

Множественный тип данных

Наумов Д.А., доц. каф. КТ, ИТГД

Алгоритмические языки и программирование, 2019

Содержание лекции

- 1 Элементы теории множеств
 - Понятия и определения
 - Отношения между множествами
 - Операции над множествами
- 2 Множественный тип в языке Pascal

Множество — одно из ключевых понятий математики; это математический объект, сам являющийся набором, совокупностью, собранием каких-либо объектов, которые называются элементами этого множества и обладают общим для всех их характеристическим свойством.

Множество может быть

- пустым и не пустым;
- упорядоченным и не упорядоченным;
- конечным и бесконечным;
- бесконечное множество - счетным и несчетным;

Элементы множества

объекты, из которых состоит множество.

Принадлежность элемента a множеству A

$$a \in A$$

Не принадлежность элемента a множеству A

$$a \notin A$$

Равенство двух множеств $A = B$

$$x \in A \iff x \in B$$

Способы задания множеств

- перечисление $Y = \{0, 2, 4, 8\}$;
- описание $Y = \{x \in X | A(x)\}$;

A включено в B , если каждый элемент A принадлежит B

$$A \subseteq B \Leftrightarrow \forall a \in A : a \in B$$

A включает B , если B включено в A

$$A \supseteq B \Leftrightarrow B \subseteq A$$

A равно B , если A и B включены друг в друга

$$A = B \Leftrightarrow (A \subseteq B) \wedge (B \subseteq A)$$

- $A = A$;
- если $A = B$, то $B = A$;
- если $A = B$, $A = B$, то $A = C$;

A строго включено в B , если A включено в B и не равно ему

$$A \subset B \Leftrightarrow (A \subseteq B) \wedge (A \neq B)$$

A строго включает B , если B строго включено в A

$$A \supset B \Leftrightarrow B \subset A$$

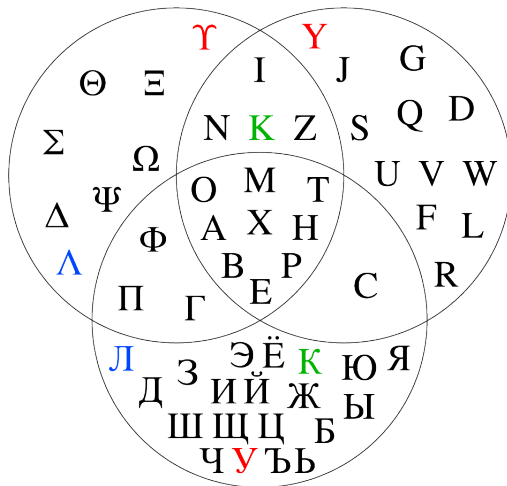
A и B не пересекаются, если у них нет общих элементов

$$AB \Leftrightarrow \forall a \in A : a \notin B$$

A и B находятся в общем положении

$$\exists a, b, c : (a \in A) \wedge (a \notin B) \wedge (b \in B) \wedge (b \notin A) \wedge (c \in B) \wedge (c \in B)$$

Диаграмма Венна (также используется название диаграмма Эйлера — Венна) — схематичное изображение всех возможных отношений (объединение, пересечение, разность, симметрическая разность) нескольких (часто — трёх) подмножеств универсального множества.



Бинарные операции

Пересечение

$$A \cap B := \{x | x \in A \wedge x \in B\}$$

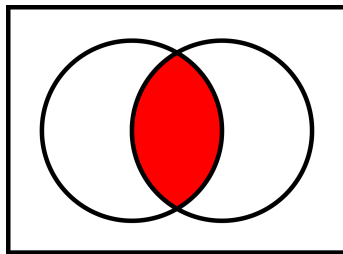


Рис.: Диаграмма Венна для пересечения множеств

Бинарные операции

Объединение

$$A \cup B := \{x | x \in A \vee x \in B\}$$

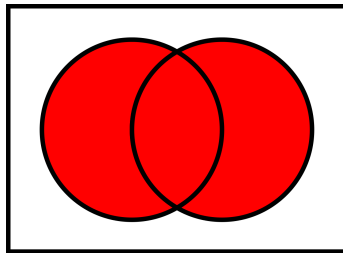


Рис.: Диаграмма Венна для объединения множеств

Бинарные операции

Разность множеств

$$A \setminus B := \{x | x \in A \wedge x \notin B\}$$

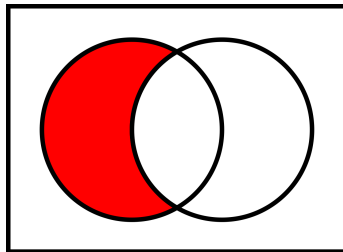


Рис.: Диаграмма Венна для разности множеств

Бинарные операции

Симметрическая разность множеств

$$A \triangle B := (A \setminus B) \cup (B \setminus A)$$

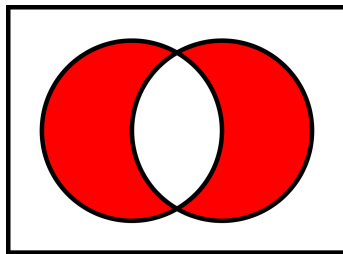


Рис.: Диаграмма Венна для симметрической разности множеств

Множество

структура данных, представляющая ограниченную неупорядоченную совокупность различных элементов одного типа.

```

1  a := []; //пустое множество
2  b := [2, 4, 7, 10]; //множество из четырех элементов целого
3  c := ['e', 'u', 'i', 'o', 'a', 'y']; //множество из шести строчных
4  k := 15; d := [1, k, 2*k]; //множество трех элементов
5  e := [1..100];
6  f := [k..2*k];
7  g := ['A'..'Z', 'a'..'z'];
8  h := [1..1, 5..1]; //эквивалентно [1]
9  j := [5..1]; //эквивалентно []

```

Порядок перечисления элементов в множестве не играет роли.

```

10 m := [5, 3, 1]; //эквивалентно [1, 3, 5]
11 n := [1, z, 2*z]; //эквивалентно [1, 2] при z = 1

```

Множественный тип (тип множества)

структурный тип данных, значением которого является множество.

```
type ИмяТипа = set of БазовыйТип;
```

Базовый тип - любой порядковый (+ ограничения реализации Pascal).

```
1  type
2      TLetters = 'A'..'Z';
3      TLettersSet = set of TLetters;
4      TCharSet = set of char;
5      TIndexSet = set of 1..100;
```

В программе могут быть заданы константы и переменные множественного типа.

```
6  const
7      Vowels = ['e', 'u', 'i', 'o', 'a', 'y'];
```

Конструктор множества

позволяет задать значение переменной или константы множественного типа.

```
1  type TLetters = set of char;  
2  var  
3      EnglishCap, EnglishStr: TLetters;  
4  begin  
5      RussianCap := ['A'..'Z'];  
6      EnglishStr := ['a'..'z'];
```

Переменные и типизированные константы могут участвовать в операциях присваивания, если они принадлежат к идентичным типам.

Операции над множествами

- объединение $[1, 2, 4] + [2, 4, 8]$; //результат $[1, 2, 4, 8]$;
- пересечение $[1, 2, 4] * [2, 4, 8]$; //результат $[2, 4]$;
- разность $[1, 2, 4] - [2, 4, 8]$; //результат $[1]$;

Операции отношения

- равенство $(A = B)$;
- неравенство $(A <> B)$;
- A содержится в B $(A \leq B)$;
- A содержит B $(A \geq B)$;

Принадлежность элемента множеству

- $x \text{ in } A$;

Приоритет операций: пересечение, (объединение, разность), сравнение и принадлежность

Пример: вывести элементы множества

```
1  const
2      Low = 1;      //минимальное значение элемента множества
3      High = 32; //максимальное значение элемента множества
4  type
5      TBase = Low..High; // базовый тип - отрезок Low..High
6      TSet = set of TBase; //множественный тип
7  var
8      SimpleNumber: TSet; //множество чисел
9      Number: TBase; //переменная-элемент множества
10 begin
11     SimpleNumber := [2, 3, 5, 7, 11, 13, 17, 23, 29, 31];
12     for Number := Low to High do
13         if Number in SimpleNumber then
14             write(Number:3);
```


Пример: сформировать множество цифр числа

```
1  type TSet = set of 0..9; //множественный тип
2  var
3      DigitSet: TSet; //множество чисел
4      Digit: 0..9;    //переменная-элемент множества
5      N, M: longint;  //исходное и вспомогательное число
6  begin
7      write('N = '); readln(N); M := N;
8      while M > 0 do begin
9          include(DigitSet, M mod 10);
10         M := M div 10;
11     end;
12     for Digit := 0 to 9 do
13         if Digit in DigitSet then
14             write(Digit:2);
```

Пример: ввести с клавиатуры множество символов

```
1 var
2   c: char;
3   s: set of char;
4
5 begin
6   write('Enter a string (Ctrl+Z - end if input): ');
7   s := [];
8   repeat
9     read(c);
10    if EOLN then
11      break;
12
13    include(s, c); //или так: s := s + [c];
14  until false;
15  readln;
```

Пример: ввести из введенной последовательности только английские буквы

```
16 //пересечение множества s и множества английских букв
17 s := s * ['A'..'Z', 'a'..'z'];
18
19 //выводим множество
20 for c := chr(32) to chr(255) do
21     if c in s then
22         write(c, ' ');
23 writeln;
```

Пример: вывести цифры, входящие в оба числа (M, N)

```
1 program set_02;
2 type
3   TBase = 0..9;
4
5 var
6   Digit: TBase;
7   SetM, SetN, ResultSet: set of TBase;
8   M, N, M1, N1: longint;
9
10 begin
11   write('Input M = ');
12   readln(M);
13   write('Input N = ');
14   readln(N);
15   writeln;
```

Пример: вывести цифры, входящие в оба числа (M, N)

```
16  M1 := M; N1 := N;
17
18  SetM := [];
19  while M1 > 0 do
20  begin
21      SetM := SetM + [M1 mod 10];
22      M1 := M1 div 10;
23  end;
24
25  SetN := [];
26  while N1 > 0 do
27  begin
28      SetN := SetN + [N1 mod 10];
29      N1 := N1 div 10;
30  end;
31
```

Пример: вывести цифры, входящие в оба числа (M, N)

```
33  writeln('Digits in ', M, ':');
34  for Digit := 0 to 9 do
35      if Digit in SetM then
36          write(Digit, ' ');
37
38  writeln; writeln('Digits in ', N, ':');
39  for Digit := 0 to 9 do
40      if Digit in SetN then
41          write(Digit, ' ');
42
43  writeln; writeln('Digits in both numbers:');
44  for Digit := 0 to 9 do
45      if Digit in ResultSet then
46          write(Digit, ' ');
47  readln;
48  end.
```

Упражнения

- определить, является ли последовательность символов корректным идентификатором языка Pascal;
- определить, из скольких различных цифр состоит число;
- сформировать множество простых чисел, не превышающее N ;
- Вывести цифры, не входящие в десятичную запись числа.
- Определить, можно ли из символов одного слова составить другое слово.
- Определить повторяющиеся гласные буквы в тексте.
- Определить, каких символов в тексте больше: гласных или согласных.
- Вывести буквы, входящие в текст только один раз.
- Вывести буквы, входящие в текст более одного раза.
- Вывести буквы, входящие в текст ровно два раза.

- ❶ Множество // Математическая энциклопедия (в 5 томах). — М.: Советская Энциклопедия, 1982. — Т. 3. — С. 762.
- ❷ К. Куратовский, А. Мостовский. Теория множеств / Перевод с английского М. И. Кратко под редакцией А. Д. Тайманова. — М.: Мир, 1970. — 416 с.
- ❸ Н. Бурбаки. Основания математики. Логика. Теория множеств // Очерки по истории математики / И. Г. Башмакова (перевод с французского). — М: Издательство иностранной литературы, 1963. — С. 37—53. — 292 с. — (Элементы математики).
- ❹ Г. Кантор. Труды по теории множеств. — М.: Наука, 1985. — 430 с. — (Классики науки). — 3450 экз..