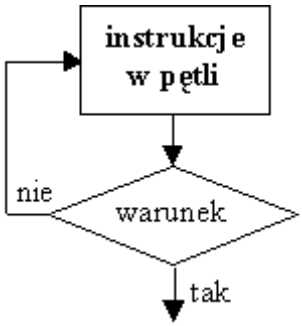
pętla logiczna REPEAT...UNTIL... (powtarzaj aż)



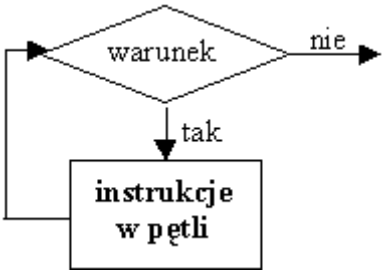

```
REPEAT
    instrukcja;
    instrukcja;
    ...
UNTIL warunek;
```

Ciąg instrukcji pomiędzy słowami repeat until jest powtarzany, aż warunek logiczny stanie się prawdziwy. Dopóki jest fałszywy instrukcje są wykonywane. Warunek logiczny jest sprawdzany po wykonaniu instrukcji i dlatego instrukcje muszą się wykonać przynajmniej jeden raz. Zmiana zmiennej sterującej musi być w tej pętli realizowana „ręcznie" - inaczej niż to było w instrukcji for. Zmienna musi też być ustawiona na początku (zainicjowana), aby w trakcie wykonywania pętli nie przyjmowała wartości nieokreślonych.

```
i:=0;
REPEAT
   i:=i+1;
   line(100+i,100,100+i,200)
UNTIL i>=10;
```

Powyższy fragment rysuje 10 pionowych linii

pętla logiczna WHILE...DO... (dopóki)




```
WHILE warunek DO
    instrukcja;
```

Jeśli w pętli trzeba wykonać kilka instrukcji stosujemy instrukcję złożoną BEGIN...END

```
WHILE warunek DO
BEGIN
    instrukcja;
    instrukcja;
    ...
END;

i:=0;
WHILE  i<10 DO
BEGIN
   i:=i+1;
   line(100+i,100,100+i,200)
END;
```

Powyższy fragment rysuje 10 pionowych linii

Instrukcja w pętli jest wykonywana dopóki warunek logiczny jest prawdziwy. Gdy warunek logiczny stanie się fałszywy, pętla kończy działanie. Najpierw sprawdzany jest warunek, a potem dopiero wykonuje działanie. Gdy. Instrukcja nie wykona się ani raz jeśli warunek będzie spełniony zaraz na początku.

Animacja komputerowa

Wrażenie, że obiekty są ruchome uzyskujemy dzięki szybkiemu następowaniu po sobie kolejnych {nieruchomych) obrazów. Aby "oszukać" wzrok - by animacja była płynna - kolejne obrazy muszą być wyświetlane w tempie przynajmniej 30 obrazów na sekundę. W praktycznych zastosowaniach animacja realizowana jest w różny sposób. Najprostszym z nich jest rysowanie i wymazywanie kolejnych obrazków na ekranie. Po narysowaniu obrazka wstrzymujemy na chwilę działanie programu i nie wymazujemy go od razu - byłby widoczny zbyt krótko i oko mogłoby w ogóle go nie zauważyć albo byłoby widać "mruganie".

Animacja realizowana jest schematycznie w następującej pętli:

- **rysujemy obiekt**
- **wstrzymujemy działanie programu**
- **usuwamy obiekt**
- **obliczamy nowe położenie obiektu**

```
fragment programu GRAF11 przesuwanie kwadratu po ekranie

  {zainicjować zmienne}
  x:=1;
  y:=100;
  w:=40;
  s:=40;
  k:=10;
  repeat
    ProstokatK(x,y,s,w,k);    {rysujemy obiekt}
    delay(10);                {wstrzymujemy działanie programu, żeby zobaczyć obiekt}
    ProstokatK(x,y,s,w,0);    {usuwamy obiekt z ekranu}
    x:=x+1;                   {wyliczamy nowe współrzędne}
  until x > (640-s);          {sprawdzamy kiedy koniec animacji}
```

taki sam efekt przesuwania można zrealizować za pomocą pętli WHILE - tutaj na początku sprawdzamy czy możemy dalej przesuwać obiekt

```
while  x < (640-s) do
begin
    ProstokatK(x,y,s,w,k);
    delay(10);
    ProstokatK(x,y,s,w,0);
    x:=x+1;
end;
```

Odbijanie od brzegów

```
repeat
  repeat
    ProstokatK(x,y,s,w,k);
    delay(10);
    ProstokatK(x,y,s,w,0);
    x:=x+dx;                  {tutaj zwiększamy wartość x - ruch w prawo}
  until x>(640-s)

  repeat
    ProstokatK(x,y,s,w,k);
    delay(10);
    ProstokatK(x,y,s,w,0);
    x:=x-dx;                  {tutaj zmniejszamy wartość x - ruch w lewo}
  until x<0
until KEYPRESSED;
```

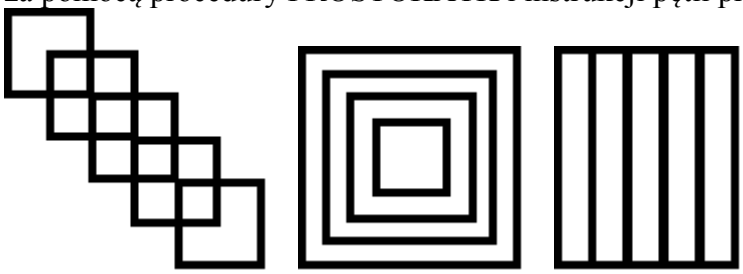
funkcja KEYPRESSED zwraca wartość TRUE gdy naciśniemy jakikolwiek klawisz.

Ćwiczenia

Za pomocą linii pionowych lub poziomych wypełnij kwadrat dowolnym kolorem używając pętli REPEAT lub WHILE oraz LINE, LINETO lub LINEREL

narysuj za pomocą LINE, LINETO, LINEREL następujące rysunki (długości i odległości dowolne):

za pomocą procedury PROSTOKATK i instrukcji pętli przygotuj następujące rysunki:



Za pomocą pętli REPEAT lub WHILE oraz procedury ProstokatK napisz program, który będzie powiększał bok kwadratu od 10 do 100 punktów a następnie go zmniejszał do 10.
Dopisz do poprzedniego programu instrukcje, które spowodują, że podczas powiększania kwadrat będzie się przesuwał w prawo, a podczas zmniejszania - w lewo.