

## T: Programowanie w języku Scratch – 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:

### Temat 12.

## Programowanie w języku Scratch

1. Tworzenie programu w języku Scratch
2. Powtarzanie poleceń
3. Zmienne i obliczenia
4. Sytuacje warunkowe
5. Stosowanie procedur
  - 5.1. Procedury bez parametrów
  - 5.2. Procedury z parametrami
6. Projekt grupowy – gra

### Warto powtórzyć

1. Czym jest program komputerowy?
2. Czym jest język programowania?
3. Jaki algorytm nazywamy algorytmem z warunkami, a jaki – iteracyjnym?
4. Dlaczego stosuje się procedury?
5. Jakże są etapy i zasady przygotowania projektu grupowego?

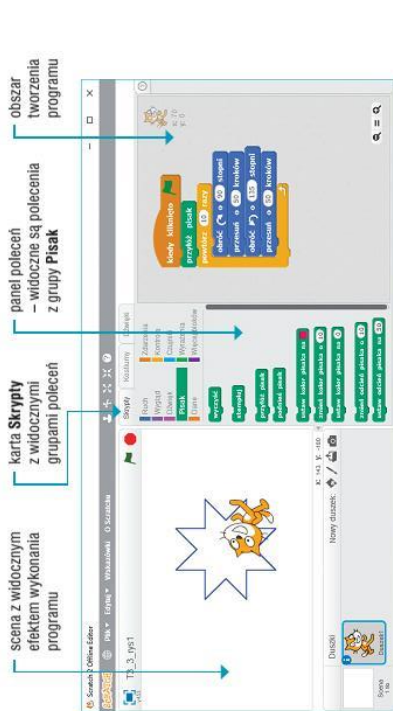
### 1. Tworzenie programu w języku Scratch

Program Scratch jest dydaktycznym środowiskiem programowania umożliwiający tworzenie programów w języku Scratch. Polecenia języka Scratch są reprezentowane przez elementy graficzne. Program Scratch jest dostępny bezpłatnie w Internecie.

Tworzenie programu w języku Scratch polega na umieszczaniu odpowiednich poleceń w obszarze tworzenia programu. Polecenia podzielone są na dziesięć grup, które zobaczymy po wybraniu karty **Skrypty** (rys. 1).

Na scenie widoczna jest postać (zwana „duszkciem”), która standardowo ma wygląd kota. Wygląd duszka można zmienić. Duszek porusza się po scenie lub wykonuje inne czynności (np. miauczy) zgodnie z wydawanymi poleceniami. Na przykład polecenie **przesuwa o 50 kroków** przesuwa duszka o 50 kroków. Wartość parametru określającego liczbę kroków można zmieniać.

Polecenia możemy ze sobą łączyć (jak puzzle), umieszczając jedno pod drugim – tak zestawione polecenia będziemy w programie Scratch nazywać **skryptem**. Program może zawierać więcej niż jeden skrypt. Utworzony program można zapisać w pliku (opcja **Plik/Zapisz jako**).



Rys. 1. Okno programu Scratch



Rys. 2. Przykład skryptu

**Aby uruchomić skrypt**, wystarczy kliknąć jego dowolny element. Jeśli na początku skryptu umieścimy polecenie **kiedy kliknięto** (rys. 2.) z grupy **Kontrola**, będziemy mogli uruchomić skrypt (skrypty), klikając ikonę **(Rozpocznij skrypty zielonej flagi)**.

### Ćwiczenie 1. Tworzymy program w języku Scratch

1. Uruchom program Scratch. Zapoznaj się z interfejsem programu, m.in. z grupami poleceń. Sprawdź również, w jaki sposób wpisuje się liczbę kroków w poleceniu przesuń i wielkość kąta w poleceniu obróć.
2. Utwórz skrypt składający się z poleceń pokazanych na rysunku 2. Uruchom go.
3. Zapisz program w pliku pod nazwą *Projekt1*.

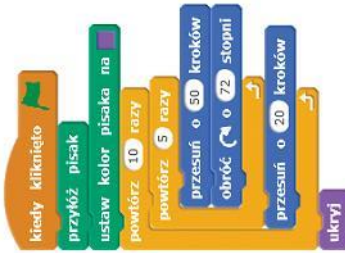
### Kilka praktycznych porad

- Polecenie umieszczone w obszarze skryptów można zdublować, wybierając z menu kontekstowego danego polecenia opcję **duplikuj**. Dane polecenie zostanie zdublowane wraz ze wszystkimi poleceniami dołączonymi do niego poniżej.
- Aby usunąć polecenie, należy je odłączyć od pozostałych poleceń i z menu kontekstowego wybrać polecenie **usuń**. Jeśli nie odłączymy polecenia od pozostałych poleceń, zostanie ono usunięte razem ze wszystkimi poleceniami dołączonymi do niego poniżej. Można też przeciągnąć polecenie na obszar panelu poleceń.



#### Ćwiczenie 4. Stosujemy pętlę w pięci

1. Otwórz plik **Pięciokąt** zapisany w ćwiczeniu 3. Korzystając z rysunku 4a, zmodyfikuj program rysujący pięciokąt foremny, tak aby powstała kompozycja pokazana na rysunku 4b. Zmień kolor i rozmiar pisaka według własnego pomysłu. Uruchom program.
2. Zapisz program w pliku pod nazwą **Pięciokąty**.
3. Odpowiedz na pytania: *Ile razy zostanie wykonana instrukcja przesuń o 50 kroków, a ile razy – przesuń o 20 kroków?*



Rys. 4a. Przykład realizacji pętli zagnieżdżonej



Rys. 4b. Efekt wykonania programu z rys. 4a



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

1. Utwórz program rysujący dziewięciokąt foremny o boku 80 kroków. Zmień kolor i rozmiar pisaka według własnego pomysłu. Zapisz program w pliku pod nazwą **Dziewięciokąt**.
2. Utwórz kompozycję według własnego pomysłu, stosując pętlę w pięci i wykorzystując program zapisany w pliku **Dziewięciokąt**.
3. Zapisz utworzony program w pliku pod nazwą **Dziewięciokąty**.

## 3. Zmienne i obliczenia

Zamierzamy obliczyć obwód i pole powierzchni prostokątnej działki o bokach przyjmujących dowolne wartości dodatnie. W jaki sposób można w języku Scratch utworzyć program umożliwiający wprowadzanie danych z klawiatury i wykonanie na nich obliczeń?

W programie Scratch można tworzyć zmienne, które mogą być wykorzystywane m.in. w obliczeniach.



**Aby utworzyć nową zmienną**, należy wybrać element **Utwórz zmienną (z grupy Dane)** – otwórz się okno dialogowe, w którym wpisujemy nazwę zmiennej. Utworzone zmienne będą widoczne w panelu poleceń (rys. 5).



Zmiennej o danej nazwie możemy przypisać wartość, stosując polecenie **ustaw liczbę na 0**. W polu za słowem **na** możemy podać konkretną wartość (rys. 5) lub umieścić element **odpowiedź** (z grupy **Czujniki**), umożliwiający wprowadzanie danych z klawiatury (rys. 7). Możemy też przypisać zmiennej wyrażenie (rys. 5 i 7). Aby była widoczna wartość zmiennej na scenie, w panelu poleceń musi być zaznaczone pole obok nazwy zmiennej (rys. 5).

- Niektóre polecenia możemy wydawać duszkowi, klikając wybrany element w panelu poleceń, m.in. z grup **Ruch**, **Wygląd** i **Dźwięk**.
- Aby wyciszyć scenę, należy użyć polecenia **wyciszyć** z grupy **Pisak**.
- Aby zmienić położenie duszka na scenie, możemy uchwycić go i przeciągnąć w inne miejsce sceny.



#### Ćwiczenie 2. Modyfikujemy program i zapisujemy w pliku

1. Wyciszyć scenę.
2. Do programu zapisanego w ćwiczeniu 1. dodaj jeszcze kilka poleceń z grup **Ruch** i **Dźwięk** – według własnego pomysłu.
3. Uruchom program na pełnym ekranie.
4. Zapisz program w pliku pod tą samą nazwą.

## 2. Powtarzanie poleceń

Zamierzamy utworzyć program, w którym zostanie wyświetlona kompozycja składająca się z dziesięciu pięciokątów foremnych. Aby powstał pięciokąt, duszek musi narysować pierwszy bok, czyli przejsz np. 100 kroków i obrócić się o 72 stopnie itd. W jaki sposób zapisać powtarzające się polecenia? Jak wykorzystać utworzony pięciokąt do wykonania ciekawej kompozycji?



**Aby w języku Scratch zapisać powtarzające się polecenia**, możemy użyć polecenia **powtórz**, reprezentowanego przez element (z grupy **Kontrola**). Liczbę powtórzeń określamy, zmieniając odpowiednią wartość w polu tekstowym (domyślnie jest w nim wpisana wartość 10).

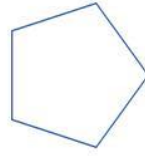


Polecenia, które mają być powtarzane, umieszczamy wewnątrz elementu **powtórz**. W szczególnym przypadku można powtarzać polecenie **powtórz** (rys. 4a), na przykład wtedy, gdy potrzebne jest zastosowanie pętli zagnieżdżonych. Korzystając z polecenia **powtórz**, możemy zapisywać m.in. algorytmy iteracyjne.



#### Ćwiczenie 3. Realizujemy powtórzenia (iterację)

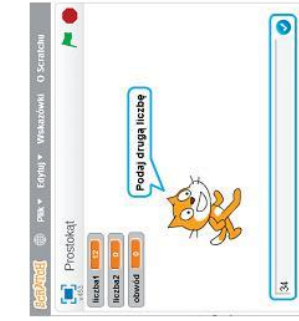
1. Korzystając z rysunku 3a, utwórz program rysujący pięciokąt foremny. Uruchom program.
2. Zapisz program w pliku pod nazwą **Pięciokąt**.
3. Odpowiedz na pytania: *Jakie polecenia wykonywane są w pętli? Dlaczego kąt obrotu wynosi 72 stopnie?*



Rys. 3a. Przykład realizacji pętli prostej – ćwiczenie 3.

Rys. 3b. Efekt wykonania programu z rys. 3a





Rys. 7. Program obliczający obwód prostokąta – ćwiczenie 7.

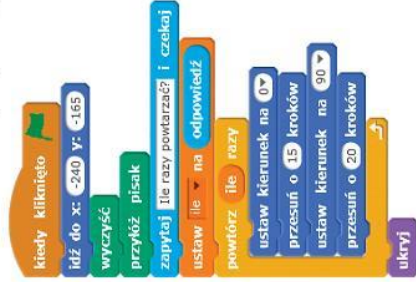
Wartości zmiennych możemy również używać – zamiast z góry określonych liczb – do określania liczby powtórzeń poleceń, np. przesuwania, zmiany kierunku. W programie przedstawionym na rysunku 8. liczba powtórzeń jest określona przez zmienną *ile*.



### Ćwiczenie 8. Stosujemy zmienne do określania liczby powtórzeń

1. Umieść w obszarze skryptów polecenia pokazane na rysunku 8. (pamiętaj o utworzeniu najpierw zmiennej *ile*). Uruchoom program.
2. Odpowiedz na pytania: *Jaki efekt otrzymaliśmy na scenie? Jakie polecenia są umieszczone w kolejnych wierszach programu? Które z nich są wykonywane w pętli?*
3. Zmodyfikuj program, aby liczby kroków, o które ma przesuwać się duśzek, były podawane z klawiatury.
4. Zapisz program w pliku pod nazwą *Wzorek*.

**Wskazówka:** W programie zastosowano polecenie zmiany położenia duśzka na scenie za pomocą współrzędnych sceny (rys. 9.). Aktualne współrzędne kursora myszy są widoczne w prawym dolnym rogu sceny.



Rys. 8. Zastosowanie zmiennej do określania liczby powtórzeń

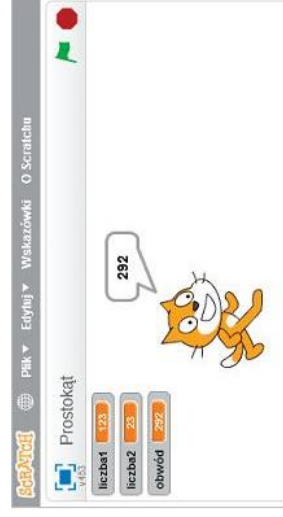


Rys. 5. Program obliczający obwód prostokąta – ćwiczenie 6.



### Ćwiczenie 6. Tworzymy zmienne i wykonujemy na nich obliczenia

1. Korzystając z rysunku 5., utwórz program obliczający obwód prostokąta i wyświetlający wynik na ekranie. Sprawdź, w której grupie są elementy umożliwiające tworzenie wyrażeń.
2. Zmieniaj kilkakrotnie wartości boków prostokąta, uruchamiaj każdorazowo program i obserwuj zmianę wartości zmiennych na scenie (rys. 6.).
3. Zapisz program w pliku pod nazwą *Prostokąt*.



Rys. 6. Wynik działania programu widoczny na scenie – ćwiczenie 6. Na scenie możemy uwidocznic wartości zmiennych stosowanych w programie



### Ćwiczenie 7. Wprowadzamy wartości zmiennych z klawiatury

1. Otwórz program *Prostokąt* zapisany w ćwiczeniu 6. Zmień program, dodając i modyfikując polecenia tak, jak pokazano na rysunku 7.
2. Dodaj do programu polecenia obliczenia pola powierzchni prostokąta. Zapisz program w pliku pod tą samą nazwą.
3. Uruchoom program kilkakrotnie. Zauważ, że po każdorazowym uruchomieniu programu poprzednie wartości zmiennych zostają zastąpione nowymi.

### Wskazówki:

- Przed poleceniem **ustaw**, należy dodać polecenie **zapytaj**.
- Wartości zmiennych należy wprowadzić za pomocą klawiatury po uruchomieniu programu.

## 4. Sytuacje warunkowe

Zamierzamy utworzyć program, w którym zależnie od wyniku obliczeń zostanie wyświetlony inny napis. Jak w języku Scratch zapisać sytuację warunkową?

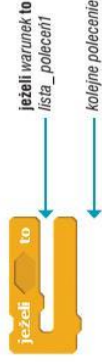
W programie Scratch polecenie realizujące sytuację warunkową (instrukcja warunkowa) reprezentuje element pokazany na rysunku 10a. W wersji uproszczonej polecenie reprezentuje element pokazany na rysunku 10b.

W wersji pełnej, jeśli *warunek* jest spełniony, wykonywane są polecenia *lista\_poleceń1*, w przeciwnym przypadku wykonywane są polecenia *lista\_poleceń2*.

W wersji uproszczonej, jeśli *warunek* jest spełniony, realizowana jest lista poleceń programu umieszczona po słowie *to*. Następnie (niezależnie od spełnienia warunku) realizowane są kolejne polecenia programu.



Rys. 10a. Element reprezentujący polecenie warunkowe w wersji pełnej



Rys. 10b. Element reprezentujący polecenie warunkowe w wersji uproszczonej

Warunek może być prosty, np. `liczba1 > 0`, lub złożony, np. `liczba = 2 i lub liczba = 4 i liczba > 0 i liczba < 10`.

### Ćwiczenie 9. Stosujemy instrukcję warunkową

- Umieść w obszarze skryptów polecenia pokazane na rysunku 11. Uruchoom program.
- Stormiluj zadanie, którego rozwiązaniem jest ten program. Zwróć uwagę na sposób obliczania sumy ocen.
- Odpowiedz na pytania: *Jakie polecenia są wykonywane w pętli? Jakie polecenie jest wykonywane, gdy warunek jest spełniony, a jakie gdy nie jest spełniony?*
- Zmodyfikuj zadanie tak, aby w przypadku nagrody kot „podskoczył” do góry i opadł z powrotem w dół, a w przypadku braku nagrody zamiauczał.
- Zapisz program w pliku pod nazwą *Srednia*.

**Wskazówka:** Przypomnij sobie sposób obliczania sumy  $n$  liczb w pętli (rysunek 4., temat 10.).

### Ćwiczenie 10. Stosujemy instrukcję warunkową z warunkiem złożonym

- Utwórz program obliczający pole powierzchni trójkąta o boku  $a$  i opuszczonej na ten bok wysokości  $h$ , jeśli podane zostaną dodatnie wartości  $a$  i  $h$ . W przeciwnym wypadku powinien zostać wyświetlony odpowiedni komunikat.
- Zapisz program w pliku pod nazwą *Pole trójkąta*.



Rys. 11. Przykład zastosowania instrukcji warunkowej – ćwiczenie 9.



### Ćwiczenie 11. Stosujemy instrukcję iteracyjną i warunkową

- Umieść w obszarze skryptów polecenia pokazane na rysunku 12. Uruchoom program.
- Przeanalizuj kod programu i odpowiedz na pytania: *Kiedy rysowany jest kwadrat? Dlaczego kot zawsze miauczy?*
- Popraw program tak, aby kot miauczał tylko w przypadku, gdy zmienna *litera* nie będzie równa „K”.
- Zapisz program w pliku pod nazwą *Kwadrat*.

Rys. 12. Przykład zastosowania instrukcji warunkowej i iteracyjnej – ćwiczenie 11.

## 5. Stosowanie procedur

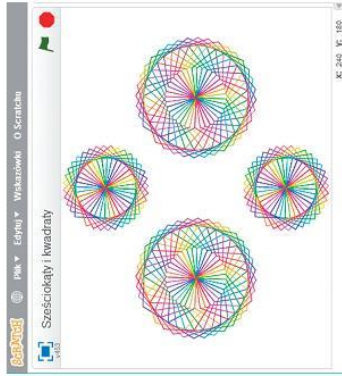
W języku Scratch procedury nazwano blokami. Możemy definiować procedury (bloki) bez parametrów i z parametrami.

### 5.1. Procedury bez parametrów

Chcemy utworzyć kompozycję składającą się z sześciokątów i kwadratów umieszczonych w wybranych miejscach sceny. W jaki sposób można w języku Scratch opracować cząstkowe problemy, jakimi są narysowanie sześciokąta i kwadratu, a potem wykorzystywać je kilkakrotnie w programie?

Aby w programie nie powtarzać poleceń rysujących sześciokąt i kwadrat, możemy zdefiniować procedury (bloki) rysujące te figury i wywołać je w programie (nawet wielokrotnie).





Rys. 14. Kompozycja – ćwiczenie 13.



### Ćwiczenie 13. Stosujemy kilka procedur w jednym programie

1. Otwórz plik **Sześciokąt** zapisany w ćwiczeniu 12. Zdefiniuj procedurę **Kwadrat**, rysując kwadrat o boku 50 kroków.
2. Zmodyfikuj program i procedury, tak aby powstała kompozycja podobna do pokazanej na rysunku 14.
3. Zapisz program w pliku pod nazwą **Sześciokąt i kwadraty**.

**Wskazówka:** Aby zmienić położenie duszka na scenie i rozpocząć rysowanie w wybranym miejscu sceny, należy użyć polecenia: **idź do x: 120 y: 0**. Kolor pisaka można zmieniać, stosując polecenie: **zmień kolor pisaka o 5** (można wpisać też inną wartość). Aby duszek nie rysował zbędnych linii pomiędzy figurami, należy odpowiednio podnieść pisak. Zastanów się, czy polecenia opuszczania i podnoszenia pisaka powinny być umieszczone w programie głównym, czy w procedurach.

## 5.2. Procedury z parametrami

Chcemy rysować wielokąty o dowolnej liczbie boków, tak by wartość tej liczby była wprowadzana z klawiatury po uruchomieniu programu. W jaki sposób zdefiniować procedurę rysującą wielokąt, aby było to możliwe?

Aby rysować wielokąty o dowolnej liczbie boków, możemy zastosować w procedurze parametr określający liczbę boków. Procedura może zawierać więcej niż jeden parametr.

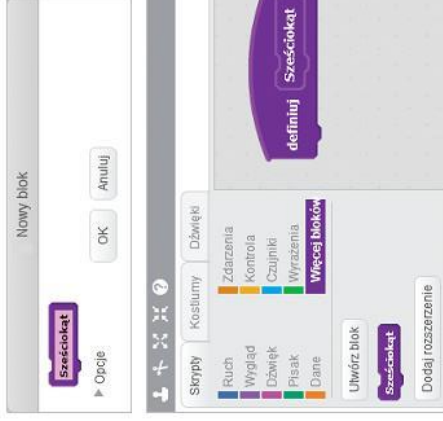


**Aby zdefiniować procedurę z parametrem**, należy:

- w oknie **Nowy blok** wybrać polecenie **Opcje** (rys. 15a),
- wybrać typ parametru, np. **liczbowy**,
- wpisać nazwę parametru i kliknąć przycisk **OK** – w obszarze tworzenia programu pojawi się element **definiuj** z nazwą procedury i parametru (rys. 15b).

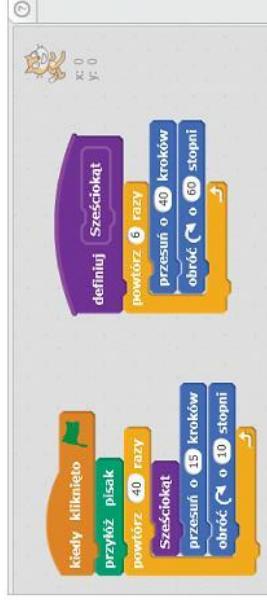


**Aby zdefiniować procedurę (blok)**, należy wybrać element (z grupy **Więcej bloków**) – otworzy się okno dialogowe **Nowy blok** (rys. 13a), w którym wpisujemy nazwę procedury. W panelu poleceń pojawi się nazwa procedury, a w obszarze tworzenia programu – element **definiuj** nazwa procedury (rys. 13b), pod którym umieszczamy polecenia procedury (rys. 13c).



Rys. 13a. Definiowanie procedury – okno dialogowe **Nowy blok**

**Rys. 13b.** W panelu poleceń widoczna jest nazwa procedury (bloku), a w obszarze tworzenia programu – element **definiuj Sześciokąt**



**Rys. 13c.** Przykład zastosowania zdefiniowanej procedury w programie – ćwiczenie 12.

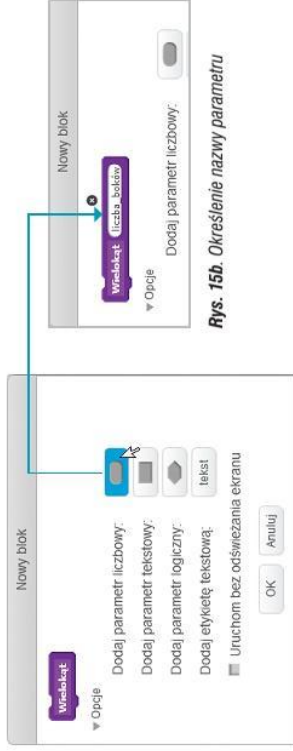


**Aby zastosować procedurę (blok) w programie**, należy ją wywołać, umieszczając nazwę procedury (bloku) w odpowiednim miejscu (miejscach) programu. Wywołanie procedury powoduje wykonanie składających się na jej treść instrukcji.



### Ćwiczenie 12. Definiujemy procedurę i wywołujemy ją w programie

1. Zdefiniuj procedurę **Sześciokąt** rysującą sześciokąt foremny o boku 40 kroków. Następnie (w tym samym obszarze bocznym) utwórz program rysujący kompozycję składającą się z czterdziestu sześciokątów (rys. 13c).
2. Zapisz program w pliku pod nazwą **Sześciokąty**.



Rys. 15b. Określenie nazwy parametru

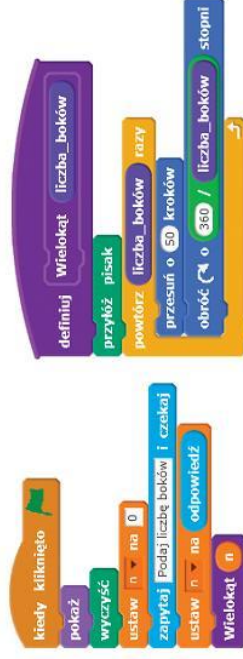
Rys. 15a. Definiowanie procedury z parametrem. Wybór typu parametru

W definicji procedury określamy **parametry formalne**, które w momencie wywołania procedury są zastępowane przez **parametry aktualne**.

### P Parametry

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

Na rysunku 16. pokazana jest przykładowa definicja procedury *Wielokąt* z parametrem określającym liczbę boków wielokąta (*liczba\_boków*). Procedura jest wywoływana z parametrem aktualnym *n*, którego wartość jest wprowadzana z klawiatury w momencie wywołania procedury.



Rys. 16. Przykład zastosowania w programie zdefiniowanej procedury z jednym parametrem – ćwiczenie 14.



### Ćwiczenie 14. Definiujemy procedurę i wywołujemy ją w programie

1. Zdefiniuj procedurę *Wielokąt* rysującą wielokąt foremny o boku 50 kroków z parametrem *liczba\_boków*, który będzie określał liczbę boków (rys. 16.). Utwórz program rysujący wielokąt o liczbie boków podawanej z klawiatury. W programie wywołaj zdefiniowaną procedurę z parametrem aktualnym *n*.
2. Zapisz program w pliku pod nazwą *Wielokąt*.