19.12.20г\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

19ИСиП 1Д \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Основы алгоритмизации и программирования

ТЕМА: П\З **Работа с данными типа множество**

**В Паскале под множественным типом понимается конечная совокупность** элементов, принадлежащих некоторому ***базовому типу*** данных.

В качестве базовых типов могут использоваться:

* перечислимые типы;
* символьный;
* байтовый;
* диапазонные на основе вышеперечисленных.

Такие ограничения связаны с формой представления множественного типа данных в Паскале и могут быть сведены к тому, чтобы функция **ord()** для используемого базового типа лежала в пределах от 0 до 255.

После того, как базовый тип задан, совокупность значений соответствующего множественного типа данных определяется автоматически. В нее входят все возможные множества, являющиеся произвольными комбинациями значений базового типа. Все эти множества являются отдельными значениями определенного множественного типа данных.

Описание множественного типа данных Паскаля

**Type** <имя\_типа>= **set** **of** <базовый\_тип>

**Пример множественного типа данных Паскаля**

**Type** symbol= **set** **of** char; {описан множественный тип symol из букв}
**Var** letter, digits, sign: symbol; {описаны три переменные множественного типа}

Для того чтобы придать переменной множественного типа значение, используют **конструктор множества** – перечисление элементов множества через запятую в квадратных скобках. Например,

sign:= ['+', '-'];

Конструктор множества может содержать диапазон значений базового типа. Тогда во множества включаются все элементы диапазона. Например,

digits:= ['0' .. '9'];
letter:= ['a' .. 'z'];

Обе формы конструирования множеств могут сочетаться. Например,

letter:= ['a' .. 'z', 'A' .. 'Z'];

Конструктор вида [] обозначает пустые множества.

В программе можно использовать множественны тип как константы, в этом случае их определяют следующим способом:

**Const** YesOrNo= ['Y', 'y', 'N', 'n'];

Можно множественный тип определить как типизированную константу:

Const digits: **set** **of** char= ['0' .. '9'];

При описании множественного тип как констант допускается использование знака “+” (слияние множеств). Например,

**Const** Yes= ['Y', 'y']; No= ['N', 'n'];
YesOrNo= Yes+ No;

Операции над множественными типами Паскаля

С множественными типами Паскаля можно выполнять действия объединения, исключения и пересечения.

**Объединение множественных типов** содержит элементы, которые принадлежат хотя бы одному множеству, при этом каждый элемент входит в объединение только один раз. Операция объединения множеств обозначается знаком '+'.

**Пример множественных типов Паскаля**

**Type** symbol= **set** **of** char;
**Var** small, capital, latin: symbol;
………………
small:= ['a' .. 'z'];
capital:= ['A' .. 'Z'];
latin:= small + capital; {образованы множества латинских букв путем объединения множеств small и capital}

Возможно объединять множественные типы и отдельные элементы. Например,

small:= ['c' .. 'z'];
small:= small + ['a'] +['b'];

***Исключение*** определяется как разность множественных типов, в котором из уменьшаемого исключаются элементы, входящие в вычитаемое. Если в вычитаемом есть элементы, отсутствующие в уменьшаемом, то они никак не влияют на результат. Операция исключения обозначается знаком '-'.

**Пример исключения множественных типов Паскаля**

letter:= ['a' .. 'z']; {множества букв латинского алфавита}
glasn:= ['a', 'e', 'o', 'u', 'i', 'y']; {множества гласных букв}
soglasn:= letter-glasn; {образовано множества согласных букв путем исключения из множества всех букв множества гласных букв}

**Пресечение множественных типов**– множества, содержащие элементы, одновременно входящие в оба множества. Операция пересечения множеств обозначается знаком '\*'.

**Пример пересечения множественных типов**

**Type** chisla= **set** **of** byte;
**Var** z,x,y: chisla;
………..
x:= [0..150];
y:= [100..255];
z:= x\*y {получено множества чисел из диапазона 100..150 в результате пересечения двух множеств}

Операции отношения множественных типов Паскаля

Наряду с рассмотренными выше операциями, над значениями множественного типа определены и некоторые операции отношения. Операндами операций над множественными значениями в общем случае являются множественные выражения. Среди операций отношения над значениями множественного типа особое место занимает специальная **операция проверки вхождения элемента во множества**, обозначаемая служебным словом **in**. В отличие от остальных операций отношения, в которых значения обоих операндов относятся к одному и тому же множественному типу значений, в операции in первый операнд должен принадлежать базовому типу, а второй – множественному типу значений, построенному на основе этого базового типа. Результатом операции отношения, как обычно, является логическое значение (true или false).

'a' **in** glasn значение операции true;
'o' **in** soglasn значение операции false;

**Операция сравнения на равенство множественных типов Паскаля**. Множества считаются равными (эквивалентными), если все элементы одного множества присутствуют в другом и наоборот. Для операции сравнения на равенство или неравенство используются символы '=' и '<>'.

A:= [2,1,3];
D:= [1,3,2];

Тогда операция A=D имеет значение true, а операция A<>D имеет значение false.

***Проверка включения***. Одно множество считается включенным в другое (одно множество является подмножеством другого), если все его элементы содержатся во втором множестве. Обратное утверждение может быть и несправедливым. Операции проверки включения обозначаются '<=' и '>='.

**letter** >= glasn;
**soglan** <= letter;

Следует отметить, что применение операций < и > над операндами множественного типа недопустимо.

Преподователь \_\_\_\_\_\_\_\_\_\_\_\_\_\_Дузаев И.К.