



Тип даних множина

*Мельник О.О.
доцент кафедри
програмного забезпечення систем*



План:

1. Множина. Опис типу.
2. Дії над множинами.
3. Приклади використання типу.
4. Переваги та недоліки використання типу.



Множина. Опис типу.

Множина – це структурований тип даних, котрий являє собою набір взаємопов'язаних за певною ознакою або групою ознак об'єктів, які можна розглядати як єдине ціле.

Кожен об'єкт множини називається *елементом множини*.

Усі елементи множини мають належати одному із скалярних типів, крім дійсного. Цей тип називається *базовим типом множини*. Базовий тип задається діапазоном або переліченням.



Множина. Опис типу.

Кількість елементів множини не повинна перевищувати **256**, отже номери значень базового типу повинні належати діапазону 0..255 .

Контроль діапазонів здійснюється включенням директиви компілятора **{ \$R+ }**.

Один елемент множини займає пам'ять **1 біт**. Об'єм пам'яті для змінної типу множина обчислюється за формулою:

$$\text{Об'єм пам'яті} = (\text{Max div } 8) - (\text{Min div } 8) + 1,$$

де Max і Min – верхня і нижня межі базового типу.



Множина. Опис типу.

Область значень типу множина – набір підмножин, складених з елементів базової множини.

У виразах на мові Паскаль значення елементів множини вказуються в квадратних дужках.

Наприклад:

[1, 2, 3, 10]

['a', 'e', 'o']

['a'..'z']

Якщо множина не має елементів, вона називається *порожньою* і позначається [].



Множина. Опис типу.

Формат опису:

Type

<назва типу> = **set of** <елемент1>, ... , <елементN>;

Var

<ідентифікатор, ...> : <назва типу>;

Або:

Var

<ідентифікатор, ...> : **set of** <елемент1>, ... , <елементN>;



Множина. Опис типу.

Приклади:

Type

Proste = set of (3, 5, 7, 11, 13);

Nomer = set of 1..25;

Var

c: Proste;

N: Nomer;

Litera: set of ['a', 'e', 'u', 'o', 'i'];



Дії над множинами

Позначення	Назва	Результат
$A = B$	рівне	<p>Дві множини вважається рівними, якщо вони складаються з одних і тих самих елементів. Порядок запису елементів у множинах, котрі порівнюються, значення не має.</p> <p><i>Тип результату: Boolean</i></p>
$A \neq B$	не рівне	<p>Дві множини вважається не рівними, якщо вони відрізняються потужністю або значенням принаймні одного елемента.</p> <p><i>Тип результату: Boolean</i></p>



Дії над множинами

Позначення	Назва	Результат
$A \geq B$	більше або рівне	<i>True</i> , якщо всі елементи множини B належать множині A . <i>Тип результату: Boolean</i>
$A \leq B$	менше або рівне	<i>True</i> , якщо всі елементи множини A належать множині B . <i>Тип результату: Boolean</i>
$a \in B$	належить	Перевірка належності певного значення множині. <i>Тип результату: Boolean</i>



Дії над множинами

Позначення	Назва	Результат
$A + B$	об'єднання	Множина, котра містить елементи множин A і B . <i>Тип результату: множина</i>
$A * B$	перетин	Множина, котра містить елементи, які входять і в множину A , і в множину B . <i>Тип результату: множина</i>
$A - B$	різниця	Множина, котра містить елементи, які входять у множину A , і не входять у множину B . <i>Тип результату: множина</i>



Приклади використання типу

Значення А	Значення В	Вираз	Результат
[1, 2, 3, 4]	[1, 2, 3, 4]	$A = B$	True
['a', 'b', 'c']	['c', 'a']	$A = B$	False
['a' .. 'z']	['z' .. 'a']	$A = B$	True
[1, 2, 3]	[3, 1, 2, 4]	$A \triangleleft B$	True
['a', 'b', 'c']	['c', 'a', 'b']	$A \triangleleft B$	True
['a' .. 'z']	['z' .. 'b']	$A \triangleleft B$	True



Приклади використання типу

Значення A	Значення B	Вираз	Результат
[1, 2, 3, 4]	[2, 3, 4]	$A \geq B$	True
['a', 'b', 'c']	['c', 'a']	$A \geq B$	True
['a' .. 'z']	['b' .. 't']	$A \geq B$	True
2	[3, 1, 2, 4]	$a \text{ in } B$	True
X2	[X1, X3, X5]	$a \text{ in } B$	True
'v'	['n' .. 'b']	$a \text{ in } B$	False



Приклади використання типу

Значення А	Значення В	Вираз	Результат
[1, 2, 3]	[1, 4, 5]	$A + B$	[1, 2, 3, 4, 5]
['a' .. 'd']	['e' .. 'z']	$A + B$	['a' .. 'z']
[]	[]	$A + B$	[]
[1, 2, 3]	[5, 2, 4, 1]	$A * B$	[1, 2]
['b' .. 'z']	['a' .. 'r']	$A * B$	['b' .. 'r']
[1, 2, 3, 4]	[3, 2, 7]	$A - B$	[1, 4]
['a' .. 'z']	['d' .. 'z']	$A - B$	['a' .. 'c']



Приклади використання типу

Операція **in** дозволяє ефективно і наглядно проводити складні перевірки умов, замінюючи велику кількість інших операцій.

Наприклад:

IF (a=1) **or** (a=2) **or** (a=3) **or** (a=4) **or** (a=5) **THEN** ...

IF a **in** [1..5] **THEN** ...



Приклади використання типу

Використання операції **in** у запереченнях:

X not in M

- помилковий запис;

not (X in M)

- правильна конструкція.



Приклади використання типу

Задача. У заданому тексті обчислити кількість латинських літер.

Program Strings12;

Uses Crt;

Type

CharSet = **set of** Char;

Const

Letters : CharSet = ['a'..'z', 'A'..'Z'];

Var

St, St1: string;

n, k: word;



Приклади використання типу

BEGIN

```
clrscr;  
write (' Enter text: '); readln (St);  
n:=length (St);  
k:=0;  
For i:=1 to n do  
begin  
    St1:= Copy (St,i,1);  
    IF St1 in Letters THEN inc (k);  
end;  
Writeln (' Letters counter =', k);  
repeat until keypressed;  
END.
```



Переваги та недоліки використання типу

Переваги:

- спрощуються умови складених операторів IF/THEN;
- покращується наочність програми, розуміння алгоритму розв'язування задачі;
- економиться пам'ять, час компіляції і виконання.

Недоліки:

відсутні стандартні засоби введення/виведення елементів множини, програміст повинен створювати їх сам.



Рекомендована література:

1. Turbo Pascal 7.0: Пер. с нем. / Винфрид и Фолькер Кассера. – К.: Издательство «ДиаСофт», 2003.
2. Грызлов В.И., Грызлова Т.П. Турбо Паскаль 7.0. – 4-е изд., испр. – М.: ДМК Пресс, 2005.
3. Паскаль для персональных компьютеров: Справ. Пособие / Ю. С. Бородич, А.Н. Вальвачев, А.И. Кузьмич. – Мн.: Выш. Шк.: БФ ГИТМП «Ника», 1991.
4. Паскаль. Программирование на языке высокого уровня: Учебник для вузов / Т.А. Павловская. – СПб: Питер, 2003.
5. Программирование в среде Turbo Pascal 7.0 / Марченко А.И., Марченко Л.А.: Под ред.: Тарасенко В.П. – 8-е изд. – К.: ВЕК+, СПб: КОРОНА принт, 2004.
6. Зеленьяк О.П. Практикум программирования на Turbo Pascal. Задачи, алгоритмы и решения. – СПб.: ООО «ДиаСофтЮП», 2002.