

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. Jakie są etapy i zasady przygotowania projektu grupowego?

1. Tworzenie programu w języku Scratch

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

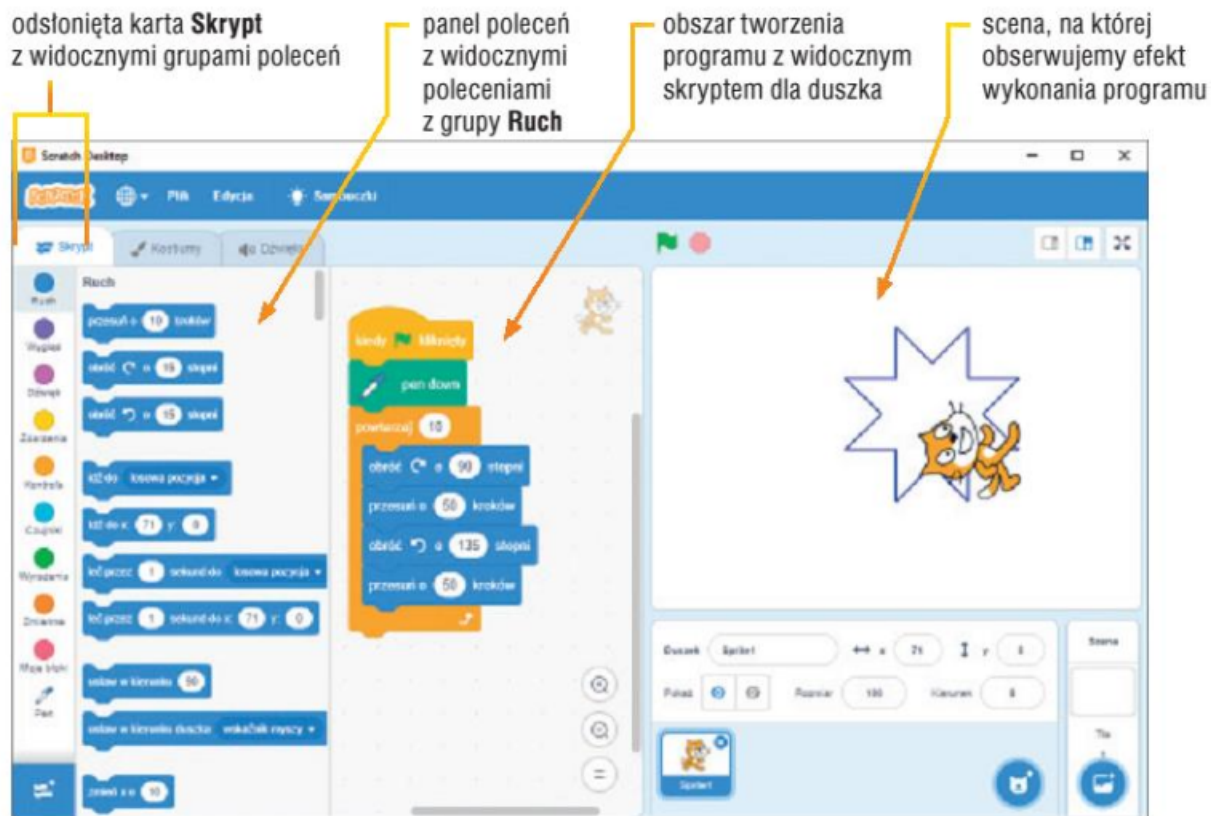
Tworzenie programu w języku Scratch polega na umieszczaniu odpowiednich poleceń w obszarze tworzenia programu. Polecenia podzielone są na dziesięć grup, które zobaczymy po wybraniu karty **Skrypt** (rys. 1.).

Na scenie widoczna jest postać (zwana „duszką”), 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 pole-

cenie  przesuwają 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 na swoim komputerze**).




Rys. 1. Okno programu Scratch



Rys. 2. Przykład skryptu

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

Polecenia dotyczące pisaka (np. **przyłóż pisak**, **wyczyść wszystko**) umieszczone są w grupie **Pióro**. Aby w programie Scratch 3.0 dodać je do panelu poleceń, należy w lewym dolnym rogu okna kliknąć przycisk **Dodaj rozszerzenie**  i wybrać grupę **Pióro**.

Polecenie umieszczone w obszarze skryptów można zdublować, wybierając z menu kontekstowego danego polecenia opcję **Duplikuj**. Aby usunąć polecenie, należy je odłączyć od pozostałych poleceń i z menu kontekstowego wybrać polecenie **Usuń blok** lub je przeciągnąć na obszar panelu poleceń.



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 przesun 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*.



Ćwiczenie 2. Modyfikujemy program i zapisujemy w pliku

1. Wyczyść 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 przejść 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 **powtarzaj**, 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 **powtarzaj**. W szczególnym przypadku można powtarzać polecenie **powtarzaj** (rys. 4a), na przykład wtedy, gdy potrzebne jest zastosowanie pętli zagnieżdżonych. Korzystając z polecenia **powtarzaj**, możemy zapisywać m.in. algorytmy iteracyjne.

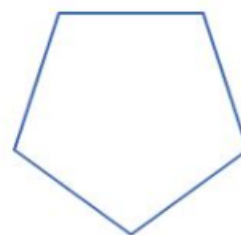


Ćwiczenie 3. Realizujemy powtórzenia (iterację)

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. 4a. Przykład realizacji pętli zagnieżdżonej



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



Ćwiczenie 4. Stosujemy pętlę w pętli

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 *Pięciokąt*.
3. Odpowiedz na pytania: *Ile razy zostanie wykonana instrukcja przesun o 50 kroków, a ile razy – przesun o 20 kroków?*



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 pętli 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.

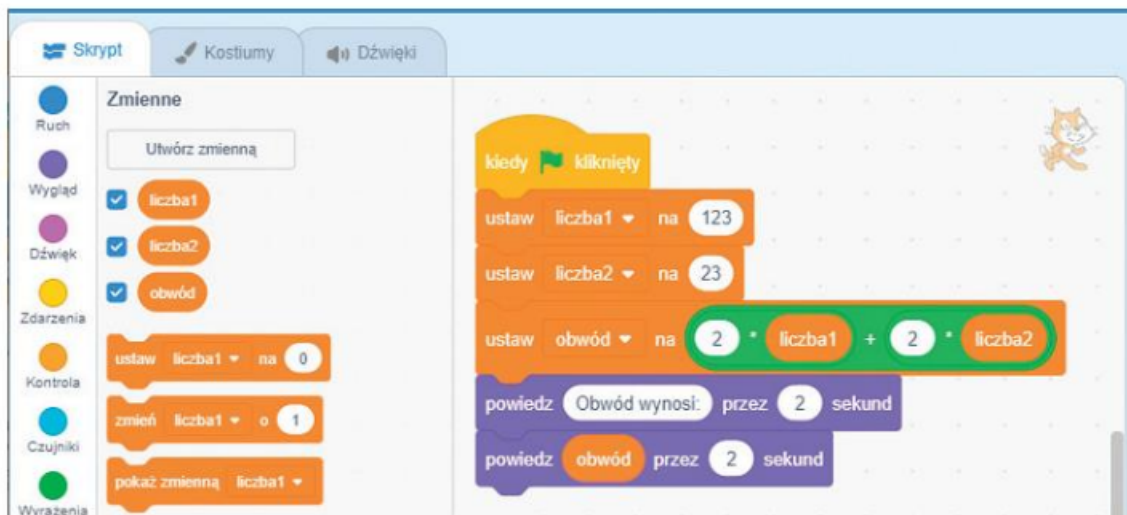


Aby utworzyć nową zmienną, należy wybrać element z grupy **Zmienne** – otworzy się okno dialogowe, w którym wpisujemy nazwę zmiennej. Utworzone zmienne będą widoczne w panelu poleceń (rys. 5.).

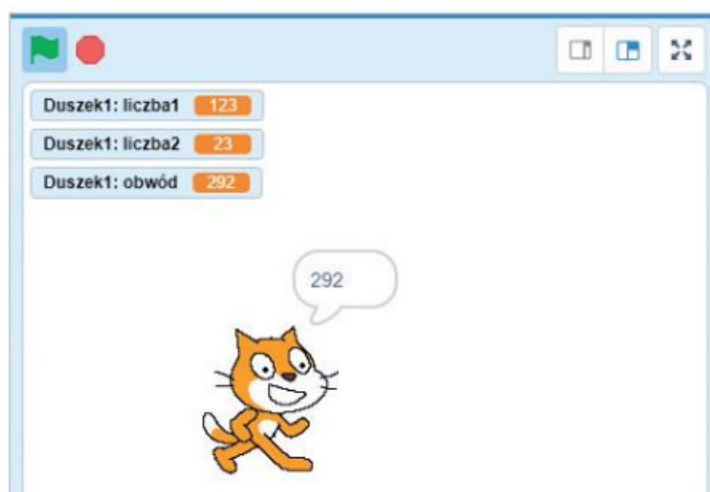
Zmiennej o danej nazwie możemy przypisać wartość, stosując polecenie



. W polu za słowem **na** możemy podać konkretną wartość (rys. 5.) lub umieścić element 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.).



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



Rys. 6. Wynik działania programu – ćwiczenie 6. Na scenie widać wartości zmiennych stosowanych w programie



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*.



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 polecenie obliczenia pola powierzchni prostokąta. Zapisz program w pliku pod tą samą nazwą.
3. Uruchom 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.



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*). Uruchom 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 duszek, były podawane z klawiatury.
4. Zapisz program w pliku pod nazwą *Szlaczek*.



Rys. 8. Zastosowanie zmiennej do określania liczby powtórzeń

Wskazówka: W programie zastosowano polecenie zmiany położenia duszka na scenie za pomocą współrzędnych sceny (rys. 9).



Rys. 9. Współrzędne sceny

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 jest kolejne polecenie programu.



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



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

Warunek może być prosty, np.



lub złożony, np.



Ćwiczenie 9. Stosujemy instrukcję warunkową

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

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



Ćwiczenie 10. Stosujemy instrukcję warunkową z warunkiem złożonym

1. 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.
2. Zapisz program w pliku pod nazwą *Pole trójkąta*.



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



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



Ćwiczenie 11. Stosujemy instrukcję iteracyjną i warunkową

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

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. 13a. Definiowanie procedury – okno dialogowe **Utwórz blok**



Aby zdefiniować procedurę (blok), należy wybrać element

Utwórz blok (z grupy **Moje bloki**) – otworzy się okno dialogowe **Utwórz blok** (rys. 13a), w którym wpisujemy nazwę procedury i klikamy **OK**. W panelu poleceń pojawi się nazwa procedury, a w obszarze tworzenia programu – element **definiuj nazwa procedury**, pod którym umieszczamy polecenia procedury (rys. 13b).



Rys. 13b. 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.



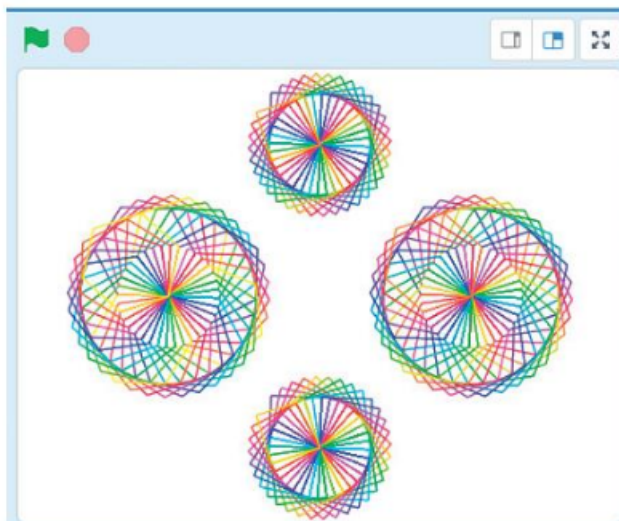
Ćwiczenie 12. Definiujemy procedurę i wywołujemy ją w programie

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



Ćwiczenie 13. Stosujemy kilka procedur w jednym programie

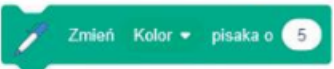
1. Otwórz plik *Sześciokąty* zapisany w ćwiczeniu 12. Zdefiniuj procedurę *Kwadrat*, rysującą kwadrat o boku 50 kroków.
2. Zmodyfikuj program i procedury, tak aby powstała kompozycja podobna do pokazanej na rysunku 14.
3. Zapisz program w pliku pod nazwą *Sześciokąty i kwadraty*.



Rys. 14. Kompozycja – ćwiczenie 13.

Wskazówka: Aby zmienić położenie duszka na scenie i rozpocząć rysowanie w wybranym

miejscu sceny, należy użyć polecenia: . Kolor pisaka można zmieniać,

stosując polecenie:  (można wpisać też inną wartość). Aby duszek nie rysował zbędnych linii pomiędzy figurami, należy odpowiednio podnosić 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? }

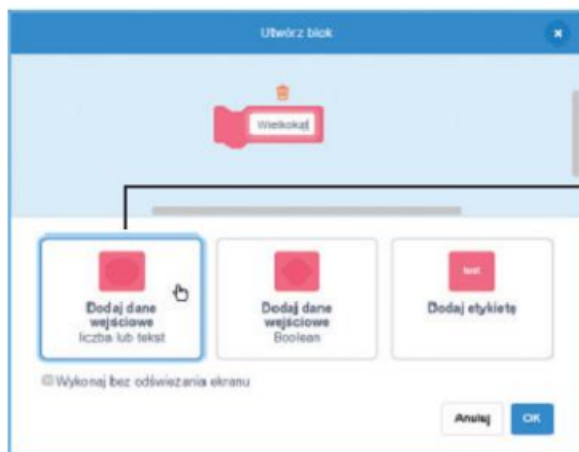
Aby rysować wielokąty o dowolnej liczbie boków, możemy zastosować w procedurze parametr określający liczbę boków. Procedura może zawierać więcej niż jeden parametr.



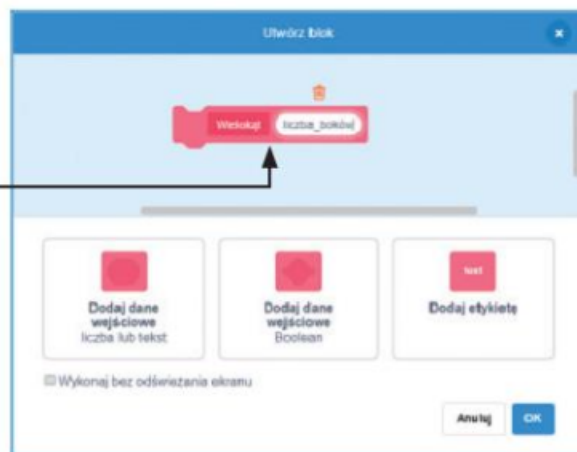
Aby zdefiniować procedurę z parametrem, należy:

- wybrać grupę **Moje bloki** i kliknąć element **Utwórz blok**,
- w oknie **Utwórz blok** wpisać nazwę procedury i kliknąć obiekt **Dodaj dane wejściowe** (rys. 15a),
- wpisać nazwę parametru i kliknąć przycisk **OK** (rys. 15b).

W obszarze tworzenia programu pojawi się element **definiuj** z nazwą procedury i parametru.



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



Rys. 15b. Określanie nazwy parametru

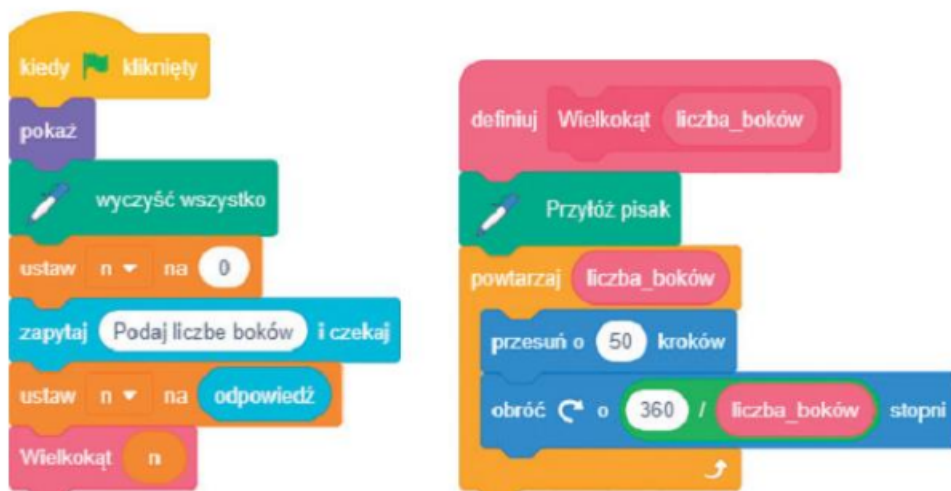


W definicji procedury określamy **parametry formalne**, które w momencie wywołania procedury są zastępowane przez **parametry aktualne**.

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*.

- Aby wywołać procedurę (blok) w programie, należy umieścić jej nazwę w odpowiednim miejscu (miejscach) programu. Wywołanie procedury powoduje, że wykonywane są instrukcje składające się na jej treść.



Pytania i polecenia

1. Wyjaśnij na przykładzie, w jaki sposób tworzy się prosty program w języku Scratch.
2. Omów polecenia realizujące powtarzanie poleceń w języku Scratch.
3. Podaj przykład zastosowania pętli zagnieżdżonej.
4. W jaki sposób tworzy się nową zmienną?
5. Omów polecenia realizujące sytuacje warunkowe w języku Scratch.
6. Omów na przykładzie sposób definiowania procedury (bloku) bez parametrów.
7. Omów na przykładzie sposób definiowania procedury (bloku) z parametrami.
8. W jaki sposób wywołujemy procedurę (blok) w programie.

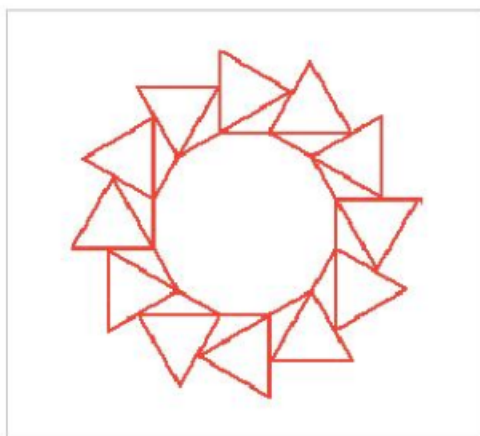


Zadania

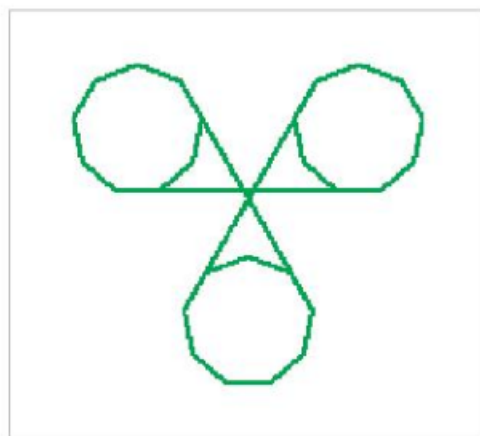
1. Utwórz program, w którym duszek przejdzie z lewego górnego rogu sceny do prawego dolnego rogu, rysując kolorowy wzorek. Zapisz program w pliku pod nazwą *Wzorek*.
2. Utwórz program, w którym duszek narysuje cztery jednakowe sześciokąty foremne – każdy umieszczony w innym rogu sceny. Wymiary dobierz tak, aby figury nie zachodziły na siebie po narysowaniu. Zapisz program w pliku pod nazwą *Cztery sześciokąty*.
3. Utwórz program, w którym duszek rysuje niebieski kwadrat o wymiarach podawanych z klawiatury. Zapisz program w pliku pod nazwą *Niebieski kwadrat*.
4. Korzystając z poleceń przesuwania i obracania duszka oraz z powtarzania poleceń, zmodyfikuj program utworzony w zadaniu 3. tak, aby otrzymać ciekawą kompozycję. Zapisz program w pliku pod nazwą *Kompozycja4*.
5. Utwórz program, w którym duszek będzie rysował czerwony prostokąt, jeśli z klawiatury zostanie wprowadzona litera *p*, w przeciwnym razie duszek-kot ma zamiauczeć. Zapisz program w pliku pod nazwą *Czerwony prostokąt*.
6. Utwórz program, w którym duszek narysuje kolorową kompozycję z figur geometrycznych. Zapisz program w pliku pod nazwą *Kompozycja6*.
7. Utwórz program obliczający sumę i średnią arytmetyczną trzech liczb wprowadzanych z klawiatury. Zapisz program w pliku pod nazwą *Suma i średnia*.
8. Zmodyfikuj program utworzony w zadaniu 7. tak, aby obliczenia były wykonywane dla dziesięciu trójek liczb. Zapisz program w pliku pod nazwą *Suma i średnia 10*.
9. Otwórz plik *Kwadrat* zapisany w ćwiczeniu 11. Zmodyfikuj program tak, aby dla wartości zmiennej *litera* różnej od „k” został narysowany romb. Zapisz program w pliku pod nazwą *Kwadrat lub romb*.
10. Utwórz program obliczający pole powierzchni kwadratu o boku *a*, jeśli podana zostanie dodatnia wartość *a*. W przeciwnym wypadku powinien zostać wyświetlony odpowiedni komunikat. Zapisz program w pliku pod nazwą *Pole kwadratu*.
11. Korzystając z ćwiczenia 13., utwórz kompozycję składającą się z trzech różnych wielokątów. Zapisz program w pliku pod nazwą *Kompozycja11*.
12. Zdefiniuj procedurę *Trójkąt* z jednym parametrem, rysującą trójkąt równoboczny. Parametr ma określać długość boku. Utwórz program rysujący pięć trójkątów: jeden umieszczony na środku sceny, a pozostałe – w rogach. W programie wywołaj zdefiniowaną procedurę. Zapisz program w pliku pod nazwą *Pięć trójkątów*. Uruchom program kilkakrotnie dla różnych danych.

Dla zainteresowanych

13. Utwórz program obliczający wartość bezwzględną liczby x wprowadzonej z klawiatury. Dla x większego od zera lub równego zero ma być wyświetlona wartość x . Dla x mniejszego od zera ma być wyświetlona wartość $-x$. Zapisz program w pliku pod nazwą *Wartość bezwzględna*.
14. Utwórz program obliczający sumę kolejnych liczb naturalnych od 1 do n , gdzie n jest wprowadzane z klawiatury. Zapisz program w pliku pod nazwą *Suma n*.
15. Otwórz program *Wielokąt* zapisany w ćwiczeniu 14. Dodaj do procedury *Wielokąt* drugi parametr określający długość boku. Zmodyfikuj odpowiednio program. Zapisz program pod tą samą nazwą.
16. Korzystając z procedury *Trójkąt* utworzonej w zadaniu 12., utwórz program rysujący figurę podobną do pokazanej na rysunku 22. W programie wywołaj zdefiniowaną procedurę. Zapisz program w pliku pod nazwą *Kompozycja16*.



Rys. 22. Kompozycja – zadanie 16.



Rys. 23. Koniczynka – zadanie 17.

17. Utwórz program rysujący figurę podobną do pokazanej na rysunku 23. Zdefiniuj odpowiednią procedurę. Zapisz program w pliku pod nazwą *Koniczynka*.
18. Zdefiniuj procedurę *Ośmiokąt* z parametrem *bok* określającym długość boku, która umożliwi rysowanie osmiokąta. Utwórz program rysujący ciekawą kompozycję składającą się z wielu kolorowych osmiokątów. W programie wywołaj zdefiniowaną procedurę z parametrem aktualnym x . Wartość parametru ma być podawana z klawiatury. Zapisz program w pliku pod nazwą *Kompozycja18*. Uruchom program kilkakrotnie dla różnych danych.

Dodatkowe zadanie projektowe

19. Przygotuj projekt i utwórz według własnego pomysłu grę dla dwóch graczy.



Przeczytaj, jeśli chcesz wiedzieć więcej...

Ze względu na modele programowania wyróżniamy m.in. **programowanie zdarzeniowe**. W programowaniu zdarzeniowym kolejność wykonywania instrukcji (poleceń) programu nie jest określona przez twórcę programu, lecz zależy od zaistnienia określonych zdarzeń (np. naciśnięcia przez użytkownika przycisku myszy, wprowadzenia jakiejś danej, upływu określonego czasu).

Język Scratch realizuje model programowania zdarzeniowego, ponieważ każdy ze skryptów składających się na program może być wykonywany po wybraniu przez użytkownika osobnej akcji.