

1. Proste polecenia języka Logo
2. Wykonywanie obliczeń i stosowanie zmiennych
3. Powtarzanie poleceń
4. Stosowanie procedur
  - 4.1. Procedury bez parametrów
  - 4.2. Procedury z parametrami
5. Sytuacje warunkowe
6. Zmianianie postaci żółwia
  - 6.1. Stosowanie gotowych postaci
  - 6.2. Korzystanie z Edytora postaci
  - 6.3. Tworzenie animacji w Edytorze postaci



#### Warto powtórzyć

1. Jakże poznaliśmy sposoby przedstawiania algorytmów?
2. Czym jest język programowania?
3. Czym charakteryzuje się interpretacja programu?
4. Czym jest zmienna w programie komputerowym? W jaki sposób przypisywaliśmy wartość zmiennym w języku Scratch?
5. Czym jest podprogram? Co należy zrobić, aby zastosować podprogram w programie?
6. Czym jest parametr w procedurze?
7. Kiedy mamy do czynienia z sytuacją warunkową?
8. Na czym polega iteracja? W jaki sposób zapisywaliśmy powtarzające się polecenia w języku Scratch?

### 1. Proste polecenia języka Logo

Język Logo jest edukacyjnym językiem programowania. Do tworzenia programów w tym języku będziemy wykorzystywać środowisko programowania Logomocja, umożliwiające przygotowywanie projektów, w skład których mogą wchodzić m.in. utworzone rysunki (aktualna treść ekranu graficznego), dodane postacie żółwi oraz procedury.

Okno programu Logomocja (rys. 1.) jest podzielone na dwa ekrany: **graficzny** i **tekstowy**. Polecenia języka Logo wpisujemy w **wierszu poleceń** na ekranie tekstowym, po znaku zachęty „?” i zatwierdzamy naciśnięciem klawisza **Enter**. Komputer od razu sprawdza każde polecenie. Jeśli jest ono poprawne, zostaje wykonane. Efekt wykonania polecenia możemy zobaczyć w oknie programu na ekranie graficznym. Jeśli polecenie jest niepoprawne, program informuje nas o tym za pomocą odpowiedniego komunikatu. Jest to przykład **interpretacji programu**. Możemy powiedzieć, że „rozmawiamy z komputerem”.

## T: Programowanie w języku Logo – część 1

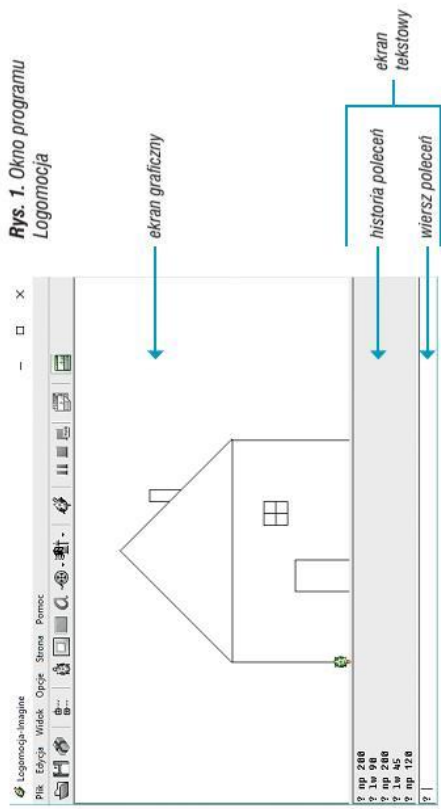
W związku z niedogodnościami i brakiem możliwości korzystania z programu Scratch i Logomocja te i kolejne zajęcia odbywać się będą na zasadzie odczytania przez uczniów treści z podręcznika.

**Jeśli będą jakiegokolwiek pytania odnoszące się do rozumienia poszczególnych pojęć proszę o kontakt poprzez adres strony klasy lub na Messengera.**

Dla tych, którzy zostawili podręcznik w szkole:

Jedną z cech języka Logo jest możliwość tworzenia rysunków poprzez sterowanie żółwem. Żółw, przesuając się, zostawia za sobą ślad w postaci linii. 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°.



Rys. 1. Okno programu Logomocja

Tabela 1. Podstawowe polecenia języka Logo

Polecenie	Skrót polecenia	Opis
<b>naprzód liczba_kroków</b>	<b>np liczba_kroków</b>	Przesuwa żółwia do przodu o podaną liczbę kroków.
<b>wstecz liczba_kroków</b>	<b>ws liczba_kroków</b>	Przesuwa żółwia do tyłu o podaną liczbę kroków.
<b>lewo kąt</b>	<b>lw kąt</b>	Obraca żółwia w lewo o podany kąt.
<b>prawo kąt</b>	<b>pw kąt</b>	Obraca żółwia w prawo o podany kąt.
<b>podnieś</b>	<b>pod</b>	Podnosi pisak żółwia (żółw nie będzie zostawiał śladów przy poruszaniu się).
<b>opuść</b>	<b>opu</b>	Opuszcza pisak żółwia.
<b>schowajMnie</b>	<b>sż</b>	Ukrywa żółwia.
<b>pokażMnie</b>	<b>pż</b>	Pokazuje żółwia.
<b>powtórz liczba_powtórzeń /lista_poleceń]</b>	brak skrótu	Powtarza listę poleceń podaną liczbę razy.
<b>pisz wyrażenie</b>	<b>ps wyrażenie</b>	Wypisuje wartość wyrażenia na ekranie tekstowym; w razie potrzeby wartość ta jest obliczana.

<b>przypisz nazwa wartość</b>	<b>przypisz nazwa wartość</b>	Tworzy zmienną o podanej nazwie <i>nazwa</i> i przypisuje jej wartość podaną jako drugi parametr.
<b>kolo średnica</b>	brak skrótu	Rysuje koło o podanej średnicy.
<b>okrąg średnica</b>	brak skrótu	Rysuje okrąg o podanej średnicy.
<b>czyszć</b>	<b>cs</b>	Czyści ekran graficzny.
<b>czekaj liczba_milisekund</b>	brak skrótu	Zatrzymuje wykonanie programu na podaną liczbę milisekund (np. <b>czekaj 7000</b> zatrzymuje wykonanie programu na 7 sekund).
<b>ustalKolorPis</b> <b>ustalGrubość</b>	<b>ukp</b> <b>ugp</b>	<b>ukp</b> – ustala kolor pisaka. <b>ugp</b> – ustala grubość pisaka. Polecenia <b>ukp</b> i <b>ugp</b> należy wpisać przed poleceniami rysowania figury. Po wpisaniu każdego z poleceń otwiera się okno, w którym wybiera się odpowiednio kolor lub grubość linii. Polecenie <b>ukp</b> można zapisać również z nazwą koloru poprzedzoną cudzysłowem, np. <b>ukp "niebieski</b> , <b>ukp "czerwonny</b> , a polecenie <b>ugp</b> – z liczbą określającą grubość linii, np. <b>ugp 5</b> , <b>ugp 20</b> .
<b>ustalKolorMal</b>	<b>ukm</b>	Ustala kolor malowania, np. <b>ukm "żółty</b> zmienia kolor na żółty.
<b>zamaluj</b>	brak skrótu	Zamalowuje obszar zamknięty bieżącym kolorem lub zmienia kolor linii na bieżący kolor, np. <b>ukm "żółty zamaluj</b> zamaluje obszar (jeśli żółw jest umieszczony w obszarze zamkniętym) lub linie na żółto.

Utworzony rysunek (ekran graficzny) możemy zapisać w pliku, korzystając z opcji **Plik/Zapisz projekt jako** lub **Plik/Zapisz projekt**. Projekty tworzone w Logomocji zapisywane są z rozszerzeniem *imp* (polecenia umieszczone w historii poleceń nie są zapisywane w projekcie). Ekran graficzny można wydrukować (opcja **Plik/Drukuj**).

**Uwaga**

Historię poleceń można zachować. W tym celu należy ją skopiować i wkleić do dokumentu tekstowego, a następnie zapisać dokument w pliku. Po ponownym uruchomieniu programu Logomocja wystarczy wkleić polecenia do **wiersza poleceń** i po naciśnięciu klawisza **Enter** zostaną one wykonane.



**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**





### Ćwiczenie 3. Wykonujemy działania

Wykonaj działania w języku Logo:

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

W obliczeniach można również używać 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. Wartość może być podana w formie wyrażenia. Przed nazwą 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** przypisuje zmiennej *s* wartość sumy liczb przypisanych do zmiennych *a* i *b*,

**ps :s** wypisuje wartość zmiennej *s* na ekranie tekstowym.

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



### Ćwiczenie 4. Wykonujemy obliczenia z użyciem zmiennych

- Pisząc odpowiednie polecenia w języku Logo, oblicz dla *a* równego 20 i *b* równego 50:
  - sumę liczb *a* i *b*,
  - pole powierzchni prostokąta o bokach *a* i *b*,
  - pole powierzchni kwadratu o boku *a*,
  - pole powierzchni kwadratu o boku *b*.
- Wykonaj te same obliczenia dla danych:  $a = 1.5$  i  $b = 2.5$ . Wykorzystaj wcześniej wpisane polecenia.

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

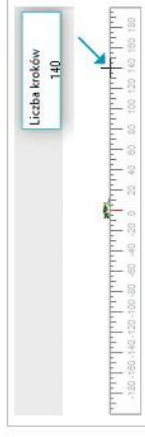
Zamierzamy wyświetlić na ekranie kompozycję składającą się z ośmiokątów foremnych. Aby powstał ośmiokąt, żółw musi narysować pierwszy bok, czyli przejść np. 50 kroków i obrócić się o 45 stopni itd. W jaki sposób zapisać powtarzające się polecenia? Jak wykorzystać utworzony ośmiokąt do wyświetlenia ciekawej kompozycji?

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

## Kilka praktycznych porad

- Wprowadzane polecenia muszą być zgodne z regułami języka Logo. Pomiedzy nazwą (skrótorem nazwy) polecenia a wartością liczbową powinna być umieszczona spacja. W przeciwnym wypadku 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**.
- Podczas pisania programu 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 (polecenia) skopiowane (**Ctrl+C**) wcześniej do **Schowka**.

Jeśli nie wiemy, o ile kroków należy przesunąć żółwia, możemy wpisać komentarz **np** bez podawania liczby kroków i nacisnąć klawisz **Enter**. Na ekranie ukaze się **linijka** (rys. 2). Po wskazaniu kursorem myszy miejsca, do którego ma dojść żółw, i kliknięciu program sam uzupełni potrzebną liczbę kroków, a polecenie **np** zostanie umieszczone w historii poleceń. Podobnie postępujemy, jeśli nie wiemy, o jaki kąt należy obrócić żółwia.



Rys. 2. Linijka w programie Logomocja



### Ćwiczenie 2. Rysujemy dom

- Narysuj w programie Logomocja dom podobny do pokazanego na rysunku 1.
- Zapisz projekt w pliku pod nazwą *Dom*.

## 2. Wykonywanie obliczeń i stosowanie zmiennych

Zamierzamy obliczyć pole powierzchni prostokątnej działki o bokach przyjmujących dowolne wartości dodatnie. W jaki sposób można w języku Logo zapisać polecenia umożliwiające wykonanie obliczeń i wyświetlenie wyniku na ekranie?

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



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

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



#### Ćwiczenie 6. Definiujemy procedurę bez parametrów

1. Zdefiniuj procedurę bez parametrów OŚMIOKA1 (rys. 3.), rysującą ośmiokąt foremny o boku 50 kroków i o krawędziach koloru niebieskiego.
2. Zapisz projekt w pliku pod nazwą *Projekt1*.

**Wskazówka:** Przed poleceniem **powtórź** dodaj: **ukp "niebieski** (zmiana koloru pisaka na niebieski).

Aby wywołać procedurę, należy wpisać jej nazwę w wierszu poleceń po znaku za-  
chety „?”. Zdefiniowanej procedury możemy używać w definicjach innych procedur.  
Na przykład, używając procedury OŚMIOKA1, możemy otrzymać obraz przedsta-  
wiony na rysunku 4.:

oto OŚMIOKA1Y1  
**powtórź 12 [OŚMIOKA1 lw 30]**  
już



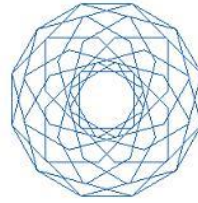
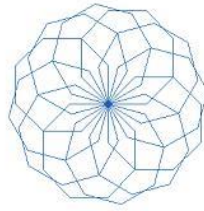
Jeśli zapiszemy projekt w pliku, procedura również zostanie w nim zapisana.  
Po otwarciu projektu z pliku możemy procedurę ponownie wywoływać,  
używając jej nazwy. Utworzoną procedurę można też wykorzystywać w innej  
procedurze. W jednym projekcie możemy utworzyć i zapisać wiele procedur.



#### Ćwiczenie 7. Korzystamy z utworzonych wcześniej procedur

1. Otwórz plik *Projekt1* zapisany w ćwiczeniu 6. Korzystając z procedury OŚMIOKA1, zdefiniuj procedurę OŚMIOKA1Y1, rysującą figurę taką, jak pokazano na rysunku 4.
2. Korzystając z procedury OŚMIOKA1Y1, zdefiniuj procedurę OŚMIOKA1Y2, rysującą figurę taką, jak pokazano na rysunku 5.
3. Zapisz plik pod tą samą nazwą.

**Wskazówka:** Do treści procedury OŚMIOKA1Y1 należy dodać w odpowiednim miejscu polecenie **np 50**.



**Rys. 4.** Wykonanie procedury OŚMIOKA1Y1 – ćwiczenie 7. (punkt 1.)

**Rys. 5.** Wykonanie procedury OŚMIOKA1Y2 – ćwiczenie 7. (punkt 2.)



#### Ćwiczenie 8. Drukujemy ekran graficzny

Otwórz plik *Projekt1* zapisany w ćwiczeniu 7. Wydrukuj ekran graficzny z wybranym rysunkiem (efektem wykonania wybranej procedury).



Aby powtarzać operacje w języku Logo, należy użyć polecenia **powtórź**, które ma postać:  
**powtórź liczba\_powtórzeń [lista\_poleceń]**

Na przykład:

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



#### Ćwiczenie 5. Realizujemy powtórzenia (iteracje)

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

## 4. Stosowanie procedur

W języku Logo możemy korzystać ze standardowych procedur tego języka bądź definiować własne (bez parametrów i z parametrami).

### 4.1. Procedury bez parametrów

Zamierzamy opracować cząstkowy problem, jakim jest narysowanie ośmiokąta, w postaci programu. W jaki sposób zdefiniować podprogram w języku Logo? W jaki sposób można go potem wykorzystać do definiowania nowego podprogramu?

W Logo możemy korzystać ze standardowych procedur tego języka bądź definiować własne. 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 OŚMIOKA1  
**powtórź 8 [np 50 lw 45]**  
już

zostanie zdefiniowana procedura OŚMIOKA1, rysująca ośmiokąt foremny o boku 50 kroków.



**Rys. 3.** W oknie pamięci widoczna jest treść zdefiniowanej procedury OŚMIOKA1. W tym miejscu można też edytować treść procedury



Chcemy narysować  $K$  takich samych trójkątów, które będą przesunięte o podaną liczbę kroków i obrócone o podany kąt. Jak to zrobić, definiując najpierw procedurę TRÓJKĄT?



**Ćwiczenie 10.** Definiujemy procedurę rysującą trójkąt równoboczny

1. Definiuj procedurę TRÓJKĄT z parametrem  $N$ , rysującą trójkąt równoboczny ( $N$  – długość boku trójkąta).
2. Zapisz projekt w pliku pod nazwą *Projekt2*.

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



**Ćwiczenie 11.** Definiujemy procedurę rysującą kompozycję składającą się z trójkątów

1. Korzystając z procedury TRÓJKĄT zapisanej w pliku *Projekt2* w ćwiczeniu 10., zdefiniuj procedurę iteracyjną TRÓJKĄTY z parametrem  $K$ , gdzie  $K$  to liczba trójkątów. Trójkąt ma być przesuwany o 10 kroków i obracany w lewo o kąt  $35^\circ$ . Przyjmij bok trójkąta (parametr aktualny) równy 100.
2. Wywołaj procedurę TRÓJKĄTY dla  $K = 40$ .
3. Zapisz plik pod tą samą nazwą.

## 5. Sytuacje warunkowe



W języku Logo polecenie realizujące sytuację warunkową ma postać:

**jeżeli warunek [lista\_poleceń1] [lista\_poleceń2].**

Jeśli warunek jest spełniony, to realizowane są polecenia *lista\_poleceń1*, w przeciwnym przypadku realizowane są polecenia *lista\_poleceń2*.

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

1. 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.
2. Zapisz projekt w pliku pod nazwą *Projekt3*.

## 4.2. Procedury z parametrami

Chcielibyśmy rysować wielokąty o dowolnej liczbie boków, wprowadzanej z klawiatury. W jaki sposób zdefiniować odpowiednią procedurę?

W języku Logo, podobnie jak w języku Scratch, możemy definiować procedury z parametrami.



**Aby zdefiniować procedurę z parametrami**, należy wpisać:

**oto nazwa\_procedury .parametr1 .parametr2**

*treść procedury*

**już**

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

Na przykład:

```
oto SZĘŚCIOKĄT :N
```

```
  powtórz 6 [np :N lw 60]
```

**już**

definiuje procedurę rysowania sześciokąta foremnego o boku długości  $N$ .

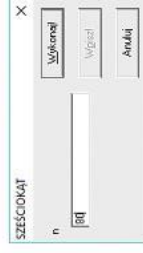
W definicji procedury określamy parametry formalne. W procedurze SZĘŚCIOKĄT określiliśmy parametr o nazwie  $N$ . W momencie wywołania procedury parametr formalny jest zastępowany przez parametr aktualny, którego wartość może być wprowadzona np. z klawiatury. Możemy też wywołać procedurę, przekazując do niej od razu konkretną daną, w tym przypadku liczbową. Procedurę SZĘŚCIOKĄT wywołamy, podając konkretną wartość parametru  $N$ , np. SZĘŚCIOKĄT 100, SZĘŚCIOKĄT 50. W tym przypadku parametr  $N$  oznacza długość boku sześciokąta.



**Ćwiczenie 9.** Definiujemy procedurę z parametrami

1. Zdefiniuj procedurę SZĘŚCIOKĄT z parametrem  $M$ , rysującą sześciokąt foremny ( $N$  – długość boku sześciokąta) o krawędziach koloru czerwonego.
2. Wywołaj procedurę SZĘŚCIOKĄT dla kilku różnych danych (długości boku).

**Wskazówka:** Jeśli w wierszu poleceń wpiszesz samą nazwę procedury (bez podania wartości parametru), otworzy się okno, w którym będziemy mogli wpisać wartość parametru (rys. 6).



**Rys. 6.** Okno, w którym wpisujemy dane przekazywane do procedury

## 6. Zmianianie postaci żółtwa

### 6.1. Stosowanie gotowych postaci

W programie Logomocja możemy zmienić żółtwa na inną postać, także animowaną. Można ją wybrać z gotowych propozycji zapisanych w plikach i dołączonych do programu Logomocja. Możemy również utworzyć własną postać w Edytorze postaci, który jest częścią programu Logomocja, i zapisać ją w pliku.



**Aby zmienić postać żółtwa na inną, należy:**

- na ekranie graficznym kliknąć żółtwa prawym przyciskiem myszy i z menu kontekstowego wybrać polecenie **Zmień żł**,
- na zakładce **Postać** kliknąć przycisk **Ustal postać**,
- w otwartym oknie **Wybierz postać** odszukać i kliknąć nazwę pliku zawierającego obraz przedstawiający postać, w którą ma zmienić się żółw – w oknie **Wybierz postać** pojawi się podgląd postaci,
- kliknąć przycisk **Otwórz** – w oknie zmiany żółtwa możemy zmienić parametry postaci, m.in. powiększyć rysunek postaci. W tym celu trzeba wpisać większą wartość w polu **Skala X** (domyślnie rozmiar jest ustawiony na 1),
- kliknąć przycisk **OK**.



### Ćwiczenie 13. Zmieniamy postać żółtwa

1. Zmień żółtwa na inną, wybraną przez siebie postać.
2. Utwórz program, w którym nowy żółw (nowa postać) „przejdzie” dookoła ekranu, nie rysując ramki.

### 6.2. Korzystanie z Edytora postaci

W Edytorze postaci można tworzyć rysunki w podobny sposób jak w programie Paint. Edytor postaci umożliwia ponadto tworzenie **obrazu animowanego**.

Do rysowania figur geometrycznych program udostępnia narzędzia malarskie: **Elipsa**, **Prostokąt**, **Wielobok**, **Linia**. Obraz można powiększać, wypełniać obszary zamknięte i rysować luki. Możemy ustalić grubość rysowanej linii, wybierając rodzaj **stałówki**. Kolory wybieramy z **palety barw**. Z narzędzi tych korzysta się podobnie jak w programie Paint. Różnice są nieznaczne, np.:

- aby narysować figurę (prostokąt, wielokąt lub elipsę) z wypełnieniem, należy przed rozpoczęciem rysowania kliknąć dane narzędzie raz lub dwa razy (figura zostaje wówczas wypełniona),
- niektóre narzędzia wybieramy z listy wyboru, np. **Pióro**, **Ołówek**, **Gumkę**; narzędzia podzielono na dwie grupy – w związku z tym można tworzyć kombinacje narzędzi, np. **Gumka** może przyjmować kształt elipsy; dlatego podczas rysowania należy zwrócić uwagę, które narzędzie zostało wybrane z listy.

Rysunek utworzony w Edytorze postaci zapisuje się w pliku z rozszerzeniem *lgf*.



### Ćwiczenie 14. Zapoznajemy się z Edytorem postaci

1. Uruchom Edytor postaci. Zwróć uwagę na budowę okna programu.
2. Zapoznaj się z interfejsem programu. Zwróć uwagę na wybór i działanie narzędzi. Porównaj je z możliwościami programu Paint.
3. Wypróbuj możliwość rysowania figur z wypełnieniem i bez wypełnienia.
4. Sprawdź, w jaki sposób rysuje się kwadrat i koło, a następnie porównaj go ze sposobem rysowania tych figur w programach Paint i GIMP.



### Ćwiczenie 15. Tworzymy rysunek w Edytorze postaci

1. Utwórz rysunek postaci składający się z różnych figur geometrycznych. Postać przedstawiona na rysunku 7. może stanowić inspirację.
2. Zapisz rysunek w pliku pod nazwą *Ludzik*.

**Wskazówka:** Z menu kontekstowego palety kolorów można wybrać inny sposób jej wyświetlenia.



**Aby umieścić zaznaczony fragment rysunku na przezroczystym tle,** należy wybrać polecenie **Rysowanie przezroczyste** z menu kontekstowego zaznaczonego fragmentu lub z menu **Rysowanie/Zaznaczenie**.

### 6.3. Tworzenie animacji w Edytorze postaci

{  
Zamierzamy zamienić żółtwa na utworzoną samodzielnie ruchomą postać. W jaki sposób w środowisku Logomocja można przygotować animowany rysunek?  
}

Edytor postaci umożliwia tworzenie **animacji komputerowych**. Animacja składa się z klatek. Im więcej klatek jest wyświetlanych w jednostce czasu, tym bardziej płynny jest ruch. Kolejne klatki animacji tworzymy, korzystając z kopiowania przez **Schówek** (rys. 7). W obszarze do rysowania aktualnie widoczna klatka jest obramowana na niebiesko.



### Ćwiczenie 16. Oglądamy gotową animację

1. Uruchom Edytor postaci.
2. Otwórz i uruchom kilka gotowych animacji dostępnych w tym programie.

Tworzenie nowej animacji zaczynamy od narysowania rysunku, który będzie podstawą do przygotowania kolejnych klatek animacji. Następnie kopiujemy klatkę i na kopii наносimy zmiany, korzystając z możliwości programu, m.in. z przekształceń rysunku.

W Edytorze postaci rodzaj przekształcenia można wybrać, korzystając z opcji **Rysowanie/Zaznaczenie** lub z menu kontekstowego zaznaczonego fragmentu obrazu.





## Warto zapamiętać

- Środowisko programowania Logomocja umożliwia tworzenie projektów, w skład których mogą wchodzić ekran graficzny, utworzone procedury oraz nowe postacie żółwi.
- 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 z parametrami, 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**.
- W Edytorze postaci można utworzyć własną postać animowaną, w którą można zmienić żółwia.

Zaznaczony fragment rysunku można również przekształcić, klikając wewnątrz zaznaczenia i chwytając za narożnik zaznaczenia. Jednokrotne kliknięcie umożliwia zmianę rozmiaru i wykonanie odbicia, dwukrotne – obrócenie, a czterokrotne – pochylenie.



**Aby utworzyć kolejną klatkę animacji, należy:**

- kliknąć w dolnym oknie klatkę, na podstawie której chcemy przygotować kolejną klatkę,
- skopiować klatkę do **Schowka** (opcja **Kopiuj**) – rys. 7.,
- wkleić skopiowaną klatkę za wybraną klatką lub przed nią (opcja **Wklej za** lub **Wklej przed**),
- na wklejonej klatce nanieść zmiany.

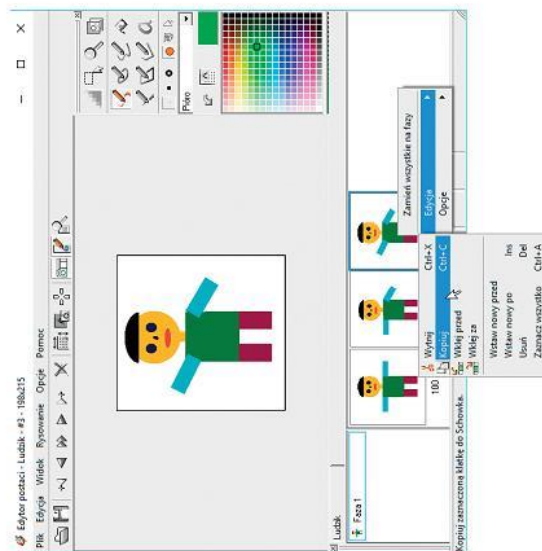
**Aby uruchomić animację, należy kliknąć przycisk **Podgląd****

aby ją zatrzymać – przycisk **Zakończ podgląd**.



**Ćwiczenie 17.** Tworzymy własną animację w Edytorze postaci

1. Utwórz animację postaci utworzonej w ćwiczeniu 15. Postaraj się przygotować przynajmniej pięć klatek.
2. Zapisz plik pod tą samą nazwą.



**Rys. 7.** Przykład tworzenia animacji w Edytorze postaci – ćwiczenia 15. i 17. W oknie **Treści** widoczne są kolejne klatki animacji