

ЛАБОРАТОРНАЯ РАБОТА

Тема: “Сложный тип данных – множества”

Цель работы

1. Получение навыков в задании переменных типа множество и организации ввода и вывода данных типа множество.
2. Получение практических навыков в выполнении операций над множествами.

Краткие сведения из теории

9.1. Объявление переменной типа множества

В математике под множеством понимается некоторый набор элементов. Например, множество фигур на плоскости (прямоугольник, круг, ромб, квадрат).

К множествам применимы следующие операции:

1. Объединение множеств ($C = A \cup B$). Каждый элемент множества C является элементом либо множества A , либо множества B .
2. Пересечение множеств ($C = A \cap B$). Каждый элемент множества C является элементом множеств A и B одновременно.
3. Разность двух множеств ($C = A \setminus B$). Каждый элемент множества C является элементом множества A , но не является элементом множества B .

Например:

- а) $\{ \text{круг, ромб} \} \cup \{ \text{круг, квадрат} \} = \{ \text{круг, ромб, квадрат} \};$
- б) $\{ \text{круг} \} \cap \{ \text{круг, ромб, квадрат} \} = \{ \text{круг} \};$
- в) $\{ \text{круг, ромб, квадрат} \} \setminus \{ \text{круг, квадрат} \} = \{ \text{ромб} \}.$

Под множеством в языке Турбо - Паскаль понимают ограниченный, неупорядоченный набор различных элементов одинакового типа.

Множественный тип задается с помощью двух служебных слов SET и OF, после которых указывается базовый тип. В качестве базового типа можно использовать следующие типы: INTEGER, BYTE, CHAR, перечислимый и ограниченный.

При определении множественных типов существует два ограничения:

- 1) вещественный тип в качестве базового в множествах использовать нельзя;
- 2) число элементов в множестве определяется каждой конкретной реализацией ЭВМ. Обычно число элементов колеблется между 64 и 256 (для Турбо - Паскаля - 256). Такая зависимость приводит к потере переносимости программ, обладающих этим типом, с машины на машину.

Множества объявляются либо в разделе описания переменных VAR, либо в разделе описании типов TYPE. Объявление множества в разделе описания переменных имеет вид:

VAR

< имя множества > : SET OF < базовый тип >;

Например:

Var

god : set of 1900..2000;
symbol : set of char;

Объявление множества с использованием раздела описания типов имеет вид:

Type

< имя типа > = set of < базовый тип >;

Var

< имя множества > : имя типа;

Например:

Type

god = set of 1900..2000;
symbol = ('A'..'Z');

Var

```
g : god;
s : set of symbol;
```

Значения переменных и констант множества задаются в разделе операторов с помощью конструктора. Конструктор представляет собой список элементов базового типа, заключенный в квадратные скобки, который затем можно присвоить переменной, или обработать.

Конструктор множества можно рассматривать как константу типа множества.

Например:

```
figura := [romb];
```

или

```
figura := [krug,romb,kvadrat];
```

```
simv := ['A','B','C'];
```

```
M1 := [1,3,5,10];
```

```
M2 := []; { пустое множество }
```

9.2. Операции над множествами

В языке Турбо-Паскаль имеются следующие группы операций над множествами:

- 1) объединение, пересечение, вычитание множеств;
- 2) проверка принадлежности элемента множеству;
- 3) проверка на равенство и неравенство множеств;
- 4) проверка на принадлежность одного множества другому.

Операции объединения, пересечения и вычитания являются традиционными действиями над множествами и обозначаются символами '+', '*', '-' соответственно. **Например:**

```
[1,2] + [3,4] = [1,2,3,4];
[1..10] + [5..15] = [1..15];
[1..10] * [5..15] = [5..10];
[1,2] * [3,4] = [];
[1..10] - [5..15] = [1..4];
```

Проверка принадлежности множеству - это логическая операция, которая обозначается служебным словом IN. Правый операнд должен быть множеством, левый - значением базового типа множества. Операция возвращает TRUE, если значение входит в множество, и FALSE в противном случае. **Например:**

```
2 in [1..10,12]; { имеет значение true}
5 in [1,2,7,10]; { имеет значение false}
```

Операцию проверки принадлежности удобно использовать для исключения более сложных проверок, например, оператор вида

```
if (symb = 'a') or (symb = 'b') or (symb = 'x') or (symb = 'y') then s;
```

может быть переписан в более компактной форме

```
if symb in ['a','b','x','y'] then s;
```

Второй вариант эффективен с точки зрения быстродействия.

Проверка на равенство, неравенство и включение множеств - это бинарные логические операции, которые обозначаются следующими символами:

= равенство (совпадение) двух множеств;

<> неравенство множеств;

<= проверка на вхождение множества из левого операнда в множество из правого операнда;

>= проверка на вхождение множества из правого операнда в множество из левого операнда.

Все эти операции вырабатывают логическое значение TRUE или FALSE в зависимости от успеха проверки. **Например:**

```
[1,2,3] = [1,2] - false;
[1,2,3] >= [1,2] - true;
[S] <= [1..10] - true,
если S - целое число из диапазона 1..10;
[1,2,3] <> [1,2,2] - true.
```

Синонимом логической операции над множествами является слово "компаратор".
Набор операций над множествами в языке Турбо-Паскаль не содержит одной практически важной операции - выборки значений из множества (или близко связанного с ней средства циклического перебора значений множества). Поэтому при необходимости подобных действий приходится организовывать цикл по всему диапазону значений базового типа, проверяя на каждой итерации принадлежность очередного значения данному множеству, **например**:

```
Var
    symbols : set of char;
    s : char;
Begin
    . . .
    For s := chr(0) to chr(255) do
        if s in symbols then
            < действия с переменной s >
    . . .
```

Контрольные вопросы

1. Что понимается под множеством?
2. Какие вы знаете операции над множествами в математике?
3. Как записываются операции над множествами в языке Турбо-Паскаль?
4. Как задаются множества на языке Турбо-Паскаль?
5. Что такое пустое множество и как оно задается?
6. Как организовать вывод элементов множества?

Задание к работе

1. Выполнить задание А.
2. Выполнить задание Б.

Методические указания

1. При выполнении индивидуального задания А необходимо:
 - а) ознакомиться с конечным и упорядоченным множеством символов, определенным на используемой для выполнения задания ЭВМ;
 - б) составить программу для конкретного варианта, работающую для произвольного набора символов.
 - в) входная строка символов может быть длиннее строки экрана терминала, при этом программа работает не с функцией EOLN, а с признаком конца строки, который задается программистом.
2. При выполнении индивидуального задания Б необходимо учесть приемы программирования, использованные в приведенной ниже программе ASMAG.
Известен набор продуктов - хлеб, масло, сыр, молоко, имеющих в ассортименте магазинов. В три магазина доставлены отдельные виды этих продуктов. Требуется построить множества А, В, С, которые содержат соответственно:
 - продукты, имеющиеся одновременно во всех магазинах;
 - продукты, имеющиеся по крайней мере в одном из магазинов;
 - продукты, которых нет ни в одном из магазинов.

```
Program ASMAG;
Const N=3;
Type
    product=(bread,butter,cheese,milk); {задается список объектов (продуктов),
определяющий базовый тип PRODUCT}
    assort = set of product; {на базовом типе PRODUCT определя-ется множественный тип
ASSORT}
```

```

    magazin = array [1..N] of assort;      {информация о наличии продуктов во всех
магазинах задается как массив множеств}
Var
    m1 : magazin; x : product;
    a,b,c, xm1 : assort;
    i,j,iw,m : integer;
Begin
    for i := 1 to N do    {ввод исходной информации}
        begin
            xm1 := [];
            writeln (' введите номера продуктов',i : '-го магазина =');
            repeat    {в цикле REPEAT формируется множество XM1,
                характеризующее наличие товаров в одном магазине.}
            read(iw);
            case iw of
                1: x := bread;
                2: x := butter;
                3: x := cheese;
                4: x := milk
            end;
            xm1 := xm1 + [x];
            until eoln;
            m1[i] := xm; {информация о наличии товаров записывается в массив M1}
            end;
            for i := 1 to 3 do {формирование множеств A,B,C и их распечатка}
                begin
                    case i of
                        1: writeln('продукты, имеющиеся одновременно во
                            всех магазинах');
                        2: writeln('ассортимент продуктов');
                        3: writeln('продукты, которых нет ни в одном магазине')
                    end;
                    for x := bread to milk do
                        if x IN a then
                            case x of
                                bread: write('хлеб');
                                butter: write('масло');
                                cheese: write('сыр');
                                milk: write('молоко')
                            end;
                        end;
                    if i = 1 then
                        a := b
                    else
                        a := c;
                    writeln
                end
            end.

```

Содержание отчета

1. Титульный лист.
2. Словесная постановка задачи.
3. Графический или текстуальный алгоритм решения задачи.

4. Листинг программы.
5. Контрольный тест и результаты тестирования программы.
6. Инструкция по эксплуатации программы.
7. Ответы на контрольные вопросы.

Варианты индивидуальных заданий

Задание А

Дана непустая последовательность символов. Требуется построить и напечатать множество, элементами которого являются символы, встречающиеся в последовательности индивидуального варианта:

Вариант	Последовательность символов
1	Буквы от 'A' до 'F' и от 'X' до 'Z';
2	Цифры от '5' до '9' и знаки арифметических операций;
4	Буквы от 'T' до 'X' и цифры от '1' до '4';
5	Знаки препинания и операций отношения;
6	Знаки арифметических операций и буквы от 'E' до 'N';
7	Буквы от 'A' до 'Z' и знаки препинания;
8	Знаки операций отношения;
9	Цифры от '0' до '9';
10	Знаки арифметических операций и операций отношения.

Задание Б

1. Даны два конечных множества A и B , элементами которых могут быть любые целые числа в диапазоне от 1 до 30. Найти прямое произведение этих множеств и вывести его на экран.
2. Даны два прямоугольника. Множества A и B - это множества точек, принадлежащих соответствующим прямоугольникам. Координаты точек - это натуральные числа от 1 до 10. Определить пересекаются ли данные прямоугольники, если пересекаются, то вывести на экран их общие точки.
3. Даны два конечных множества X и Y , состоящие из целых чисел. Определить выполняется ли равенство: $(A \cup B) \setminus B = A$.
4. Даны два конечных множества X и Y , состоящие из целых чисел. Определить выполняется ли равенство: $(A \setminus B) \cup (B \setminus A) = (A \cup B) \setminus (A \cap B)$.
5. Пусть A, B, C - конечные множества, такие что $B \subseteq A \subseteq C$. Найдите множество X , удовлетворяющее условиям $A \cap X = B$ и $A \cup X = C$.
6. Пусть A, B, C - конечные множества, такие что $B \subseteq A, A \cap C = \emptyset$. Найдите множество X , удовлетворяющее условиям $A \setminus X = B$ и $X \setminus A = C$.
7. Даны следующие множества $A = \{1, 2, 3\}, B = \{2, 3, 5, 4\}, U = \{0, 1, 2, 3, \dots, 9\}$. Найти и вывести на экран $A \cup B, A \setminus B, B \setminus A, U \setminus A$.
8. Даны два конечных множества A и B , состоящие из целых чисел. Найти и вывести на экран $(A \cup B) \setminus (A \cap B)$.
9. Даны два множества $A = \{1, 2\}, B = \{3, 4, 5\}$. Выведите на экран элементы множеств $A \times B, B \times A$.
10. Пусть $A = \{b, o\}$. Перечислите элементы множеств A^3 и A^4 .