*Практичні заняття*

**Множини**

**Paskal** дозволяє оперувати трьома множинами, як трьома типами даних. Для визначення типу множина використовується вираз:

**set of** простий тип

1. Наприклад, описання виду:

type

Char Set = set of ‘A’.. ‘Z’

Визначає тип множина, значеннями якого є множини символів – букв, а елементами множини – символи – латинські букви від А до Z.

2) Описання виду

type

Number Set = set of 0..50 визначає тип множина, а членами множини – цілі числа, які знаходяться в межах від 0 до 50.

1. Порожня множина є елементом всіх типів множин.
2. Приклади описів типів множина:

type

Symbol Set = set of ‘ ‘..’ ‘;

Colour = (WHITE, BLUE, RED);

Colour Set = set of Colour;

T1 = set of 0..9

Var

C: colour; Col Set: Colour Set;

T: inteper;

### TSet: T1

В даному випадку значенням змінної Т може бути будь-яка цифра від 0 до 9, а значенням змінної TSet – довільна сукупність цифр від 0 до9.

1. Над множинами в Р допустимі 4 операції;

* ***oб’єднання*** (“+”) Об’єднання множин – це множина, яка містить усі елементи цих множин без повторень.
* ***перетин*** (“ \* ”) Перетин множин – це множина, яка складається з елементів, які є спільними для всіх множин.
* ***різниця*** (“ - ”) Різницею множин А і В є множина, яка складається з елементів, що є в А, але не є в В.
* ***операція*** in.

Операція in дозволяє визначити чи належить елемент множині, чи ні. Першим операндом, розміщеним зліва від слова in, є вираз базового типу (тобто типу, якому повинні належати всі члени множини). Другий операнд, який знаходиться справа in, повинен мати тип множина.

**Наприклад:** Red in [RED, WHITE] – результат true

8 in [0..3, 6, 9] – результат false.

7) В Р. програмі множина задається в вигляді списку елементів, заключеного в [ ]. В [ ] може бути 1 або більше елементів, а може не бути жодного (порожня множина). В якості елементу може використовуватись const, змінна, вираз, значення якого належить базовому типу, а також парі елементів, розділених двома крапками (інтервал значень).

8) В Р. можна використовувати інструкції присвоєння слідуючих виразів:

ColSet : = [WHITE, RED];

ColSet : = [ ];

TSet : = [1, 7, 5];

TSet : = [1..5, 8];

TSet : = [8 mod 4, 15 div 5].

9) При роботі з множинами можна використовувати операція порівняння:

**=, < >, > =, < =**

Операції **“=”** і **“< >”** дозволяють перевірити, рівні дві множини, чи ні. З допомогою oперацій **“> =”** і **“< =”** можна визначити, чи є одна множина підмножинною іншої.

**Приклад:**

[RED, WHITE] = [ RED, GREEN] – резкльтат false

[RED] < = [RED, WHITE] – результат true.

Операції в порядку зменшення пріоритету розміщуються так:

**\***

**+**

**in, =, < >, > =, < =** (рівнопріоритетні операції)

**Приклад №1** Із файла *Input* вводиться текст, який містить символи від знаку “+” до лівої квадратної дужки “ [ “. Роздрукувати символи тексту в порядку коду *ASCII* (з символів, що зустрічаються повторно, виводити тільки один).

Program Sort (Input, Output);

Var

S: char;

Sets: set of ‘+’ [‘;

I: ‘+’..’[‘;

begin

Sets: = [ ];

Read (S)

While not Eof do begin

While not Eoln do

begin

Sets: = Sets + [S];

Read (S)

end

Readln

End

for I: = ‘+’ to ‘[‘ do

if I in Sets

then Write (I) else; writeln end.

**Приклад №2** Написати програму, яка друкує всі прості числа з відрізку 2..N діючи по методу “решета Ератосорена”

# ***“Решето Ератосорена”***

Program Rach;

Coust

N = 15

Var

S: set of 2..N

{початкова множина чисел}

i, k: integer;

begin

S: = [2..N];

for i: = 2 to N do

if i in Sthen begin

writeln (i);

{виводимо найменше із елементів S}

{забираємо із S числа, крайні і}

for k: = 1 to N div i do

S: = S – [k\*i];

end {if }

end.

## **Внутрішнє представлення множин**

Знайомство з внутрішнім представленням множин допоможе нам зрозуміти особливості і обмеження, властиві цьому типові даних.

Всі значення множини представляються в пам’яті послідовностями бітів однакової довжини. За кожне значення базового типу “відповідає” один біт. Якщо множина вміщує деякий елемент, в “відповідальному” за нього біті зберігається 1, якщо не вміщує – зберігається 0.

**Приклад.**

Var X: set of 1..15;

Внутрішнє представлення Х

X: = [ ]; 000.0.000.0.000.0.000 >.

011010000000000 >

X: = [2, 3,5];

X: = [1..15]; 111111111111111 >

Операції над множинами зводяться до “поразрядныx“ логічних операцій над послідовністю бітів, наприклад об’єднання множин використовується шляхом “поразрядного” логічного додавання бітів.

X: = [2, 3, 5]; 011010000000000 >

Y: = [3, 5, 7, 8]; 0010101.10000000 >

Z: = X+Y; 01101.0110000000 >

“Поразрядные” операції входять в набір команд процесора ЕОМ, тому виконуються швидко.