базового типа

# Ограниченный (интервальный) тип

- type year = 1900..1999; letter = 'A'..'Z';  
digit = '0'..'9';  
var y : year; L : letter;  
y := 1973; L := 'W';  
y := 1291; L := '9';
- Допустимость присваиваний вида  
y := i (целый); L := c (символьный);  
можно проверить только во время  
выполнения программы
- Ограниченный тип часто используется в  
определении таких структур, как массивы  
type index = 1..20;  
var i, j, k, m, l : index;



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

Прочитан в дисциплине  
«Информатика»