

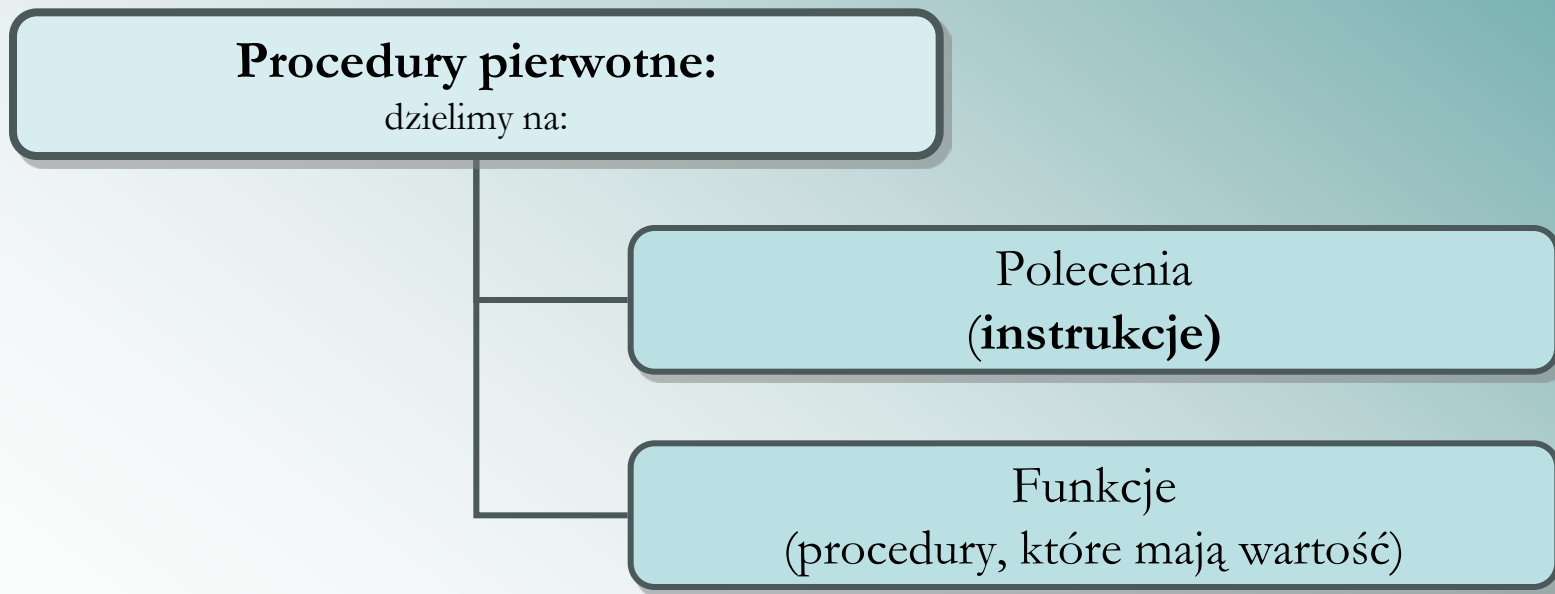
Logo Komeniusz

Gimnazjum w Tęgoborzy

Mgr Zofia Czech

Język Logo

to język strukturalny, umożliwiającą dzielenie algorytmu na wyraźnie wyodrębnione problemy, których rozwiązanie opisuje się za pomocą **procedur** (tzn. poleceń zrozumiałych dla tego języka).



Procedury pierwotne (polecenia):

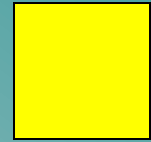
po wpisaniu procedury wciskamy klawisz *Enter*

- # **pż** – (pokaż żółwia) żółw pokazuje się na ekranie
- # **sż** – (schowaj żółwia) żółw przestaje być widoczny na ekranie
- # **naprzód *liczba_kroków*** lub **np *liczba_kroków***– przesunięcie żółwia do przodu o podaną liczbę kroków (np. np 100)
- # **wstecz *liczba_kroków*** lub **ws *liczba_kroków*** – przesunięcie żółwia do tyłu o podaną liczbę kroków (np. ws 70)
- # **prawo *kąt*** lub **pw *kąt***– obrót żółwia w prawo o podany w stopniach kąt (np. pw 90)
- # **lewo *kąt*** lub **lw *kąt***– obrót żółwia w lewo o podany w stopniach kąt (np. lw 45)
- # **podnieś** lub **pod** – powoduje podniesienie pisaka żółwia
- # **opuść** lub **opu** - powoduje opuszczenie pisaka żółwia
- # **zmaż** – wymazuje wszystkie rysunki z ekranu bez zmiany pozycji żółwia
- # **wróc** – ustawia żółwia w pozycji początkowej na środku ekranu
- # **czyść** lub **cs** - wymazuje wszystkie rysunki z ekranu i ustawia żółwia w pozycji początkowej
- # **UstalGrubośćPisaka *grubość*** lub **ugp *grubość*** -ustala jaką szerokość ma mieć „rysik” pisaka
- # **UstalKolorPisaka *nr_koloru*** lub **ukp *nr_koloru*** - ustala kolor pisaka, pod numerami od 1 do 15 są poszczególne kolory
- # **UstalTło *nr_koloru*** - ustala kolor tła
- # **zamaluj** - Należy podnieść pióro, ustawić się wewnątrz figury zamkniętej i opuścić pióro. Zamaluje (wypełni) wnętrze tej figury kolorem, który jest aktualnie ustalony (poleceniem ukp)
- # **powtórz *n [lista_poleceń]*** - powoduje n-krotne powtórzenie listy poleceń (np. powtórz 2 [np. 50 pw 90])

Rysowanie figur geometrycznych

Kwadrat o boku długości 100

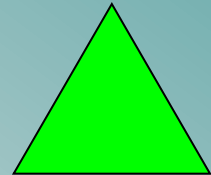
np 100 pw 90 np 100 pw 90 np 100 pw 90 np 100 pw 90



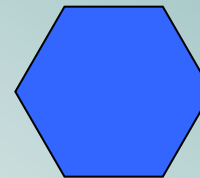
Prostokąt o bokach 200 na 100



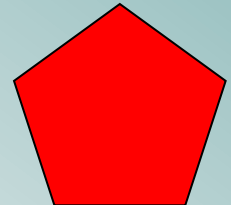
Trójkąt równoboczny o boku długości 150



Sześciokąt foremny o boku 40



Pięciokąt foremny o boku 70



Procedury bez PARAMETRÓW

Polecenia wydawane żółwiowi to tzw. **procedury**. Jedne z nich nie wymagają podawania parametrów, czyli dodatkowych informacji np. procedura **cs**, **sż**, **pż**, **pod**, itd. Jednak inne np. **lw**, **pw**, **np**, **ws**, **ugp**, **ukp** wymagają podania danych, są nimi odpowiednio: kąt, odległość na jaką przesuwamy, grubość, kolor, itp.

Można samemu zdefiniować jakąś procedurę!

Budowa każdej procedury:

oto *nazwa_procedury*

Treść_procedury

już

Np. procedura „rysująca” kwadrat:



oto *kwadrat*

Powtórz 4[*np 60 pw 90*]

już

Wywołanie procedury!

Wciśnięcie klawisza **F4** (menu Okno – Pokaż **pamięć**)
– powoduje wyświetlenie danej procedury (kwadrat)
wraz z jej zawartością.

Aby napisać kolejną, nową procedurę możemy
postępować jak wcześniej lub po otwarciu pamięci (**F4**)
- **Obiekty – Dodaj procedurę - ... – koniec**

W pamięci edytora poruszmy się jak po zwykłym
edytorze tekstowym, możemy wprowadzać zmiany
(dwa razy klikamy na nazwie procedury)

Aby zobaczyć czy procedura działa piszemy jej nazwę i
Enter (czyli w tym wypadku: *kwadrat* – *Enter*).

Ćwiczenia do samodzielnego wykonania:

Zdefiniuj procedury rysowania:

(boki długości 60)

- ☞ Trójkąta równobocznego
- ☞ Pięciokąta foremego
- ☞ Sześciokąta foremego
- ☞ Ośmiokąta foremego

Procedury z PARAMETRAMI

Do procedury „kwadrat” wprowadzimy *parametr*
– którym będzie liczba określająca długość boku
tego kwadratu

parametr formalny

oto *kwadrat* **:bok**

powtórz 4[*np* **:bok** *pw* 90]

już

parametr aktualny

wywołanie procedury:

kwadrat 150

Objaśnienia:

W chwili wywołania procedury w miejsce tzw. *parametru formalnego* (:bok) wstawiana jest jego wartość (np. 150) – tzw. *parametr aktualny*.

W ten sposób możemy wywołać tę samą procedurę z różnymi parametrami aktualnymi, czyli w tym przypadku - rysować kwadraty o różnych długościach boków, bez konieczności pisania nowej procedury.

Nowe pojęcia:

Parametry formalne

– to parametry procedury występujące w jej definicji (np. :bok, :ką, :a, :h, itp.)

Parametry aktualne

– to wartości podawane w chwili wywołania procedury, tzn. wartości dla których dana procedura ma być wykonana (np. 100, 90, 350, itd.)

Ćwiczenia:

I. Popraw procedury „rysujące”:

- ✓ trójkąt równoboczny
- ✓ pięciokąt foremny
- ✓ sześciokąt foremny
- ✓ ośmiokąt foremny

tak, by można było przy ich wywołaniu określić długość boku.

II. Zdefiniuj procedurę *figura*, służącą do rysowania wielokąta foremnego o dowolnej liczbie boków i dowolnej ich długości ([rozwiązanie](#))

Ćw. II.

oto *figura* **:n :bok**

powtórz **:n** [np **:bok** pw **360/:n**]

już

dwa parametry formalne:

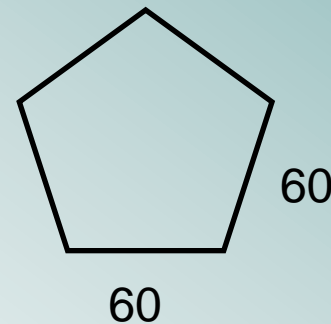
:n – ilość kątów wielokąta,

:bok – długość boku tego wielokąta

360/:n – kąt o jaki będzie się obracał żółw rysując „n-kąt”
foremny

Wywołanie: *figura 5 60* /enter

Efekt:



Kwadraty

oto **kwadrat** :bok

powtórz 4 [np :bok pw 90]

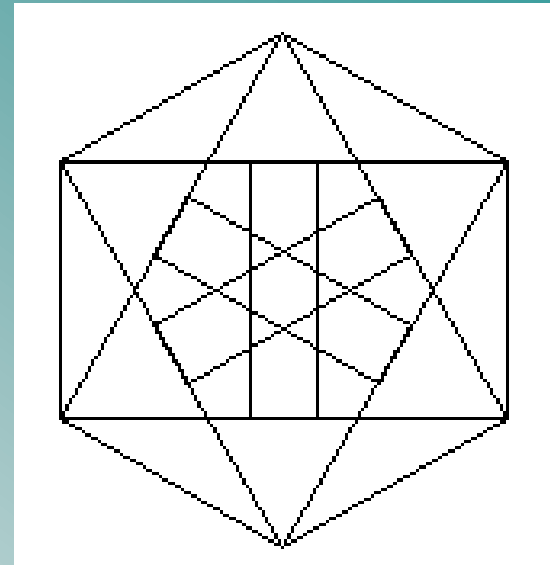
już

kwadraty6

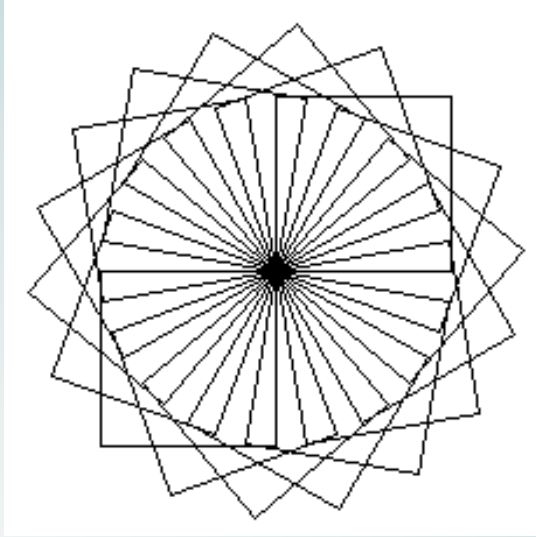
oto kwadraty6

powtórz 6 [**kwadrat** 80 np 80 pw 60]

już



Ornament z kwadratów



oto kwadrat :bok

powtórz 4 [np :bok pw 90]

już

oto kwadraty

cs

powtórz 18 [kwadrat 75 pw 20]

już

Pawie oczko

oto **okrag** :promień

powtórz 36 [np 0,175 * :promień pw 10]

już

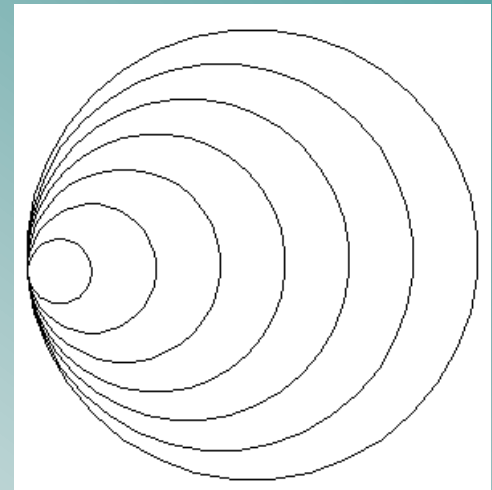
oto pawie_oczko

cs

przyypisz "promień 20

powtórz 7 [**okrag** :promień przyypisz "promień
:promień + 20]

już



Zmienne i funkcje w Logo

I. Tworzenie zmiennej i nadawanie jej wartości:

Polecenie:

przypisz "nazwa_zmiennej wartość

lub

przyp "nazwa_zmiennej wartość

Np.

przypisz "liczba 123

przyp "x :liczba+10

wartość

nazwa zmiennej

Uwaga!!!

"liczba – oznacza nazwę zmiennej

:liczba – oznacza wartość zmiennej o nazwie *liczba*

II. Odczytywanie wartości utworzonych zmiennych (x i liczba)

pisz :nazwa_zmiennej

czyli:

pisz :liczba /Enter 123

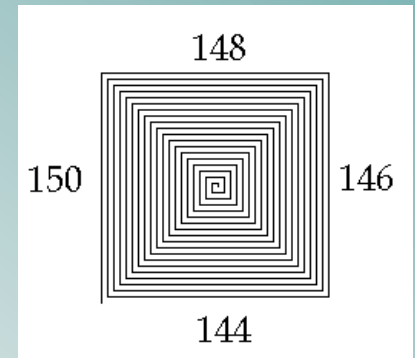
pisz :x /Enter 133

Ćw. 1.

Zmodyfikuj procedurę *figura*, by kąt obrotu żółwia nie był obliczany przy każdym powtórzeniu. Zmienną przechowującą wartość kąta obrotu żółwia ($360 / :n$) nazwij *obrót*.

Ćw. 2.

Zdefiniuj procedurę *figury*, służącą do rysowania wielokątów foremnych o dowolnej liczbie **boków** i **obwodzie**.



Ćw. 3.

Napisz procedurę rysowania spirali zwijającej się do środka, jak na rysunku obok

Rozwiązanie ćw. 1.

oto *figura* :n :bok

przypisz ”*obrót* 360/:n

powtórz :n [np :bok pw :*obrót*]

już



Rozwiązanie ćw. 2.

oto *figury* :n :obwód

przypisz ”*obrót* 360/:n

przypisz ”*bok* :obwód/:n

powtórz :n [np :**bok** pw :*obrót*]

już

dwa parametry (:n, :obwód)

dwie zmienne (”*obrót* 360/:n, ”*bok* :obwód/:n)



Rozwiązanie ćw. 3.

oto *spiralą*

przypisz "bok 150

powtórz 75 [np :bok pw 90 przypisz "bok :bok-2]

już



III. Funkcje

Funkcja – to procedura zwracająca wartość

Np. funkcja *sześcian*, której wartością jest sześcian podanej liczby:

oto *sześcian* :x

wynik :x*:x*:x

już

gdzie:

wynik *parametr*

(lub **wy** *parametr*) -- definiowanej funkcji przypisuje wartość
swojego parametru (nadaje funkcji wartość)

Wywołanie:

Np.: **pisz** *sześcian* 5 /Enter 125

Procedury rekurencyjne

Instrukcja warunkowa:

Jeśli *warunek* [*polecenie1*][*polecenie2*]

jak warunek jest spełniony, to wykonane jest polecenie1, w przeciwnym wypadku -polecenie2

Jeśli *warunek* [*polecenie*]

jak warunek jest spełniony, to wykonane jest polecenie, w przeciwnym wypadku polecenie jest pomijane

Ćw. 1. Napisz procedurę (funkcję) „większa”, w wyniku której zostanie wyznaczona większa z dwóch danych liczb

Rozwiązanie ćw. 1.

Oto *większa* :a :b

jeśli :a>:b [wynik :a][wynik :b]

już

wywołanie:

pisz *większa* 12 3 /Enter

efekt:

12



Procedura rekurencyjna – to procedura wywołująca samą siebie. Żeby nie wykonywała się w nieskończoność musi zawierać **warunek zakończenia działania**.

Np.:

oto *spiralarek* :bok

jeśli :bok \leq 0 [**stop**]

np :bok

pw 90

spiralarek :bok-2




już

warunek zakończenia
działania programu

odwołanie do samej siebie, ze
zmniejszonym parametrem

stop – powoduje zakończenie działania procedury!

Zadania

1. Losowe „błądzenie” żółwia po monitorze 
2. Spirala o dowolnie zmieniającym się boku i kącie 
3. Procedura obliczająca a^n – **funkcja** potęga 

Rozwiązanie zadania 1.

oto *błądzenie* :ile

jeśli :ile < 1 [**stop**]

np **losowa** 10

pw **losowa** 360

błądzenie :ile-1

już



Rozwiązanie zadania 2.

oto *spirala* :bok :ką :dodatek

jeśli :bok>100 [**stop**]

np :bok pw :ką

spirala :bok+:dodatek :ką :dodatek

już



Rozwiązanie zadania 3.

oto *potęga* :*a* :*n*

jeśli :*n*=0 [**wynik** 1]

wynik :*a***potęga* :*a* :*n*-1

już

