



Grażyna Koba

Algorytmika i programowanie

Programowanie w języku Logo

materiały dodatkowe do podręcznika
Informatyka dla gimnazjum

MiGra

1. Proste polecenia języka Logo
2. Wykonywanie obliczeń i stosowanie zmiennych
3. Powtarzanie poleceń
4. Stosowanie podprogramów
 - 4.1. Procedury bez parametrów
 - 4.2. Procedury z parametrami
5. Sytuacje warunkowe



Warto powtórzyć

1. Jakie poznaliśmy sposoby prezentacji algorytmów?
2. Czym jest program komputerowy?
3. Czym jest język programowania?
4. Czym charakteryzuje się interpretacja programu?
5. Kiedy mamy do czynienia z sytuacją warunkową?
6. Na czym polega iteracja?

1. Proste polecenia języka Logo

Język Logo jest edukacyjnym językiem programowania. Do tworzenia programów w tym języku będziemy wykorzystywać środowisko Logomocja. Okno programu Logomocja (rys. 1.) podzielone jest na dwa ekrany: **graficzny** i **tekstowy**.

Polecenia języka Logo wpisujemy w **wierszu poleceń** na ekranie tekstowym po znaku zachęty „?” i zatwierdzamy naciśnięciem klawisza **Enter**. Każde polecenie jest od razu sprawdzane przez komputer. Jeśli jest poprawne, zostaje wykonane i efekt widoczny jest w oknie programu na ekranie graficznym. Jeśli nie jest poprawne, program informuje nas o tym, wypisując odpowiedni komunikat. Jest to przykład **interpretacji programu**. Możemy powiedzieć, że „rozmawiamy z komputerem”.

Jedną z najciekawszych cech języka Logo jest możliwość tworzenia rysunków poprzez sterowanie tzw. **żółwiem**. Na przykład:

- **naprzód** *liczba kroków* (w skrócie **np**) przesuwa żółwia do przodu o podaną liczbę kroków, czyli **np 200** przesuwa go o 200 kroków,
- **prawo** *kąt* (w skrócie **pw**) obraca żółwia w prawo o podany kąt, czyli **pw 90** obraca go o 90°.

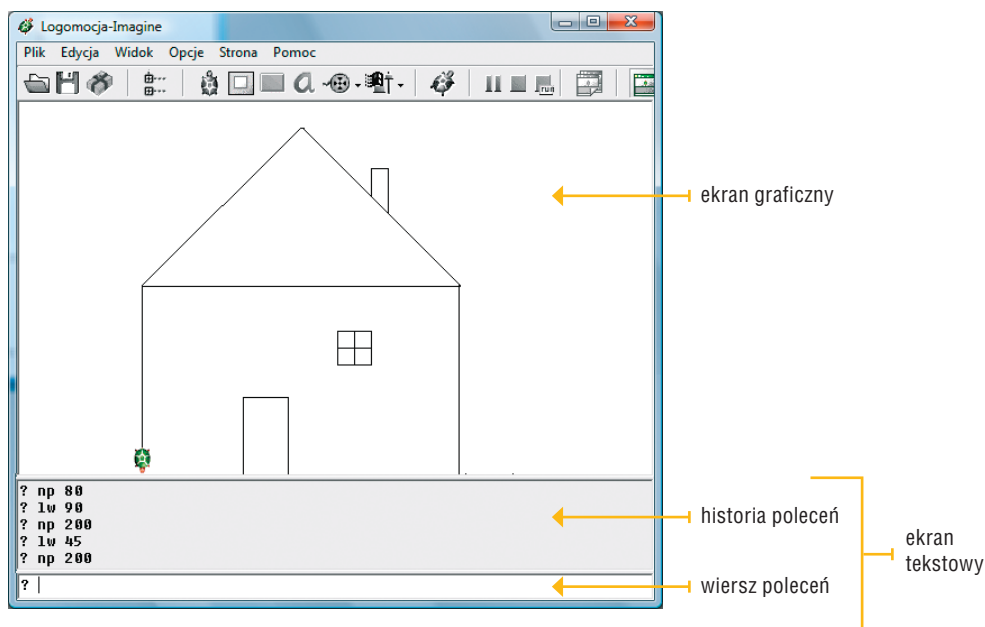
Jeśli nie wiemy, o ile kroków należy przesunąć żółwia, możemy skorzystać z **linijki** (rys. 2.), która pokazuje się po wpisaniu odpowiedniej komendy bez podawania liczby



Uwaga

Opis podstawowych poleceń języka Logo znajdziesz w pliku *Polecenia Logo.pdf* (CD).

kroków, np. komendy **naprzód (np)**. Po wskazaniu kursorem myszy miejsca, do którego ma dojść żółw, program sam uzupełni potrzebną liczbę kroków.



Rys. 1. Okno programu Logomocja



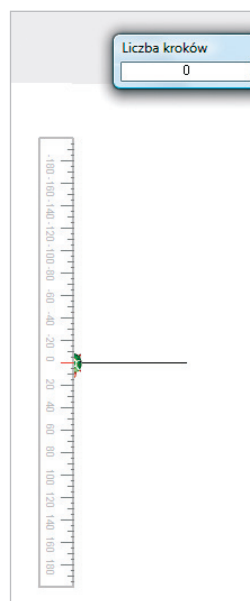
Ćwiczenie 1. Piszemy polecenia w języku Logo

W części tekstowej okna programu Logomocja napisz kolejno polecenia:

pż
np 100
pw 135
np 45
lw 90
np 45
pw 135
np 100

Kilka praktycznych porad

- Wprowadzane przez nas polecenia muszą być zgodne z regułami języka Logo. Np. pomiędzy nazwą (skrótom nazwy) polecenia a wartością liczbową powinna pojawić się spacja. W przeciwnym przypadku polecenie nie zostanie wykonane – pojawi się komunikat o błędzie. Na przykład po napisaniu **np200** pojawi się napis *Błąd: Nie wiem, jak wykonać np200*. Należy napisać polecenie poprawnie, czyli ze spacją po nazwie polecenia: **np 200**.



Rys. 2. Linijka w programie Logomocja

- Pisząc program w języku Logo, najwygodniej jest stosować skróty nazw poleceń.
- Polecenia napisane wcześniej można wyświetlać, używając klawiszy sterujących (strzałka w górę, strzałka w dół).
- Do wiersza poleceń można wkleić (**Ctrl+V**) polecenie skopiowane (**Ctrl+C**) wcześniej do **Schowka**.



Ćwiczenie 2. Rysujemy dom

Narysuj w programie Logomocja dom podobny do pokazanego na rysunku 1.

2. Wykonywanie obliczeń i stosowanie zmiennych

W języku Logo można również stosować polecenia, które spowodują wykonanie obliczeń lub wyprowadzenie napisu. Wyniki obliczeń i napisy pojawiają się na ekranie tekstowym.



Aby w języku Logo wyświetlić wartość wyrażenia na ekranie tekstowym, należy napisać polecenie **ps wyrażenie.**

Na przykład polecenie: **ps** 345+234 spowoduje wypisanie w wierszu poleceń liczby 579, a **ps** 234/2 – liczby 117.



Ćwiczenie 3. Wykonujemy działania arytmetyczne

Wykonaj działania w języku Logo:

- 2345/324,
- $(234+2725)*0.5$,
- $(2642-97)/65$.

W obliczeniach można używać również zmiennych. Przed użyciem zmiennej należy przypisać jej wartość, stosując polecenie **przypisz nazwa wartość** (w skrócie **przyp**), które tworzy zmienną o podanej nazwie i przypisuje jej wartość podaną jako drugi parametr. Przed nazwą tej zmiennej należy umieścić znak górnego cudzysłowu, natomiast przed nazwą zmiennej używanej w *wyrażeniu* – znak dwukropka.

Na przykład:

przyp "a 25.4 przypisuje zmiennej *a* wartość 25,4 (w języku Logo do wydzielenia części ułamkowej liczby używamy kropki, a nie przecinka),
przyp "b 440 przypisuje zmiennej *b* wartość 440,
przyp "s :a+:b zmiennej *s* przypisuje wartość sumy liczb przypisanych do zmiennych *a* i *b*,
ps :s wypisuje wartość zmiennej *s*.

Uwaga: Dwa ostatnie polecenia, powodujące obliczenie i wyświetlenie wartości sumy *a* i *b*, możemy zapisać również jako jedno: **ps :a+:b**.



Ćwiczenie 4. Wykonujemy obliczenia z użyciem zmiennych

1. Pisząc odpowiednie polecenia w języku Logo, oblicz dla a równego 20 i b równego 50:
 - a. sumę liczb a i b ,
 - b. pole prostokąta o bokach a i b ,
 - c. pole kwadratu o boku a ,
 - d. pole kwadratu o boku b .
2. Wykonaj te same obliczenia dla danych: $a = 1,5$ i $b = 2,5$.

Polecenia **ps** możemy również używać do wypisywania w języku Logo ciągu słów, tzw. **listy**, np. **ps** [lubię informatykę] wypisze w wierszu poleceń słowa *lubię informatykę*. Ciąg słów należy umieścić w nawiasach kwadratowych.

3. Powtarzanie poleceń

W języku Logo istnieje polecenie **powtórz**, które umożliwia powtarzanie ciągu poleceń określoną liczbę razy. Korzystając z tego polecenia, możemy zapisywać m.in. algorytmy iteracyjne.



Aby powtarzać operacje w języku Logo, należy użyć polecenia **powtórz**, które ma postać:
powtórz *liczba_powtórzeń* [*lista poleceń*].

Na przykład:

powtórz 8 [**np** 50 **lw** 45] narysuje ośmiokąt foremny o boku 50 kroków.



Ćwiczenie 5. Realizujemy powtórzenia (iteracje)

Korzystając z polecenia **powtórz**, narysuj ośmiokąt foremny o boku 40 kroków.

4. Stosowanie podprogramów

Podprogram **P**
 (procedura lub funkcja)

Wyodrębniona część programu, mająca jednoznaczną nazwę i ustalony sposób wymiany danych z innymi częściami programu.

Podprogram, realizując określone zadanie, może być wielokrotnie wykorzystywany w programie.

Kroki, które powtarzają się w algorytmie, można potraktować jako problem cząstkowy i przedstawić w postaci **podprogramu** (procedury lub funkcji). Umożliwia to opracowanie każdego problemu cząstkowego oddzielnie.

Aby napisać cały program, wystarczy zestawić podprogramy i odpowiednio je połączyć. Połączenia podprogramu z programem głównym realizuje się poprzez instrukcję wywołania podprogramu.

W Logo możemy korzystać ze standardowych procedur tego języka bądź definiować własne. Za pomocą procedury będziemy określać ciąg czynności, które powinien wykonać komputer. Tak zdefiniowane procedury będziemy następnie wykorzystywać w innych programach.

4.1. Procedury bez parametrów

Zdefiniujemy procedurę rysującą ośmiokąt. W oknie pamięci (klawisz **F4** lub odpowiednia opcja menu) można zobaczyć zdefiniowane procedury. Można je również w tym oknie edytować (klawisz **F11** lub odpowiednia opcja menu).



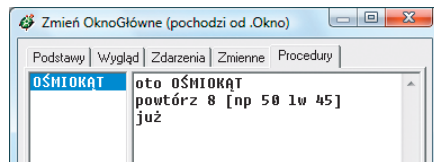
Aby zdefiniować procedurę bez parametrów, należy wpisać

oto *nazwa_procedury*
treść procedury
już.

Na przykład po napisaniu:

oto OŚMIOKĄT
powtórz 8 [np 50 lw 45]
już

zostanie zdefiniowana procedura OŚMIOKĄT.



Rys. 3. Widoczna w oknie pamięci zdefiniowana procedura OŚMIOKĄT

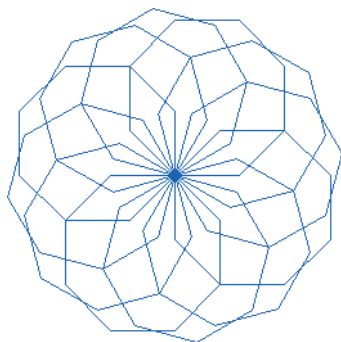


Ćwiczenie 6. Definiujemy procedurę bez parametrów

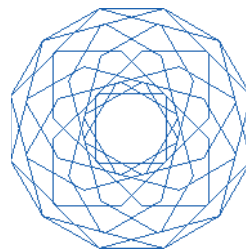
1. Zdefiniuj procedurę bez parametrów OŚMIOKĄT (rys. 3.).
2. Przed poleceniem **powtórz** dodaj: **ukp** "niebieski" (zmiana koloru pisaka na niebieski).

Aby wywołać procedurę, piszemy jej nazwę po znaku zachęty „?”. Możemy teraz używać zdefiniowanej procedury w definicjach innych procedur czy w poleceniach, np. **powtórz**. Na przykład, używając procedury OŚMIOKĄT, możemy otrzymać obraz przedstawiony na rysunku 4:

oto OŚMIOKĄTY1
powtórz 12 [OŚMIOKĄT lw 30]
już



Rys. 4. Wykonanie procedury OŚMIOKĄTY1



Rys. 5. Wykonanie procedury OŚMIOKĄTY2



Ćwiczenie 7. Korzystamy z utworzonych wcześniej procedur

1. Korzystając z procedury OŚMIOKĄT, zdefiniuj procedurę OŚMIOKĄTY1 (rys. 4.).
2. Korzystając z procedury OŚMIOKĄTY1, utwórz procedurę OŚMIOKĄTY2, rysującą figurę pokazaną na rysunku 5.

Wskazówka: Do treści procedury OŚMIOKĄTY1 należy dodać w odpowiednim miejscu polecenie **np 50**.

Aby w programie Logomocja zapisać treść zdefiniowanych procedur, należy zapisać je w pliku, tworząc tzw. projekt (opcja **Zapisz projekt**). W skład projektu wchodzi m.in. aktualna treść ekranu graficznego, zdefiniowane procedury i własne żółwie. Można też wydrukować ekran graficzny (opcja **Drukuj**).

Uwaga



W wersji demonstracyjnej programu Logomocja (CD) nie ma możliwości zapisu projektu w pliku.



Ćwiczenie 8. Zapisujemy projekt i drukujemy ekran graficzny

Zapisz pracę utworzoną w ćwiczeniu 7. w pliku pod nazwą *Projekt1*. Wydrukuj ekran graficzny z wybranym rysunkiem.

4.2. Procedury z parametrami



Aby zdefiniować procedurę z parametrami,
należy wpisać:

```
oto nazwa_procedury :parametr1 :parametr2
treść procedury
już
```

Parametry



Dane przekazywane do procedury, dzięki którym program główny może szczegółowo określić sposób działania procedury.

Procedura może zawierać dowolną liczbę parametrów. Należy pamiętać o umieszczeniu znaku „:” przed nazwą każdego parametru.

Na przykład:

```
oto SZEŚCIOKĄT :N
powtórz 6 [np :N lw 60]
już
```

definiuje procedurę rysowania sześciokąta foremnego o boku długości *N* kroków.

Procedurę wywołujemy, podając konkretną wartość parametru *N*, np. SZEŚCIOKĄT 100.



Ćwiczenie 9. Definiujemy procedurę iteracyjną z parametrami

1. Zdefiniuj procedurę SZEŚCIOKĄT z parametrem N , rysującą sześciokąt foremny (N – długość boku sześciokąta).
2. Przed poleceniem **powtórz** dodaj: **ukp** "czerwony" (zmiana koloru pisaka na czerwony).
3. Wywołaj procedurę SZEŚCIOKĄT dla kilku różnych długości boku.

Chcemy narysować N takich samych trójkątów, które będą przesunięte o podaną liczbę kroków i obrócone o podany kąt. Przygotujemy najpierw procedurę TRÓJKĄT, rysującą pojedynczy trójkąt.



Ćwiczenie 10. Definiujemy procedurę rysującą trójkąt równoboczny

Zdefiniuj procedurę TRÓJKĄT z parametrem A , rysującą trójkąt równoboczny (A – długość boku trójkąta).

W celu narysowania N trójkątów należy, używając polecenia **powtórz**, powtórzyć N razy rysowanie trójkąta oraz jego przesuwanie i obracanie.



Ćwiczenie 11. Definiujemy procedurę rysującą trójkąty

1. Zdefiniuj procedurę iteracyjną TRÓJKĄTY z parametrem N , gdzie N to liczba trójkątów. Trójkąt ma być przesuwany o 10 kroków i obracany w lewo o kąt 35° . Przyjmij bok trójkąta (parametr aktualny) równy 100.
2. Wywołaj procedurę TRÓJKĄTY z parametrem $N = 40$.

5. Sytuacje warunkowe



W języku Logo polecenie realizujące sytuację warunkową ma postać:

jeżeli warunek [*lista poleceń 1*] [*lista poleceń 2*]

Jeśli *warunek* jest spełniony, to realizowane są polecenia *lista poleceń 1*, w przeciwnym przypadku realizowane są polecenia *lista poleceń 2*.

Na przykład:

oto FIGURA :X

jeżeli :X> 0 [OŚMIOKĄT] [TRÓJKĄT 80]

już

Gdy wywołamy procedurę FIGURA, podając jako parametr dowolną liczbę dodatnią, zostanie narysowany ośmiokąt foremny. Gdy jako parametr podamy zero lub liczbę ujemną, zostanie narysowany trójkąt równoboczny o boku 80.



Ćwiczenie 12. Stosujemy polecenia warunkowe w języku Logo

Zdefiniuj procedurę z parametrem X . Dla X dodatnich ma być wywołana procedura OŚMIOKĄTY1, a dla X niedodatnich – procedura OŚMIOKĄTY2. Procedury OŚMIOKĄTY1 i OŚMIOKĄTY2 zostały zdefiniowane w ćwiczeniu 7.



Warto zapamiętać

- W języku Logo możemy pracować w trybie interakcyjnym (inaczej: w trybie dialogu z komputerem).
- W języku Logo w obliczeniach można używać zmiennych. Przed użyciem zmiennej należy przypisać jej wartość, stosując polecenie **przypisz**.
- W języku Logo możemy zapisywać m.in. algorytmy iteracyjne. Do ich zapisu wykorzystujemy polecenie **powtórz**.
- Procedury (podprogramy) to wyodrębnione części programu, mające jednoznaczną nazwę i ustalony sposób wymiany danych z innymi częściami programu.
- W języku Logo możemy zdefiniować procedury bez parametrów i z parametrami. Definiując procedurę, określamy parametry, które w momencie jej wykonywania przyjmują konkretne wartości.
- W języku Logo możemy zapisywać m.in. algorytmy z warunkami. Do ich zapisu możemy zastosować polecenie **jeżeli**.



Pytania i polecenia

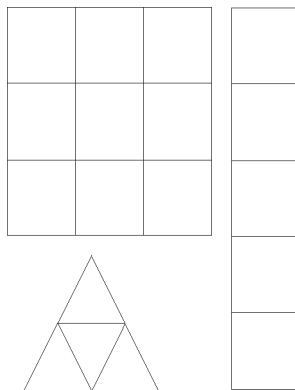
1. Wyjaśnij, co oznacza stwierdzenie: „Logo należy do języków interpretowanych”.
2. Wyjaśnij, dlaczego język Logo określany jest jako grafika żółwia.
3. W jaki sposób korzysta się ze zmiennych w obliczeniach? Omów na przykładzie.
4. W jaki sposób realizuje się powtórzenia w języku Logo?
5. Czym są podprogramy? Jakie są zalety ich stosowania?
6. Wyjaśnij na przykładzie, w jaki sposób definiuje się procedury w języku Logo. Rozróżnij procedury z parametrami i bez parametrów.
7. Omów sposób realizowania sytuacji warunkowych w języku Logo.



Zadania

1. Korzystając z poleceń języka Logo, narysuj swoje inicjały.
2. Napisz polecenia w języku Logo, aby obliczyć dla $a = 12$, $b = 15$ i $c = 10$: pole kwadratu Pa o boku a , pole kwadratu Pb o boku b , pole kwadratu Pc o boku c , sumę S pól kwadratów Pa , Pb , Pc .
Wykonaj te same obliczenia dla danych: $a = 3$, $b = 4$ i $c = 5$. Co zauważasz?
3. Zdefiniuj procedurę KWADRAT z jednym parametrem A (A – bok kwadratu). Zapisz projekt w pliku.
4. Zdefiniuj procedurę PROSTOKĄT z dwoma parametrami A i B (A , B – boki prostokąta). Zapisz projekt w pliku.
5. Korzystając z polecenia **powtórz**, narysuj sześciokąt foremny o boku 50.
6. Korzystając z wcześniej zdefiniowanych procedur, utwórz obraz przedstawiony na rysunku 6.
7. Utwórz samodzielnie kilka rysunków, korzystając z procedur zdefiniowanych w ćwiczeniach i zadaniach z tego tematu. Wydrukuj wykonane rysunki.
8. Zdefiniuj procedurę PIĘCIOKĄT z parametrem A , rysującą pięciokąt foremny o boku A . Zapisz projekt w pliku.
9. Wykorzystując procedurę PIĘCIOKĄT zdefiniowaną w zadaniu 8, przygotuj rysunki podobne do przedstawionych na ilustracjach 4. i 5. Zapisz projekt w pliku.

10. Zdefiniuj procedurę z parametrem Y . Dla $Y > 100$ ma być wywołana procedura KWADRAT (utworzona w zadaniu 3.), a w przeciwnym przypadku – procedura PROSTOKĄT (utworzona w zadaniu 4.). Zapisz projekt w pliku.



Rys. 6. Rysunek do zadania 6.

Dla zainteresowanych

11. Zdefiniuj procedurę, która umożliwi narysowanie dowolnego wielokąta foremnego. Powinna ona posiadać dwa parametry: długość boku (A) i liczbę boków (*liczba*). Zapisz projekt w pliku.
Wskazówka: Kąt obrotu żółwia w definicji procedury zapisz jako wyrażenie $360/\text{liczba}$.
12. Narysuj w Edytorze postaci własną postać animowaną, którą zastąpisz standardowego żółwia. Sprawdź, czy nowa postać narysuje trójkąt – wywołaj procedurę TRÓJKĄTY (utworzoną w ćwiczeniu 11.).
Wskazówka: Edytor postaci jest częścią programu Logomocja.
13. Zapoznaj się z działaniem funkcji **losowa** i wypisz za jej pomocą dwadzieścia liczb naturalnych o wartościach z przedziału $<1; 6>$, symulując wyniki rzutu kostką do gry.
14. Zapoznaj się z materiałami dodatkowymi (CD, folder *Dodatkowe/Algorytmika*), dotyczącymi m.in. umieszczania większej liczby żółwi na ekranie, zamieniania żółwia na inną postać, dodawania tła, umieszczania przycisków poleceń na ekranie. Przygotuj animowany projekt według własnego pomysłu.



Czy wiesz, że:

... bardziej zaawansowane drukarki, zwłaszcza laserowe, potrafią wykonywać polecenia specjalnego języka programowania PostScript? Język ten służy do opisywania wyglądu dokumentów i jest także wykorzystywany do zapisywania dokumentów w formacie PDF.