Szkoła Podstawowa nr 2 w Lublinie

AUTORSKI PROGRAM NAUCZANIA

MSWLogo - programowanie dla klas 6

Ireneusz Kołodziej
2012-09-01

I Co to jest Logo?

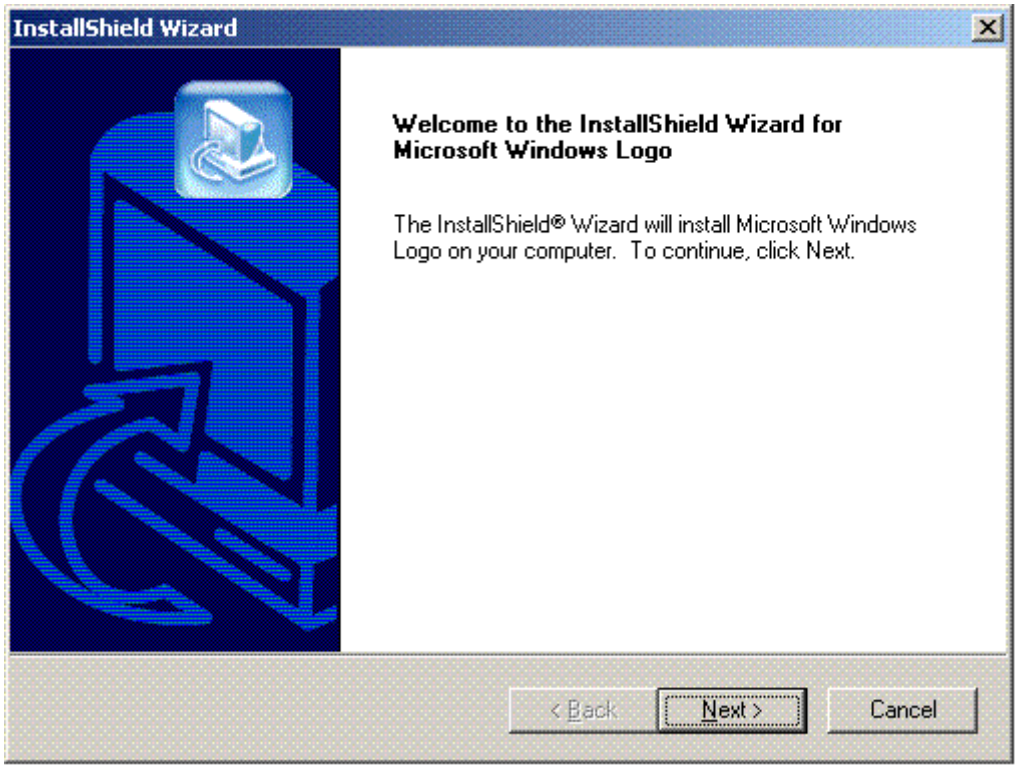
Logo jest prawdziwym językiem programowania. Wymyślony i stworzony został w celu zapoznania Cię z technikami pisania programów komputerowych, a dokładniej procedur. Procedurę można porównać z pojedynczym klockiem. Używając takich klocków możesz z nich budować bardziej skomplikowane konstrukcje.
Wynik takiego budowania zobaczysz natychmiast na ekranie monitora - najczęściej w postaci rysunku lub animacji. Możesz również stworzyć grafikę trójwymiarową albo skomponować prostą melodię. Zaawansowani programiści napiszą aplikacje okienkowe jakie znasz z rodziny systemów Windows.

II Instalujemy MSWLogo.

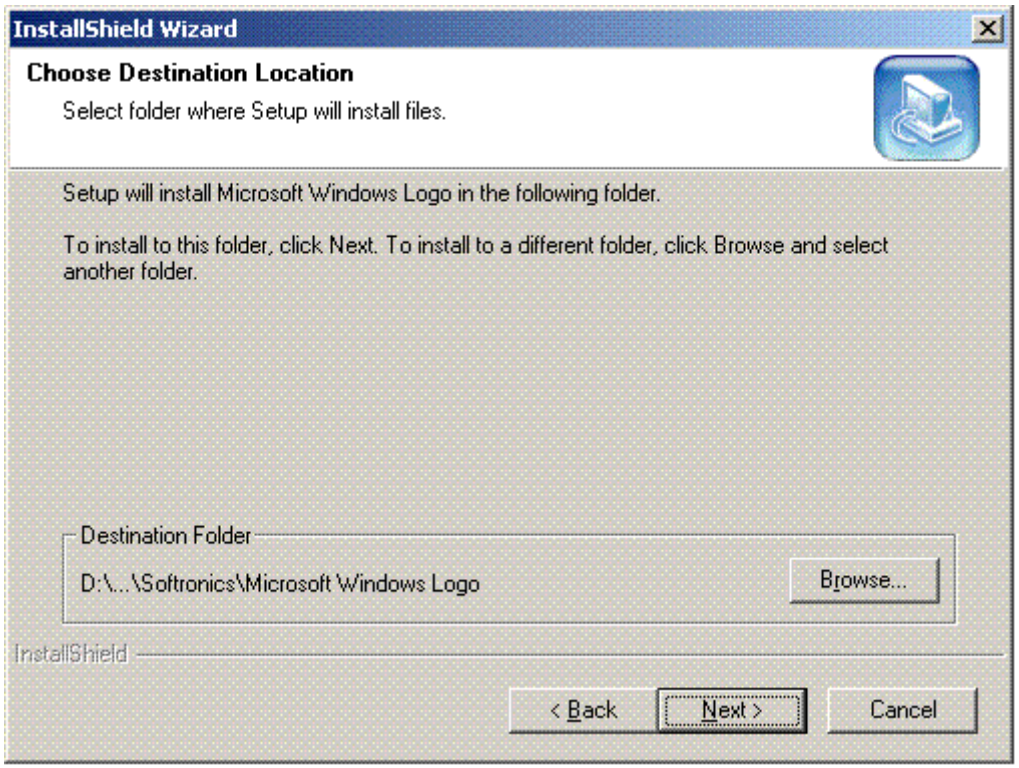
MSWLogo jest programem darmowym (licencja *freeware*). Będziesz korzystał z wersji angielskojęzycznej. Program ściągnij sobie ze strony producenta: http://www.softronix.com/logo.html

Są to dwa pliki:
1 *Setup kit – plik najważniejszy dla systemów Windows (1,7 MB)*
2 Win32Help Kit for Windows 7 - *plik pomocy dla Windows 7 (600 KB)*
3 Win32Help Kit for Vista - *plik pomocy dla Windows Vista (600 KB)*

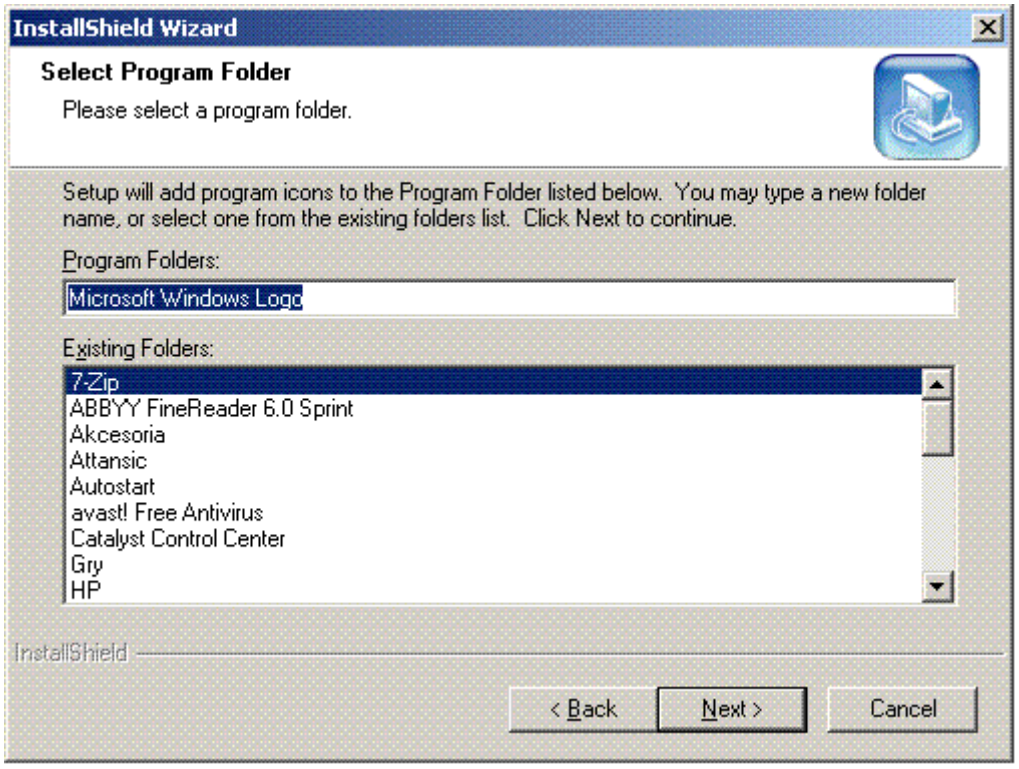
Kliknij dwukrotnie na ściągnięty plik jako administrator systemu Windows. Program zacznie się instalować.



W powitalnym ekranie klikasz *Next*.



Następny ekran proponuje domyślny folder, w którym program zostanie zainstalowany – kliknij *Next*.

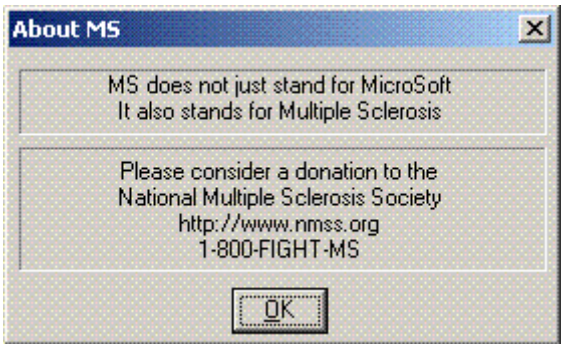


Trzeci ekran proponuje Ci folder w menu START – kliknij *Next* po raz trzeci i poczekaj chwilę na dokończenie instalacji.

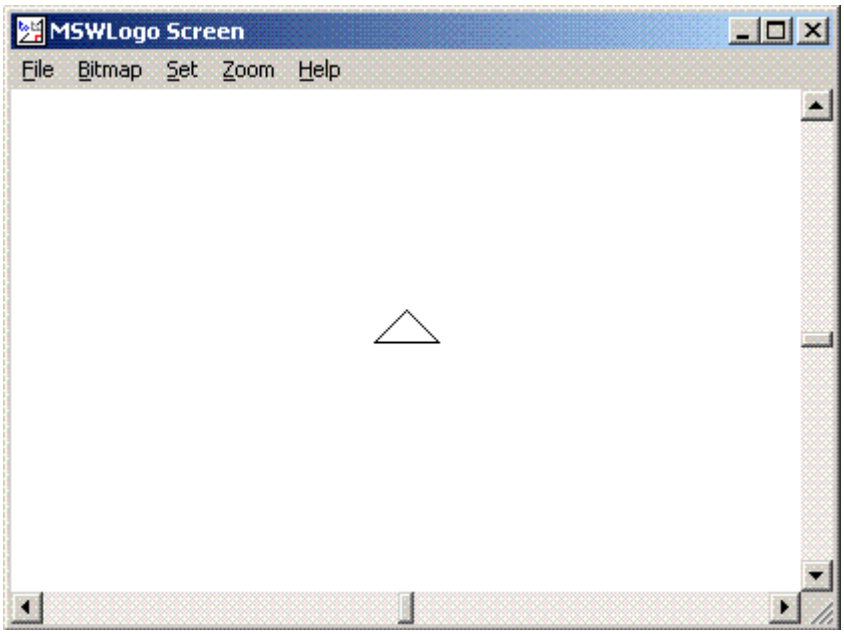
III Uruchamiamy program

START →Wszystkie programy →Microsoft Windows Logo → Microsoft Windows Logo.

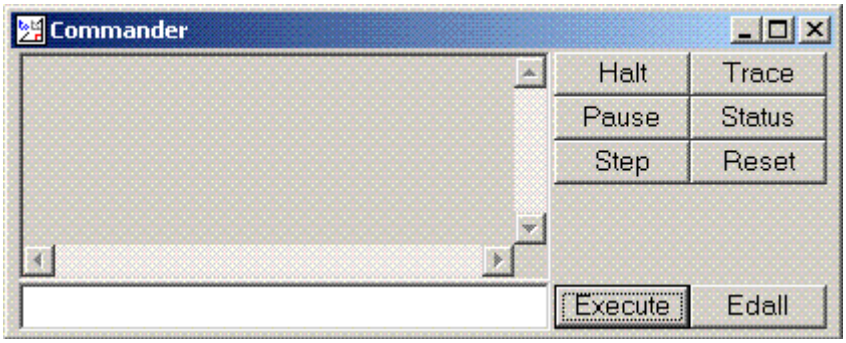
Klikamy OK w pierwszym oknie z informacjami o producencie tej wersji programu i o programistach, którzy przyczynili się do tak wyglądającej i działającej jego wersji.



W górnym prostokącie mały żart, w środkowym prośba o rozważenie datku. (Ciebie raczej to nie dotyczy :). Drugi raz OK i jesteśmy w programie.



Górne okno programu to miejsce do rysowania dla żółwia (trójkąt).



Dolne to miejsce do wpisywania żółwiowi komend, których ma słuchać (biały prostokąt). Po prawej stronie okna parę użytecznych przycisków z bardzo ważnym *Execute* – wykonaj.

Proponuję Ci krótki przegląd możliwości MSWLogo. W oknie górnym kliknij Help →Demo. Klikaj kolejne TAK , to co zobaczysz będziesz mógł już niedługo sam zaprogramować!

IV Zaczynamy programować.

Rozkazy wydajesz żółwiowi w języku angielskim. Pisać możesz wielkimi lub małymi literami, jak Ci wygodniej. Możesz używać całych wyrazów lub skrótów. Zachęcam Cię do tego drugiego sposobu – będziesz miał mniej pisania.

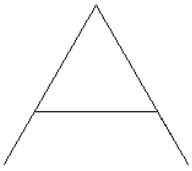
1 Pierwsze ruchy żółwia.

Idź do przodu, po angielsku **FORWARD** i koniecznie odległość jaką ma pokonać.
FD 100 – żółw narysuje odcinek o długości 100 pikseli w kierunku wierzchołka przy ramionach, licząc od środka podstawy trójkąta.
Wykonaj odcinek o długości 200 pikseli. Kliknij w dolnym oknie (Commander) przycisk *Reset* i narysuj odcinek o długości 10 pikseli.

Cofnij się, **BACK** (zawsze spacja) plus długość w pikselach.
BK 100 – odcinek o długości 100 pikseli w kierunku przeciwnym do wierzchołka przy ramionach (w stronę podstawy trójkąta).
Cofając się wykonaj odcinek o długości 150 pikseli. Kliknij przycisk *Reset* i narysuj do tyłu odcinek o długości 50 pikseli.

Obróć się w prawo, **RIGHT** oraz kąt obrotu.
RT 90 – żółw obraca się o 90^0 wokół własnej osi.
Obróć żółwiem o 45^0. Kliknij przycisk *Reset* i narysuj wielką literę „L" używając poleceń **BK**, **RT** i **FD**.

Obróć się w lewo, **LEFT** oraz kąt obrotu.
LT 90 – żółw obraca się o 60^0 wokół własnej osi.
Obróć żółwiem o 120^0. Wciśnij przycisk *Reset* i narysuj wielką literę „N" używając poleceń **FD, RT** i **LT**.
Poćwicz z innymi wielkimi literami bez zaokrągleń: A, E, F, H, K, M, T, W, Y i Z.



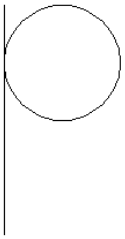
2 Niektóre użyteczne polecenia

Czasami należy podnieść żółwia, by przesunąć go bez zostawiania po sobie śladu. Do ponownego rysowania żółwia należy go ponownie opuścić.
PU skrót od **PENUP** – żółw do góry.
PD skrót od **PENDOWN** – żółw na dół.
Nie możesz cofnąć ruchów żółwia gdy błędnie coś narysujesz. Jedyny ratunek to zamienić żółwia na gumkę i cofnąć się po źle narysowanym śladzie.

PE skrót od **PENERASE** – żółw na gumkę.
Powrót do funkcji rysowania:
PPT skrót od **PENPAINT** – żółw na pisak.

3 Na okrągło!

Narysuj okrąg - w języku angielskim **CIRCLE** i koniecznie długość promienia.
CIRCLE 100 – Żółwik będąc w środku rysuje okrąg o promieniu 100 pikseli.
Wydaj polecenie narysowania pięciu okręgów o promieniach: 10, 20, 30, 40 i 50 pikseli bez resetowania poprzednich okręgów.
Poćwicz rysując małe litery: p, d, q. (Nie zapomnij o podnoszeniu i opuszczaniu żółwia). Np.:



Narysuj łuk **ARC**, ta procedura ma dwa parametry; kąt obrotu promienia i jego długość. Wypróbuj to polecenie z parametrami wpisanymi w odwrotnej kolejności, dla przykładu:
ARC 90 200, ARC 200 90 – czy już wiesz, który parametr to kąt, a który długość promienia? Zapamiętaj, z którego miejsca żółw zaczyna rysować łuki!
Poćwicz rysując duże litery: C, D, J, P, U, S, G.
Poćwicz rysując małe litery: e, f, h, r, n, m. Np.:

Elipsa (**ELLIPSE**) również wymaga dwóch parametrów: Spróbuj dwóch komend:
ELLIPSE 50 100, **ELLIPSE 100 50** – zauważyłeś różnicę?

4 Rysuj na kolorowo.

Żółwia można zmusić do rysowania na kolorowo.
Zmiana koloru pisaka to **SETPENCOLOR** plus liczba od 0 do 15 (w wersji uproszczonej). W skrócie:
SETPC n
Oto zestaw numery – kolory, który przyda Ci się do rysowania:

0 – czarny
1- niebieski

2- zielony
3 - błękitny
4 - czerwony
5 - różowy
6 - żółty
7 - biały
8 - brązowy
9 - jasnobrązowy
10 - ciemnozielony
11 - turkusowy
12 - jasnoczerwony
13 - jasnofioletowy
14 - pomarańczowy
15 – szary

Nie musisz rysować na białym tle. Zmiana koloru całej powierzchni rysunku to polecenie **SETSCREENCOLOR** plus liczba od 0 do 15. Krótko:
SETSC n

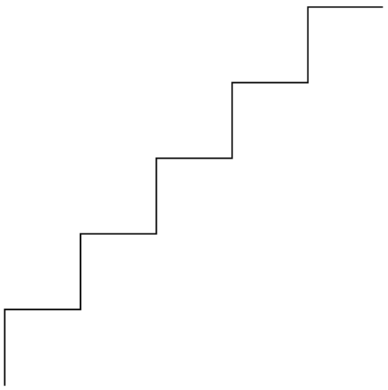
Zamknięte obszary rysunku możesz wypełniać kolorem za pomocą zestawu dwóch poleceń – **SETFLOODCOLOR** ustawia kolor wypełnienia, a następnie wjeżdżasz żółwiem do środka zalewanego obszaru i wypełniasz go rozkazem **FILL** praktycznie wydajesz takie rozkazy:
SETFC n
FILL

Dodatkowo możesz ustawić rozmiar, którym rysujesz, rozkazem **SETPENSIZE**.
SETPENSIZE [n n] – liczby w nawiasach kwadratowych oznaczają szerokość i wysokość prostokątnego pisaka, którym rysuje żółw. W MSWLogo szerokość nie jest brana pod uwagę, więc praktycznie **n** ustawia bok kwadratu.

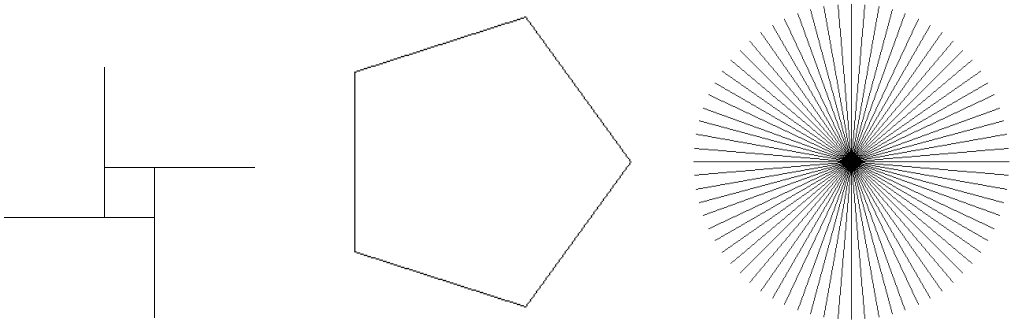
5 Powtarzające się rozkazy.

Wydaj żółwiowi polecenia w celu narysowania kwadratu. Co zauważyłeś? Sekwencje powtarzających się rozkazów można wykonać w jednym poleceniu **REPEAT** – powtórz. To polecenie jest bardziej skomplikowane od wszystkich poprzednich. Mówimy, że ma trudniejszą składnię – to taka informatyczna „gramatyka” języka programowania:
REPEAT n [tu wpisujemy rozkazy] – powtórz „n” razy polecenia w nawiasach

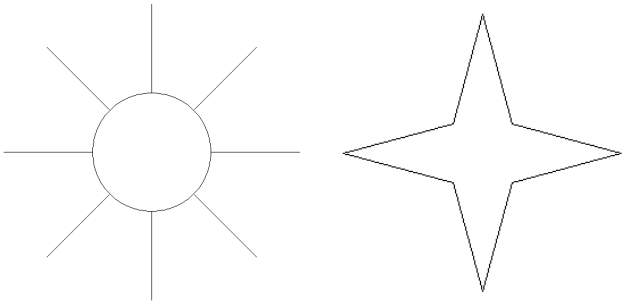
kwadratowych.
W przypadku kwadratu napiszesz: **REPEAT 4 [FD 100 RT 90]** – powtórz 4 razy dwa polecenia: idź do przodu 100 pikseli następnie obróć się o 90^0.
Używając polecenia **REPEAT** wykonaj podobne schodki:

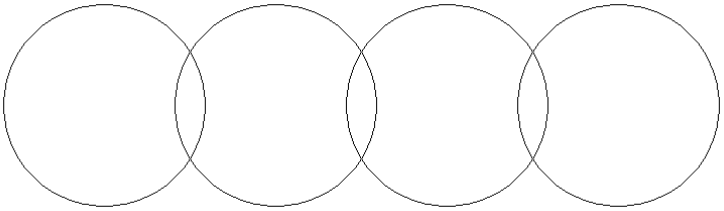


Spróbuj narysować żółwiem trójkąt równoboczny, początkowo rozkaz po rozkazie. Jeżeli Ci się uda użyj polecenia **REPEAT**. (Przydaje się matematyka!)
Poćwicz z następującymi kształtami:

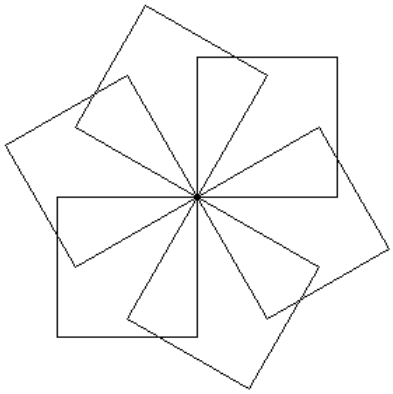


Kolejne ciekawe kształty to słoneczko, gwiazdka albo symbol Audi.



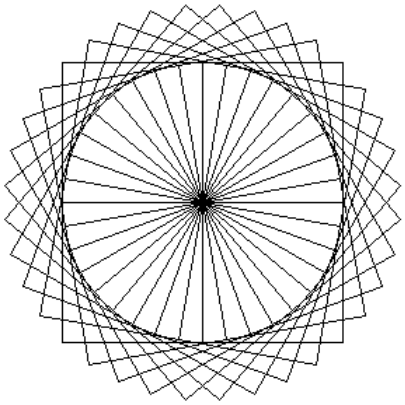


Bardzo ciekawą możliwością programowania jest zagnieżdżanie poleceń! Wyobraź sobie kwadrat obracający się wokół dolnego lewego wierzchołka. Jeżeli nie dasz rady żółw narysuje Ci to. Wydaj mu polecenie:
REPEAT 6 [*REPEAT 4 [FD 100 RT 90]* RT 60] – efekt wygląda tak:



Nie przerażaj się! Już niedługo uprościmy sobie proces programowania tworząc własne rozkazy (procedury). Tymczasem przeanalizujmy ostatnie polecenie. Wewnętrzny ***REPEAT*** napisany *kursywą* rysuje kwadrat. Zewnętrzny **REPEAT** rysuje sześć kwadratów obracając kolejny o 60^0 w prawo. ($360^0 : 6 = 60^0$)

Poniższy rysunek przedstawia 36 kwadratów obróconych wokół wierzchołka.



Spróbuj zmusić żółwia do narysowania takiej rozetki. Dobierz odpowiedni kąt obrotu kwadratów do ich ilości!

6 Żółwiu poczekaj!

Jeśli chciałbyś zobaczyć jak żółw rysuje kolejne kwadraty możesz go zmusić do wolniejszego rysowania dowolnych elementów rysunku poleceniem **WAIT**.

WAIT n – czekaj z rysowaniem — s · **n** chcąc zmusić żółwia do jednosekundowej pauzy za **n** podstaw liczbę 60.

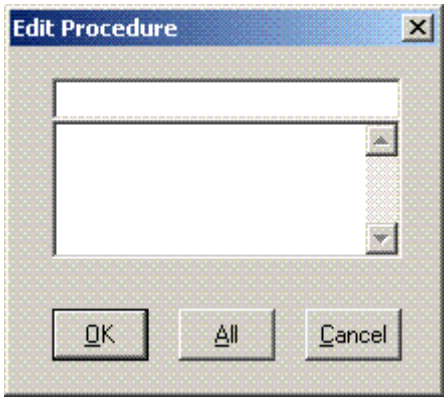
Np.: **REPEAT 6 [*REPEAT 4 [FD 100 RT 90]* RT 60 WAIT 30]** i obserwuj co – s jak żółw rysuje kolejny kwadrat.

7 Gramy melodię.

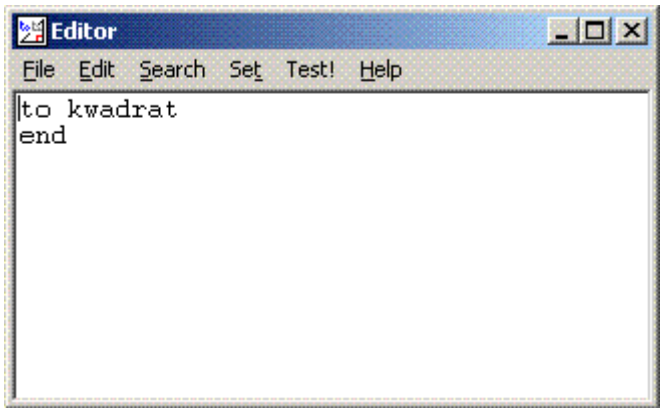
Program generuje dźwięki jednogłosowe wymuszane rozkazem **SOUND**.
SOUND [f t] – **f** to częstotliwość dźwięku w Hz, czyli liczbie drgań na sekundę, a **t** czas jego trwania. Pojedynczy dźwięk jest mało atrakcyjny. Możesz skomponować prostą sekwencję wpisując w kwadratowych nawiasach kilka par wysokość dźwięku i czas jego trwania **f t**:
SOUND [800 500 1000 500 1400 1000] – posłuchaj sam.
Jeżeli grasz na jakimś instrumencie nie będziesz miał problemu ze zmuszeniem żółwia do zagrania „Wlazł kotek na płotek”!

8 Uczymy żółwia naszych rozkazów!

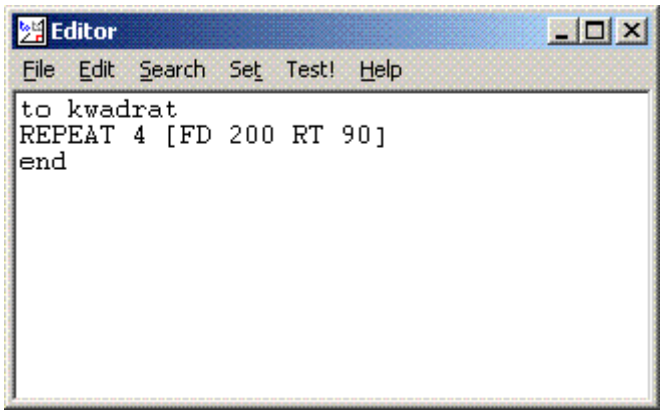
Wydaj żółwiowi rozkaz **KWADRAT**. Żółw odpowie w szarym okienku nad komanderem: *I don't know how to KWADRAT* – w wolnym tłumaczeniu *nie wiem co mam zrobić z tym KWADRAT!* No to go musisz tego nauczyć!
Z głównego menu programu wybierz File →Edit…



Na górze okienka procedury w wąski prostokąt wpisz **KWADRAT** i kliknij OK.
Otwiera się okno edycji rozkazu (procedury):



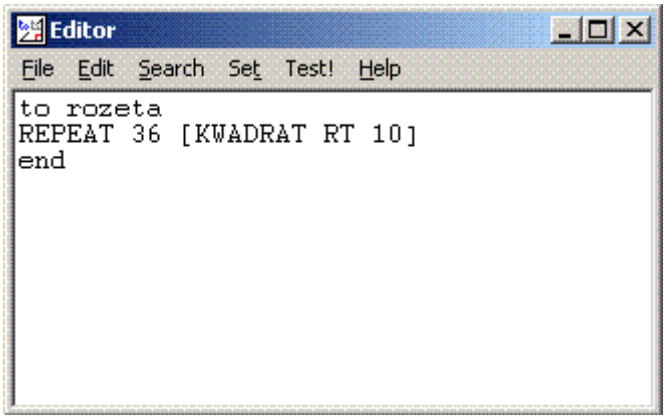
Wprowadź pustą linię między *to kwadrat* a *end* i wpisz rozkaz rysujący kwadrat – najlepiej z wykorzystaniem polecenia **REPEAT**.



W okienku Editor kliknij File → Save and Exit. Ponownie w komanderze wpisujemy KWADRAT i narysowane! To jeszcze nic takiego.

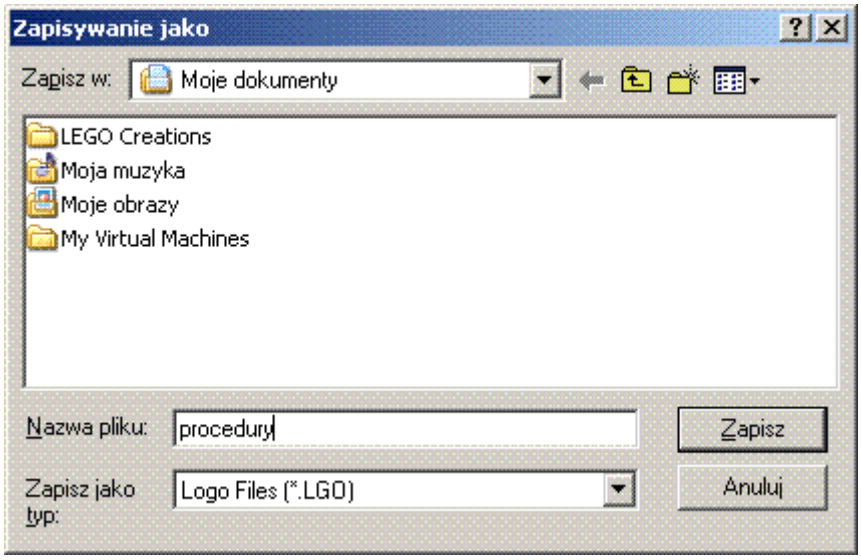
UWAŻAJ TERAZ!

Nauczysz żółwia rozetki z punktu 5. Powtórz czynności z początku punktu 8. Z głównego menu programu wybierz File → Edit… Wpisz nazwę procedury ROZETA. W oknie Editor wpisz: **REPEAT 36 [*REPEAT 4 [FD 100 RT 90]* RT 10]**. Tylko po co męczyć się z ***REPEAT 4 [FD 100 RT 90]*** skoro żółw zna poprzednie polecenie **KWADRAT**! Sprytniej będzie **REPEAT 6 [KWADRAT RT 60]**

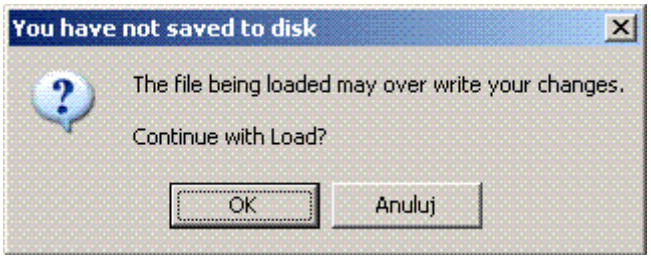


Takie zagnieżdżanie procedur (poleceń, rozkazów) jest potężnym narzędziem programisty dzięki czemu możesz programować tak jak składa się klocki!

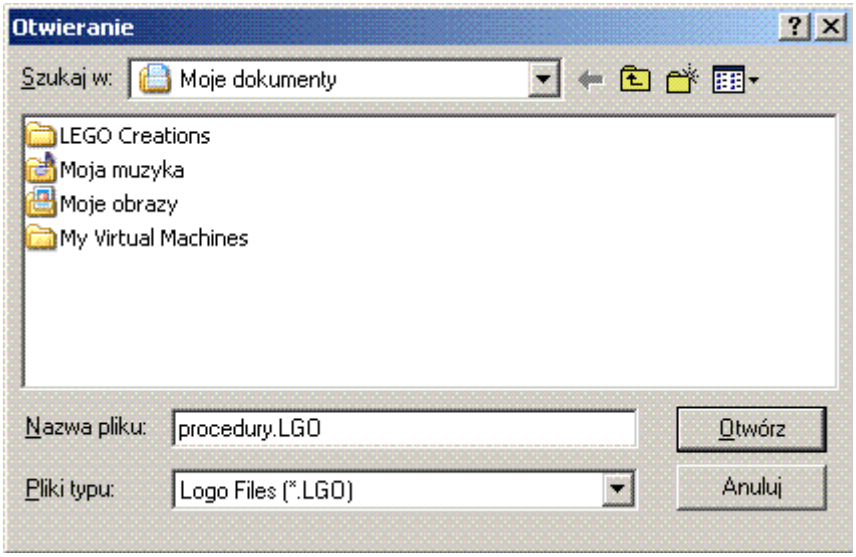
Jeśli nie chcesz stracić utworzonych przed chwilą własnych procedur, to przed zamknięciem MSWLogo zapisz je na dysku: File → Save pod jakąś nazwą np.:



Po otwarciu programu możesz je załadować ponownie: File → Load…

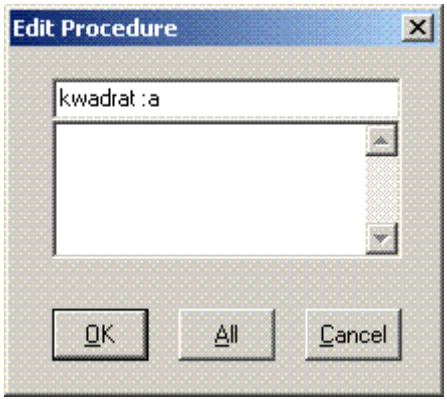


Klikasz OK a później Otwórz i procedury ponownie są do Twojej dyspozycji.

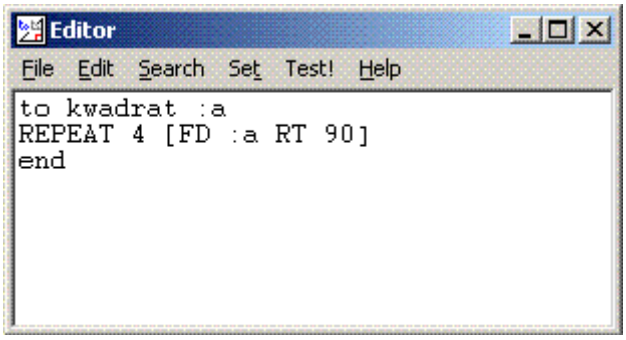


9 Procedura z parametrem.

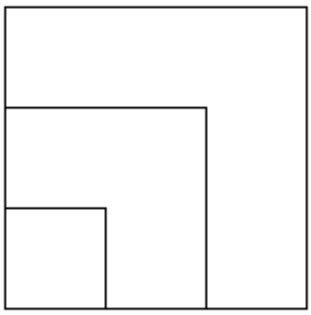
W lekcji poprzedniej stworzyłeś procedurę KWADRAT. Każde wywołanie tej procedury powoduje narysowanie kwadratu o boku 200 pikseli. Sztywne ustalenie długości boku figury jest dużym ograniczeniem.
Na szczęście jest na to rozwiązanie. Jeżeli wydajesz polecenie **FD** dodajesz parametr w postaci liczby, na jaką odległość przesunie się żółw. Podobnie można zrobić z długością boku kwadratu. Wywołaj z głównego menu programu okienko edycji procedury - File →Edit…



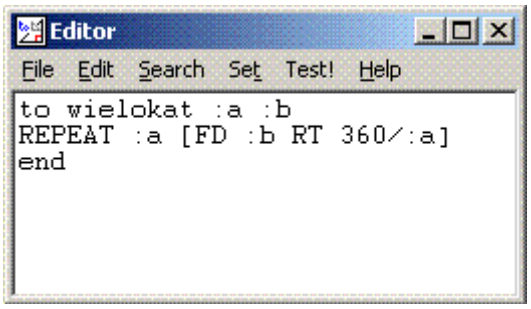
Wpisz nazwę procedury z parametrem w postaci dwukropek litera **:a** (nie pomyl kolejności) i zatwierdź OK. W oknie edytora wpisz **REPEAT 4 [FD :a RT 90]**
i zapisz gotową procedurę z parametrem - kliknij File → Save and Exit.



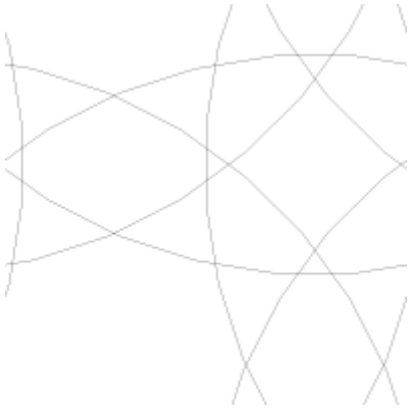
Procedurę wywołaj w następujący sposób: **KWADRAT n** – gdzie **n** jest liczbą pikseli odpowiadającą za długość boku.
Wpisz do Commandera kolejno: **KWADRAT 50**, **KWADRAT 100**, **KWADRAT 150**.
Powinieneś otrzymać coś takiego:



Jako ćwiczenie stwórz procedurę z parametrem rysującą trójkąty o zadanej długości boku. Jeżeli Ci się udało naucz żółwia rysować sześciokąty o różnych bokach. A co z pięciokątami, siedmiokątami, ośmiokątami, itd.?
Czy dasz radę stworzyć uniwersalną procedurę, która zmusi żółwia do narysowania wielokąta o dowolnej liczbie boków? Na początek, dla ułatwienia sobie zadania, ustal długość boku wielokąta na 20 pikseli. Później stworzysz procedurę z dwoma parametrami, w której będziesz mógł zdecydować o liczbie boków i ich długości! Pamiętasz polecenie **ARC 90 150** z dwoma parametrami? Jeżeli udało Ci się zaprogramować rysowanie trójkątów i sześciokątów to Ci gratuluję, a jak jeszcze poczułeś się zadowolony z efektu – masz zadatki na prawdziwego programistę.
Ostatecznie, uniwersalna procedura dla dowolnego wielokąta, wygląda następująco:

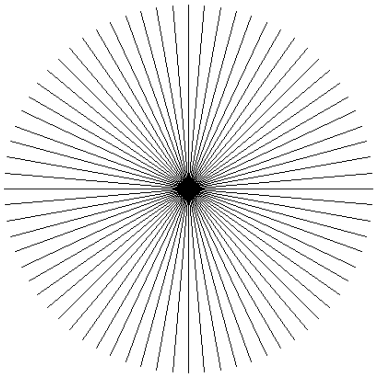


Wywołanie procedury **WIELOKAT 8 30** spowoduje narysowanie przez żółwia ośmiokąta o boku 30 pikseli. Narysuj różne wielokąty testując napisaną procedurę – pamiętaj o zmniejszaniu długości boków przy rosnącej ich ilości. Może się zdarzyć, że żółw nie zmieści wielokąta w obszarze rysowania i otrzymasz rysunek podobny do poniższego. Zmniejsz po prostu bok.



10 Ruchome rysunki – animacja w MSWLogo.

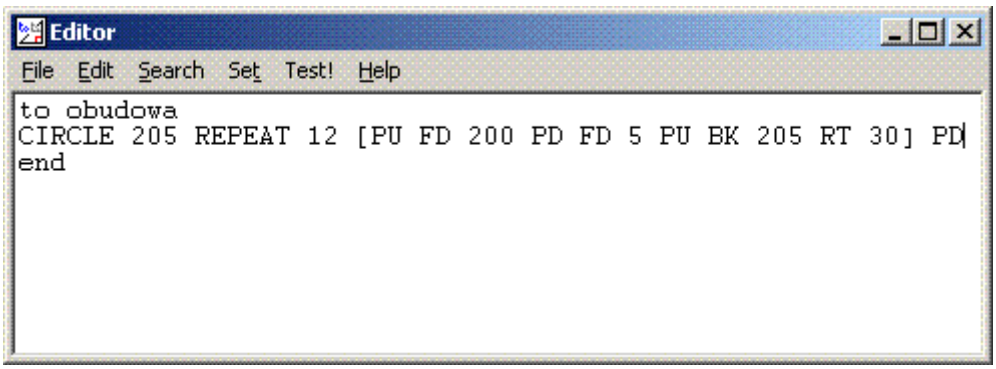
Zapewne oglądałeś filmy-kreskówki. Czy zastanawiałeś się jak są tworzone? Otóż przed oczami widza przesuwają się nieruchome obrazki z odpowiednią szybkością. 24 rysunki w ciągu jednej sekundy dają wrażenie płynnego ruchu. Animacje stworzysz rysując i wycierając na przemian kolejne rysunki. Pamiętasz ćwiczenie z rozdziału 5?



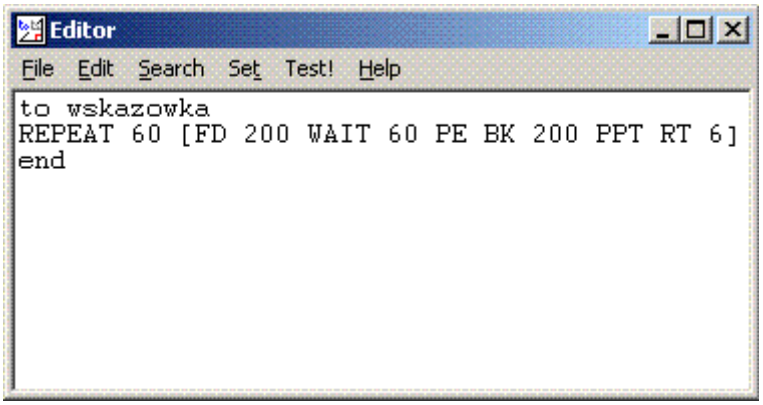
Spróbuj narysować i zaraz wytrzeć każdy z promieni. Nie tak szybko! Nic nie zobaczysz. Przed wytarciem każ żółwiowi poczekać sekundę, a otrzymasz cykającą wskazówkę: **REPEAT 60 [FD 200 WAIT 60 PE BK 200 PPT RT 6]** – super? Wygląda ubogo? Co stoi na przeszkodzie byś dorysował tarczę zegara – najlepiej wykorzystując procedury:

- obudowa
- wskazowka
- stoper (wywołuje obudowa + wskazowka)

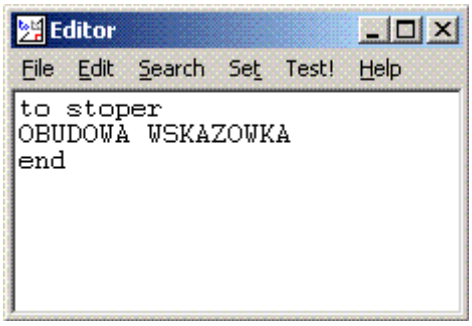
Pierwsza procedura:



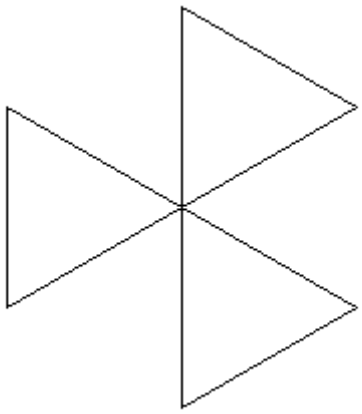
Druga procedura:



Końcowa procedura:



Wywołanie procedury **STOPER** narysuje koło zegara i cykającą w nim wskazówkę. Jak prawdziwy!
Spróbuj zaprogramować cykającą wskazówkę w postaci samotnego, małego kółeczka obiegającego dookoła tarczę.
Jako kolejne ćwiczenie narysuj obracający się wiatrak złożony z trzech równobocznych trójkątów:

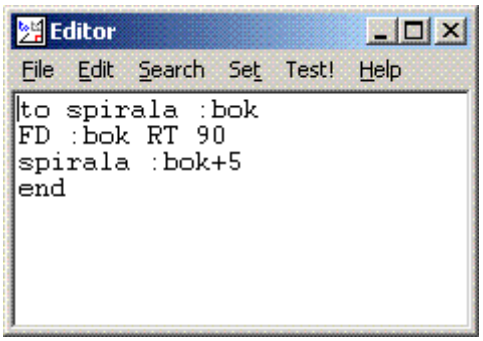


Zastosuj procedury:

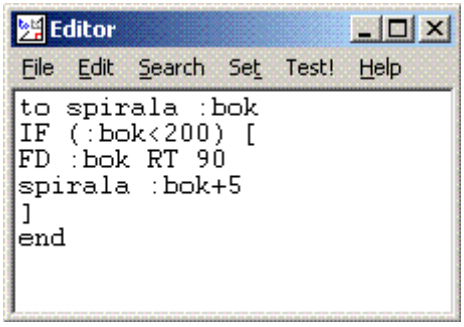
- skrzydla
- wiatraczek

11 Zapętlamy procedury.

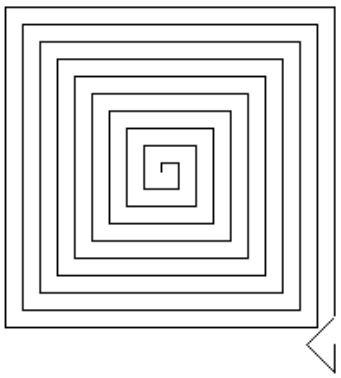
Pomyśl co się stanie jeżeli procedura wywoła sama siebie. Będzie działać w nieskończoność! Można nacisnąć przycisk *Halt*, ale jeśli jest fragmentem drugiej procedury to nie pozwoli jej wykonać się do końca. Jest to ewidentny błąd programisty. Zrealizuj go na przykładzie spirali. Należy rysować kwadrat, w którym każdy bok będzie dłuższy od poprzedniego np. o 5 pikseli. W ten sposób spirala rozwija się w nieskończoność.



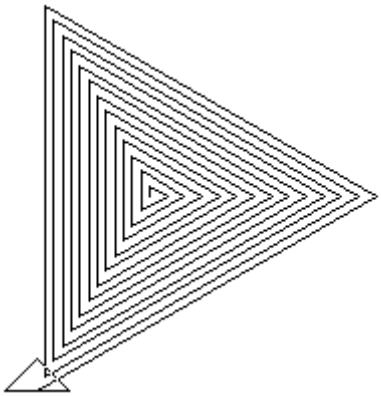
Należy znaleźć sposób na automatyczne zatrzymanie się rysowania spirali. Musisz postawić warunek dotyczący najdłuższego boku spirali. Jeżeli żółw dojdzie do tej długości procedura zatrzyma się. Wykonaj to wykorzystując instrukcję warunkową **if** o następującej składni:
if (warunek) [rozkazy]. Jeżeli warunek jest spełniony to wykonują się rozkazy w nawiasach kwadratowych, jeżeli warunek nie jest już spełniony wykonywanie rozkazów kończy się. Wygląda to tak, jak w poniższym okienku.



Efekt poprawionej procedury wygląda następująco:



Po zmianie kąta z 90 na 120 ślimak rysuje się w trójkąt:



Na tym etapie programowania sam powinieneś czuć potrzebę wprowadzania kąta w momencie wywoływania procedury: **spirala *5 90***. Modyfikację istniejącej procedury pozostawiam Tobie. Jeżeli jeszcze masz z tym kłopot wróć do punktu 9 „Procedury z parametrem”.

Wprowadź do powyższej procedury przypadkową zmianę koloru pisaka. **SETPC n** – tylko za **n** wprowadź rozkaz **RANDOM 16** co oznacza wygenerowanie losowo liczby naturalnej mniejszej od 16. (zakres zdefiniowanych kolorów 0-15) W praktyce rozkaz wygląda tak: **SETPC (RANDOM 16)** powinieneś go wstawić wewnątrz nawiasów kwadratowych procedury, np.: po **RT 90**. Zapewne zauważyłeś, że niektórych odcinków nie widać - narysowane zostały na biało! Na to też można znaleźć sposób z wykorzystaniem instrukcji warunkowej **if**. POKOMBINUJ SAM!

12 Projekt złożony.

Zbierz wiadomości ze wszystkich poprzednich punktów i postaraj się stworzyć złożoną procedurę, która narysuje obrazek: płotek z czterema sztachetkami oraz wiatrak z obracającymi się skrzydłami. Powodzenia.