

## ЛАБАРАТОРНАЯ РАБОТА № 2

### Праграмаванне вылічальных працэсаў галінаванай структуры

**Мэта:** набыць практычныя навыкі запісу галінаваных алгарытмаў на мове Паскаль.

**Умоўны аператар** дазваляе ў час выканання праграмы выбраць адно з двух дзеянняў. Для рэалізацыі галінаванай структуры існуюць два віды ўмоўнага аператара:

1. *if u then d1;*
2. *if u then d1 else d2;*

Тут *u* – умова, якая з’яўляецца выразам булеўскага тыпу; *d1*, *d2* – асобныя аператары ці аператары, сгрупаваныя разам пры дапамозе аператарных дужак *begin end* (такі аператар называюць састаўным).

**Прыклад 1.** Рашым ураўненне віду  $ax = b$ .

Для рашэння гэтага, на першы погляд, простага ўраўнення, трэба разгледзіць наступныя выпадкі:

1. Калі *a* не роўна нулю, то рашэнне ўраўнення вызначаецца пасля дзялення ліку *b* на лік *a*.
2. Калі *a* роўна нулю і *b* роўна нулю, то ўраўненне прыме выгляд  $0x = 0$ , а гэта азначае, што рашэннем яго можа быць любы лік.
3. Калі *a* роўна нулю, а *b* не роўна нулю, то ўраўненне  $0x = b$  не мае рашэння.

Варыянт праграмы 1.

```
program urav;  
  var a,b,x:real;  
BEGIN  
  writeln('уведзіце значэнні a і b');  
  readln(a,b);  
  if a<>0 then begin  
    x:=b/a;  
    writeln('адказ ',x:5:2);  
  end  
  else  
    if (b=0) then writeln('любы лік')  
    else writeln('няма рашэнняў');  
END.
```

Варыянт праграмы 2.

```

program urav;
  var a,b,x:real;
BEGIN
  writeln('уведзіце значэнні a і в');
  readln(a,b);
  if a<>0 then begin
    x:=b/a;
    writeln('адказ ',x:5:2);
  end;
  if (a=0) and (b=0) then writeln('любы лік');
  if (a=0) and (b<>0) then writeln('няма рашэнняў');
END.

```

Калі пры выкананні праграмы неабходна зрабіць многа ўзаемавыключających праверак, то зручней скарыстацца аператарам **выбару варыянта**:

```

case варыянт of
  спіс метак: аператар1;
  .....
  спіс метак: аператар n;
end;

```

альбо

```

case варыянт of
  спіс метак: аператар1;
  .....
  спіс метак: аператар n
  else аператар n+1;
end;

```

Прыклады:

```

var a,b,k:integer;
sezon:(zima, vesna, leto, osen);
s:char;
.....
case k+2 of
  1,3,5:a:=k+22;
  2,4,6:a:=k+32;
  9,12,16:a:=k+42
  else a:=k+1
end;
.....

```

```
case k of
  1..10:writeln('k знаходзіцца ў дыяпазоне 1-10');
  11..20:writeln('k знаходзіцца ў дыяпазоне 11-20')
    else writeln('k знаходзіцца па-за дыяпазонам 1-20');
end;
.....
case sezon of
  zima: writeln('зіма');
  vesna: writeln('вясна');
  leto: writeln('лето');
  osen: writeln('восень');
end;
.....
case s of
  'a'..'z':writeln('малая лацінская літара');
  '0'..'9':writeln('лічба');
end.
```

Аператар безумоўнага пераходу *goto* прымяняецца ў тым выпадку, калі пасля выканання некаторага аператара трэба выканаць не наступны па парадку, а які-небудзь другі, памечаны меткай аператар. Меткі апісваюцца ў пачатку праграмы, у раздзеле *label*.