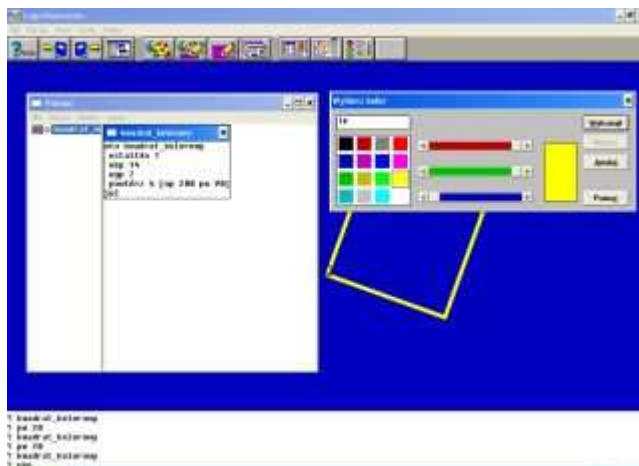


Programowanie - LOGO z programem Logo Komeniusz



Logo Komeniusz jest edukacyjnym językiem programowania i środowiskiem dydaktycznym. Oferuje użytkownikom różnorodne możliwości operacji przy pomocy myszy, menu, przycisków - wszystko z głównym bohaterem środowiska: **żółwiem**.

Wraz z głównym programem dostępny jest także **Edytor obrazów**, za pomocą którego w prosty graficzny sposób zaprojektować możemy postać żółwia statycznego lub animowanego.



Podstawowe polecenia dla żółwia i środowiska

<ul style="list-style-type: none">• pż - pokaż żółwia• sż - schowaj żółwia• cs - czyść ekran• odnowa - zresetuj ekran• pod - podnieś pisak• opu - opuść pisak• ścier - ścieraj	<ul style="list-style-type: none">• np n - idź n kroków• pw n - obróć żółwia w prawo o n stopni• lw n - obróć żółwia w lewo o n stopni• powtórz n - powtórz n razy• czekaj n - czekaj z wykonaniem n ms.	<ul style="list-style-type: none">• ustaltło - zmień kolor tła• ukp - ustal kolor pisaka• ugp - ustal grubość pisaka
---	---	---

Lista wszystkich poleceń dla żółwia i środowiska znajduje się w menu programu: **Pomoc->Lista procedur**

Podstawowa budowa programu Logo Komeniusz

Główny ekran programu Logo Komeniusz podzielony jest na dwie części. Mniejszy na dole - z napisem "Witaj w Logolandii" to **ekran tekstowy** do wprowadzania poleceń dla żółwia. Znak pytajnika - ? - oraz migający kursor zachęca do wprowadzania poleceń. Warto wpisać na początek polecenie **pż** - czyli **pokaż żółwia**.

Po wydaniu polecenia **pż** - w drugim większym ekranie pokaże się żółw :) Na tym **ekranie graficznym** więc

wykonywane są efekty pracy - jest to główne miejsce akcji.

Najważniejsze ikonki to:



Przykładowe programiki w LOGO

Programowanie w LOGO oraz sposób dochodzenia do właściwych rozwiązań prześledzimy na przykładzie **kwadratu**.

- Nasz cel to: **narysuj kwadrat o boku 200**

Wydając w ekranie tekstowym polecenie **np 200** (i potwierdzając ENTERem) żółw narysuje kreskę do góry o długości 200 umownych kroków. Następnie obracamy żółwia w prawo - **pw 90** (i ENTER). Żółw obróci się. W analogiczny sposób **np 200** oraz **pw 90** wykonujemy pozostałe boki - w efekcie na ekranie graficznym zobaczymy cały kwadrat o boku 200 kroków.

Analizując nasze postępowanie stwierdzimy, iż możemy od razu wydać żółwiowi zestaw poleceń potrzebnych do narysowania kwadratu. Oto przykład:

Przykład kodu

```
np 200 pw 90 np 200 pw 90 np 200 pw 90 np 200 pw 90
```

Zwróćmy uwagę, iż żółw cztery razy robi to samo - **np 200 pw 90**. Dlatego możemy kazać mu powtórzyć 4 razy wykonanie tych kroków:

Przykład kodu

```
powtórz 4 [np 200 pw 90]
```

Żółw narysował kwadrat - czy to oznacza, że on wie co to jest kwadrat? Wydajmy mu polecenie **kwadrat** - otrzymamy komunikat o błędzie. Nauczmy go więc, że to co narysował to jest właśnie kwadrat. Mówimy, że nauczmy go nowej procedury - w tym wypadku procedury o nazwie **kwadrat**:

Przykład kodu

```
oto kwadrat  
powtórz 4 [np 200 pw 90]  
już
```

Teraz wydajmy polecenie **kwadrat** - żółw narysuje to co trzeba. To co się nauczył przechowuje w swojej pamięci - do której mamy dostęp klikając odpowiednią ikonkę programu (patrz wyżej w Najważniejsze ikonki) W nowym oknie zobaczymy pamięć żółwia z zapamiętaną procedurą **kwadrat**.

Ponieważ nasz kwadrat ma stałą długość boku 200 kroków - zmodyfikujemy naszą procedurę (w pamięci) stosując zmienną **a**. Oto kod zmienionej procedury:

Przykład kodu

```
oto kwadrat :a
powtórz 4 [np :a pw 90]
już
```

Teraz wydając polecenie **kwadrat 150** lub **kwadrat 50** (lub z inną wartością) otrzymamy odpowiednią długość boku.

Poniższy kod prezentuje bardziej kolorowy i przestrzenny kwadrat. Warto przeanalizować zastosowane nowe polecenia - według pomocy w: [Podstawowe polecenia dla żółwia i środowiska](#).

Warto zauważyć, iż zdefiniowana jest tutaj nowa procedura o nazwie **kwadrat_kolorowy**.

Przykład kodu - kolorowy kwadrat

```
oto kwadrat_kolorowy
odnowa
ustaltło 1
ukp 14
ugp 7
powtórz 4 [np 200 pw 90]
już
```

Oto kolejny przykład rysujący kwadratowy sekundnik:

Przykład kodu - kwadratowy stoper

```
oto kwadratowy_stoper
powtórz 60 [ kwadrat pw 6 czekaj 1000 ]
już
```

• Inne przykłady

Przykład kodu - sekundnik na tarczy

```
oto sekundnik
powtórz 60 [sz opu np 200 pw 180 czekaj 1000 ścier np 200 pw 186]
już

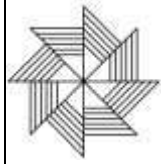
oto tarcza
np 220 pw 90 powtórz 360 [np 4 pw 1] pw 90 np 220 pw 180
już

oto zegar
tarcza sekundnik
już
```

Przykład kodu - trójkąt

```
oto trojkat
pw 90
np 200
lw 90
np 200
lw 135
np 200 * pwk 2
już
```

Przykład kodu - trójkąt, wiatrak, wiatraki



```
oto trojkat :a
przy "b :a * pwk 2
pw 90
np :a
lw 90
np :a
```

```
lw 135
np :b
już
```

```
oto wiatrak :a
powtór 8 [trojkat :a]
już
```

```
oto wiecej :c
jeśli :c > 200 [stop]
wiatrak :c
czekaj 1000
wiecej :c + 20
już
```

Przykład kodu - trójkąt z komentarzem

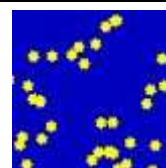
```
oto trojkat_kom
ts zt
ps [mamy narysować trójkąt prostokątny]
ps [długości przyprostokątnych są sobie równe i wynoszą założmy 200 kroków]
ps [aby obliczyć przeciwprostokątną - korzystamy ze wzoru:]
ps [200 * pierwiastek kwadratowy z 2]
ps [- - - - -]
czekaj 6000
ps [pierwiastek kwadratowy - to polecenie pwk]
ps "czyli ps [przeciwprostokątna to:] powtór 3 [wpisz char 32] ps [200 * pwk 2]
ps [- - - - -]
czekaj 6000
ps "czyli ps [nasz trójkąt wygląda tak:] powtór 3 [wpisz char 32] ps [pw 90 np 200 lw 90 np 200 lw 135 np 200 * pwk 2]
czekaj 6000
dzielekran
czekaj 1000 pw 90 czekaj 1000 np 200 czekaj 1000 lw 90 czekaj 1000 np 200 czekaj 1000 lw 135 czekaj 1000 np 200 * pwk 2
już
```

Przykład kodu - okręgi i kule

```
oto kule
powtór 9 [powtór 360 [np 2 pw 1] czekaj 300 np 30 pw 40]
już
```

Przykład kodu - romb, gwiazdka, losowe gwiazdki

```
powtór 2 [ np 70 pw 60 np 70 pw 120]
powtór 6 [powtór 2 [ np 70 pw 60 np 70 pw 120] pw 60]
powtór 6 [ romb pw 60 ]
oto awiazdki :ile :rozzut
```



```
oto romb
już
oto gwiazda
już
lub
oto gwiazda
już
```

```
przy "losowo losowa :rozzut pw losowa 360
jeśli :ile = 0 [stop]
pod
np :losowo
opu
gwiazda
czekaj 100
gwiazdki :ile - 1 :rozzut
już
```