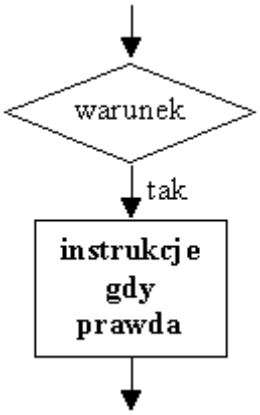
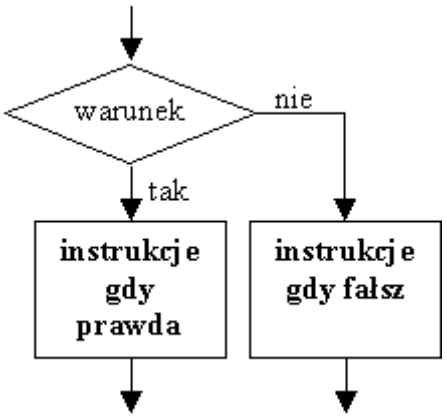
INSTRUKCJA WARUNKOWA



Niestety, nie zawsze da się skonstruować programu w prosty sposób - instrukcja po instrukcji. Bardzo często zdarza się, że w pewnym momencie musimy zdecydować, "w którą iść stronę" - który fragment programu wykonać. Na przykład: jeśli współrzędna X będzie większa niż 100 to zmień kolor rysowania na niebieski albo jeśli współrzędna X jest mniejsza od 100 dodaj do zmiennej X 1(jeden) w przeciwnym wypadku wyzeruj zmienną X. Polecenia te mają swoje odpowiedniki w Pascalu.

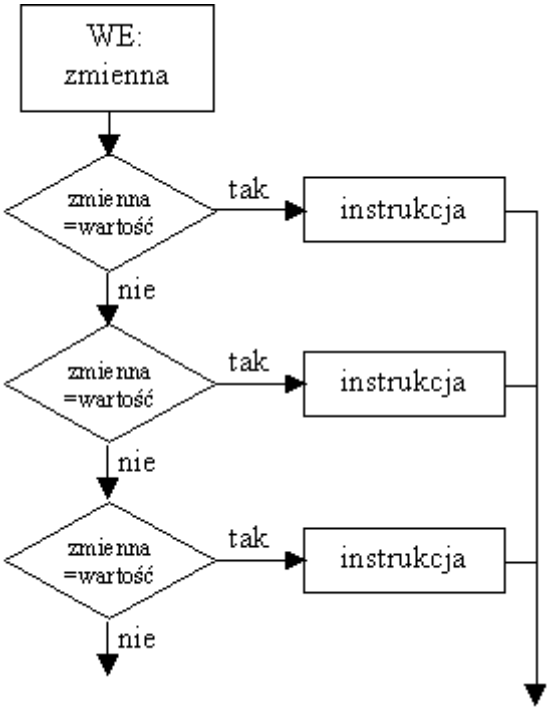
Instrukcja warunkowa
IF warunek **THEN** instrukcja
```
IF  x < 640  THEN  x:=x+1;
```
Jeśli warunek jest prawdziwy, to wykonaj instrukcję



IF warunek **THEN** instrukcja1 **ELSE** instrukcja2
```
IF  x < 640  THEN  x:=x+1  ELSE x:=0;
```
Jeśli warunek jest prawdziwy, to wykonaj instrukcję pierwszą, w przeciwnym wypadku - wykonaj instrukcję drugą (gdy warunek jest fałszywy).

W pierwszym przypadku ignorowana jest instrukcja druga (warunek prawdziwy), w drugim - ignorowana instrukcja pierwsza (warunek fałszywy).

Instrukcja wyboru
Jeśli chcemy sprawdzić po kolei kilka warunków i wybrać z nich jeden właściwy można posługiwać się kilkoma instrukcjami warunkowymi w prostym lub złożonym układzie, wygodniej jest jednak użyć instrukcji złożonej
CASE zmienna **OF**
    wartość1: instrukcja;
    wartość2: instrukcja;
    ...
    ELSE instrukcja
END;
Porównujemy zmienną z kolejnymi wartościami. Jeśli jest taka sama, to wykonywana jest instrukcja przy tej wartości.



Można w instrukcji CASE dołożyć część z ELSE - gdyby żadna z wartości nie była równa zmiennej, wykonana zostałaby instrukcja po słowie ELSE.
```
CASE rodzaj OF
  1:line(100,100,300,100);
  2:line(100,100,100,300);
  3:line(100,100,300,300);
  4:line(100,300,300,100);
  5:line(100,100,100,100);
  else line(100,200,300,400);
END;
```

Operatory relacji i logiczne
Podczas operacji logicznych porównujemy ze sobą dwie wartości (tego samego typu). Mogą to być np. liczby albo też informacje zawarte w zmiennych. Warunki logiczne przyjmują zawsze jedną z dwóch wartości PRAWDA lub FAŁSZ. Podczas operacji arytmetycznych posługujemy się operatorami dodawania, odejmowania itd., tak i podczas operacji logicznych mamy operatory relacji.

2=2 prawda	2<3 prawda	2<=3 prawda
2<>2 fałsz	2>3 fałsz	2>=2 prawda

Pojedyncze wyrażenia logiczne można łączyć za pomocą operatorów logicznych:

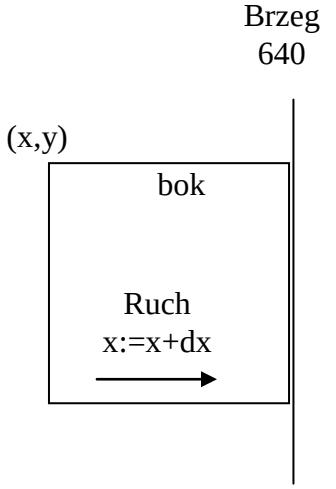
NOT - negacja
(jeśli a=b jest prawdziwe to NOT(a=b) jest fałszywe)
```
IF NOT(x<0) THEN  x:=x-1;
```

OR - suma logiczna (wyrażenie (a=b) OR (c=d) będzie prawdziwe, jeśli przynajmniej jedno z wyrażeń składowych będzie prawdziwe)
```
IF (x>0) OR (x<640) THEN  x:=x+1;
```

AND - iloczyn logiczny (wyrażenie (a=b) OR (c=d) będzie prawdziwe, jeśli będą prawdziwe oba wyrażenia składowe).
```
IF (y=0) AND (x<640) THEN  y:=480;
```

Typy stałych i zmiennych (proste)
Typy liczbowe całkowite
BYTE - liczby całkowite 0..255 (1 bajt)
INTEGER - liczby całkowite -32768..32767 (2 bajty)
WORD - liczby całkowite dodatnie 0..65535 (2 bajty)
LONGINT- liczby całkowite +- ~2 mld (4 bajty)
Typ liczbowy rzeczywisty (6 bajtów)
REAL - dowolne liczby rzeczywiste przedstawione jako ułamek dziesiętny (5.347) lub w postaci wykładniczej (1.2E+3 czyli $1.2*10^3$ czyli 1200)
Typ napisowy (tyle bajtów ile znaków w tekście)
STRING - teksty do 255 znaków w apostrofach, np. ‘Ala ma kota’, ‘a’, ‘’ - pusty
Typ znakowy (1 bajt)
CHAR - pojedyncze znaki ujęte w apostrofy, np.: ‘a’, ‘A’, ‘1’ (to nie to samo co 1)
Typ logiczny (1 bajt)
BOOLEAN - zmienne przyjmują wartość prawdy TRUE (1) lub fałszu FALSE (0)

Animacja kwadratu z poprzedniej lekcji z wykorzystaniem złożonych warunków logicznych.
Jeżeli kwadrat (lub dowolna inna figura) przesuwa się w prawo, to zwiększa się współrzędna x. Sprawdzamy, kiedy współrzędna x osiągnie wartość prawego brzegu ekranu (640). Zmniejszamy tą wartość o szerokość kwadratu, aby odbicie następowało we właściwym momencie. Skoro współrzędna x była zwiększana za każdym razem (o wartość dx), to po ruch w drugą stronę spowodujemy, gdy wartość zmiennej x będzie zmniejszana (wystarczy zmienić znak zmiennej dx na przeciwny)




```
if x > (640-bok) then dx:=-dx;
```

W podobny sposób sprawdzamy, czy kwadrat osiągnie lewy brzeg ekranu I znów zmieniamy znak zmiennej dx na przeciwny (był ujemny a teraz stanie się dodatni
```
if x < 0 then dx:=-dx;
```

Wykorzystując sumę logiczną, można te dwa warunki połączyć w jeden
```
if (x > (640-bok)) or (x < 0) then dx:=-dx;
```
jeśli współrzędna X lewego górnego rogu kwadratu przekroczy wartość 640-bok (tzn. prawa krawędź kwadratu zetknie się z prawym brzegiem ekranu) LUB współrzędna X będzie mniejsza od 0 (tzn. lewa krawędź kwadratu przekroczy lewy brzeg ekranu) TO zmień kierunek ruchu w poziomie na przeciwny (z 1 na -1 lub z -1 na 1)

I podobne działanie podczas ruchu w pionie, tym razem dla współrzędnej y
```
if (y > (480-bok)) or (y < 0) then dy:=-dy;
```

Jeśli jednocześnie będziemy zmieniać współrzędną x i y, kwadrat będzie poruszał się po przekątnych. Jeśli wartości dx i dy będą jednakowe to pod kątem 45 stopni. Gdy te wartości będą się zmieniać, to kwadrat zacznie poruszać się po innych przekątnych.

GRAF14.PAS - odbijający się od ścian kwadrat - fragment

```
...
repeat
  Kwadrat(x,y,bok,kolor);                    {rysowanie}
  delay(5);                                  {wstrzymanie}
  Kwadrat(x,y,bok,0);                        {wymazanie}
  x:=x+dx;                                   {nowe współrzędne}
  y:=y+dy;
  if (x > (640-bok)) or (x < 0) then dx:=-dx; {odbicie od ściany prawej i lewej}
  if (y > (480-bok)) or (y < 0) then dy:=-dy; {odbicie od ściany górnej i dolnej}
                                             {jeśli odbicie, to zmiana kierunku ruchu}
until keypressed;                            {koniec animacji gdy dowolny klawisz}
...
```

Ćwiczenia (na podstawie programu grafika14.pas)

Zmień program w ten sposób, aby kwadrat poruszał się nie po całym ekranie, ale w prostokącie, którego lewy górny róg znajduje się w punkcie 100,100 a prawy dolny w punkcie 500,400.

Zmień program tak, aby po każdym odbiciu od ściany kwadrat zmieniał kolor na inny, może być losowo wybrany.

Zmień program tak, aby po ekranie poruszał się prostokąt, trójkąt, twoje inicjały.

Zmień program tak, aby po 10 odbiciach od ścianki dolnej program się automatycznie zakończył