

## ЛАБАРАТОРНАЯ РАБОТА № 9

### Праграмаванне з выкарыстаннем мностваў

**Мэта:** вывучыць правілы апісання даных тыпу мноствы, дапушчальныя ў Паскалі аперацыі над мноствамі, а таксама спосабы складання праграм для апрацоўкі даных тыпу мноствы.

Мноства – гэта некаторы канечны набор ці сукупнасць аднатыпных элементаў цэлага (і яго раздзелаў), сімвальнага, лагічнага, пералічывага тыпаў, якія з’яўляюцца базавымі для пабудавання мностваў. Прыкладамі мностваў з’яўляюцца сукупнасць цэлых лікаў і сукупнасць літар лацінскага алфавіта.

Для апісання мноства выкарыстоўваюцца:

1) квадратныя дужкі. Заклучэнне элемента ў квадратныя дужкі можна разглядаць як аперацыю стварэння множнага значэння:  $3$  – проста лік цэлага тыпу, а  $[3]$  – элемент мноства;

2) спалучэнне службовых слоў

*set of: type x:=set of (1,2,3);*

*var a: set of 'a'..'z';*

Прыклады мностваў:

$[\ ]$  – пустое мноства;

$[1, 2, 6, 8]$  – мноства зададзеных цэлых лікаў;

$[2, b]$  – мноства, якое складаецца з цэлага ліку  $2$  і бягучага значэння цэлалікавай пераменнай  $b$  (пры  $b=2$  мноства складаецца з аднаго элемента);

$['a','b','c']$  – мноства зададзеных літар;

$['a',... 'z']$  – мноства ўсіх малых лацінскіх літар;

$[10..99]$  – мноства двухзначных лікаў.

Калі ў дыяпазоне  $c..x$ , які выкарыстоўваецца для задання множнай канстанты,  $c = x$ , то гэты дыяпазон дапускае толькі адзін элемент  $c$ . Калі  $c > x$ , то гэты дыяпазон не пастаўляе ў мноства ніводнага элемента:

$[2..2]=[2]$   $[9..3]=[\ ]$  – пустое.

Над мноствамі дапускаюцца наступныя аперацыі (у парадку ўбывання):

1. Аперацыя "роўна". *TRUE* ці *FALSE*. Вынік аперацыі "праўдзіва", калі мноствы складаюцца з адных і тых жа элементаў.

2. Аперацыя "не роўна".

3. Аперацыя вызначэння прыналежнасці элемента некатораму мноству.

Абазначаецца службовым словам *in*. Злева ад *in* – элемент, а справа – мноства:

$2 \text{ in } [2,5]$  – true;

$3 \text{ in } [7..25]$  – false;

'a' in ['t','j','a'] – true.

4. Аб'яднанне мностваў (+):  $C := A + B$ , дзе мноства  $C$  змяшчае элементы, якія належаць або  $A$ , або  $B$ , або  $A$  і  $B$  адначасова:

$[1,2,3,4,5] + [2,5,6,7,8] = [1,2,3,4,5,6,7,8]$ .

5) Аперацыя "менш альбо роўна"  $A \leq B$ : true, калі ўсе элементы мноства  $A$  належаць мноству  $B$ .

6) Перасячэнне мностваў (\*):  $C := A * B$ , дзе мноства  $C$  змяшчае элементы, якія належаць  $A$  і  $B$  адначасова:

$[1,2,3] * [7,2] = [2]$   $[1,2,3,4,5] * [2,5,6,7,8] = [2,5]$ .

7) Рознасць мностваў:  $C := A \setminus B$ , дзе мноства  $C$  змяшчае элементы, якія прыналежаць  $A$  і не прыналежаць  $B$ .

$[1,2,3] - [1,2] = 3$   $[1,2,3,4,5] - [2,5,6,7,8] = [1,3,4]$ .

У адрозненне ад элементаў масіву элементы мноства неўпарадкаваны, таму наступныя мноствы аднолькавыя:  $[1,2,3,4,5]$ ,  $[4,3,2,1]$ ,  $[4,2,1,3]$  і да т. п.

Пад магутнасцю мноства разумеецца агульны лік яго элементаў. Магутнасць пустога мноства роўна 0.

**Прыклад 1.** Дадзены тры мноствы:  $X=[1,2,4]$ ,  $Y=[3,4,5,6]$ ,  $Z=[1,6]$ . Сфарміруем новае мноства  $C=(X \cup Y) \cap Z$  і надрукуем яго элементы. Праверым выкананне ўмовы "Z належыць C":

```

program m1;
  var x,y,z,c: set of 1..6; k:1..6;
BEGIN
  x:=[1,2,4]; y:=[3,4,5,6]; z:=[1,6]; c:=(x+y)*z;
  writeln('Мноства C');
  for k:=1 to 6 do
    if k in c then write(k:4);
  writeln;
  if z<=c then writeln('z належыць c')
    else writeln('z не належыць c');
END.

```

**Прыклад 2.** Сфарміруем мноства двухзначных лікаў. Вылучым з яго падмноства лікаў, кратных 3 і 5 і выведзем яго на экран:

```

program m1;
  var M,C:set of 10..99; i:10..99;
BEGIN
  M:=[];
  for i:=10 to 99 do M:=M+[i];
  C:=[];

```

```

for i:=10 to 99 do
    if (i mod 3=0) and (i mod 5=0) then C:=C+[i];
for i:=10 to 99 do
    if i in C then writeln(i);

```

END.

**Прыклад 3.** Выведзем на экран ўсе простыя натуральныя лікі ад 2 да зададзенага ліку  $N$ .

Пры рашэнні гэтай задачы будзем выкарыстоўваць алгарытм, які завецца "рэшата Эратасфена": у пачатку запісваюцца ўсе натуральныя лікі ад 2 да  $n$ . Потым памячаецца лік 2 (на паперы можна абвесці кружочкам) і выкрэсліваюцца ўсе астатнія лікі, кратныя 2, з выкрэсленымі лікамі больш не працуюць. Далей бярэцца першы не закрэслены лік – гэта 3, таксама памячаецца, і выкрэсліваюцца ўсе астатнія лікі, кратныя 3. Працэс паўтараецца да таго часу, пакуль не застанецца не закрэсленых лікаў. Усе памечаныя лікі і складаюць мноства простых лікаў у зададзеным прамежку:

```

program prostye;
const n=100;
type mno=set of 2..n;
var x,y:mno; p,k:2..n;
BEGIN
x:=[2..n]; y=[]; {у у будучь фарміравацца простыя лікі}
p:=2;
repeat
    while not (p in x) do p:=p+1; {вызначаецца новы незакрэслены лік}
    y:=y+[p]; {і заносіцца ва мноства простых}
    k:=p;
    repeat
        x:=x-[k]; {выдаляюцца з X кратныя лікі}
        k:=k+p;
    until k>n;
until x=[];
{ вывад выніку на экран}
for k:=2 to n do
    if k in y then write(k:4);

```

END.

**Прыклад 3.** Вядомы набор тавараў:

- 1) запалкі,
- 2) мыла,
- 3) ніткі,

4) сшыткі,  
якія ёсць у асартыменце крам. У тры крамы дастаўлены асобныя віды гэтых прадуктаў. Пабудуем мноствы  $A$ ,  $B$ ,  $C$ , якія змяшчаюць адпаведна:

- тавары, якія ёсць адначасова ва ўсіх крамах;
- тавары, якія ёсць прынамсі ў адной з крам;
- тавары, якіх няма ў ніводнай краме;

```

program trgov;
  const n=3;
  type tovar=(spich,mylo,nitki,tetradi);
  assort=set of tovar;
  magazin=array[1..n] of assort;
  var m:magazin; x:tovar; a,b,c,xm:assort; i,j,k,d:integer;
BEGIN
  for i:=1 to n do
    begin
      xm=[];
      writeln ('увядзіце нумары прадуктаў ',i,'-й крамы);
      repeat
        read(k);
        case k of
          1:x:=spich;
          2:x:=mylo;
          3:x:=nitki;
          4:x:=tetradi;
        end;
        xm:=xm+[x];
      until eoln; {накуль не націснута клавiша ўводу}
      m[i]:=xm;
    end;
  a:=m[1];b=[]; c:=[spich..tetradi];
  for i:=1 to 3 do
    begin
      b:=b+m[i]; a:=a*m[i]; c:=c-b
    end;
  for i:=1 to 3 do
    begin
      case i of
        1:writeln('Тавары, якія ёсць адначасова ва ўсіх крамах);
        2:writeln('Тавары, якія ёсць прынамсі ў адной з крам);

```

```
      3:writeln('Тавары, якіх няма ў ніводнай краме);  
end;  
for x:=spich to tetradi do  
  if x in A then case x of  
    spich:write(' запалкі ');  
    mylo: write(' мыла ');  
    nitki: write(' ніткі ');  
    tetradi:write(' сшыткі ');  
  end;  
  if i=1 then a:=b else a:=c; writeln;  
end;  
END.
```