

# Informatyka

zdalne nauczanie 16.03.2020- 20.03.2020

*Każdy uczeń celu zapisywania swoich prac jest proszony o założenie darmowego konta na platformie [www.scratch.mit.edu](http://www.scratch.mit.edu)*

## Klasa 5

Temat: Programujemy historyjkę.

Uruchom przeglądarce platformę [www.scratch.mit.edu](http://www.scratch.mit.edu), następnie utwórz program, w którym na dowolnie wybranym tle, po krótkiej wymianie zdań pies zaczyna gonić kota.

Pamiętaj o zastosowaniu takich poleceń jak:

- powtórz,
- przesuń,
- czekaj,
- jeżeli na brzegu, odbij się,
- idź do losowa pozycja,
- powiedź.

**Zadanie dodatkowe dla chętnych** w celu powtórzenia zrealizowanego materiału:

Napisz program własnej historyjki, pamiętaj użyj co najmniej 2 duszków.

Termin realizacji zadania 27.03.2020

## Klasa 6

Temat: Stosujemy instrukcję warunkową

Uruchom przeglądarce platformę [www.scratch.mit.edu](http://www.scratch.mit.edu), następnie zapoznaj się z fragmentem podręcznika zamieszczonym poniżej i wykonaj ćwiczenie 6 oraz 7.

**Zadanie dodatkowe dla chętnych** w celu powtórzenia zrealizowanego materiału:

Utwórz program, w którym duszek narysuje schodki prowadzące od lewego górnego rogu sceny do prawego dolnego rogu. Ustal szerokość stopnia na 25 kroków, a wysokość na 20. Zmień tło sceny, ustal kolor pisaka na czerwony, a jego grubość na 2. Termin realizacji 26.03.2020.

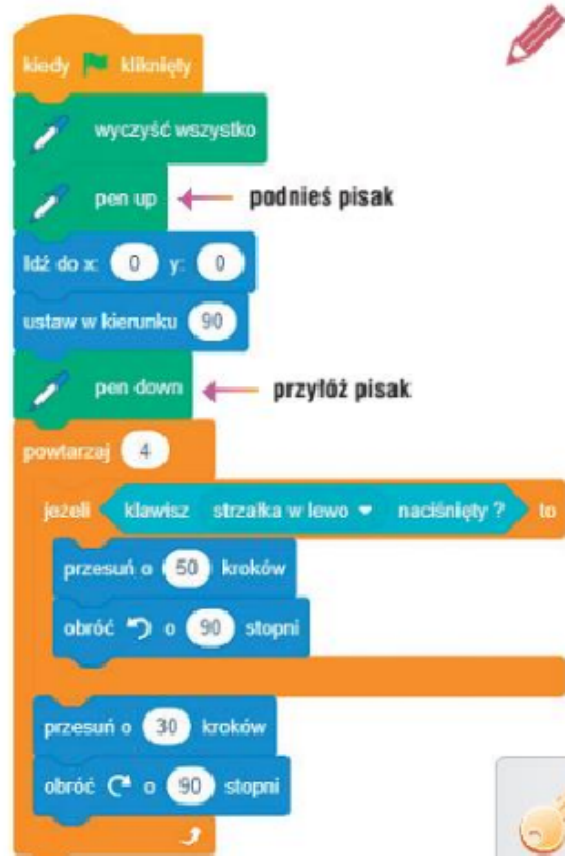
**Aby w języku Scratch zrealizować sytuację warunkową, możemy użyć polecenia warunkowego (instrukcji warunkowej) **jeżeli** reprezentowanego przez elementy:**

Jeśli *warunek* jest spełniony, realizowana jest *lista\_poleceń1* programu umieszczona po słowie **to**.

Jeśli *warunek* nie jest spełniony, realizowana jest *lista\_poleceń2* programu umieszczona po słowach **w przeciwnym razie**. Następnie realizowane jest *kolejne\_polecenie* programu.

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.

W programie pokazanym na rysunku 4. zastosowano instrukcję warunkową w wersji uproszczonej. Jeśli klawisz *strzałka w lewo* jest naciśnięty, czyli *warunek* jest spełniony, realizowane są polecenia **przesuń o 50 kroków** i **obróć (w lewo) o 90 stopni**, a następnie – polecenia **przesuń o 30 kroków** i **obróć (w prawo) o 90 stopni**. Wszystkie polecenia powtarzane są cztery razy. Jeśli klawisz *strzałka w lewo* nie jest naciśnięty, realizowane są tylko polecenia **przesuń o 30 kroków** i **obróć (w prawo) o 90 stopni**. Polecenia te powtarzane są cztery razy.



Rys. 4. Stosowanie instrukcji warunkowej w wersji uproszczonej – ćwiczenie 6.



**Ćwiczenie 6.** Stosujemy polecenie warunkowe w wersji uproszczonej i powtarzamy polecenia

1. W programie Scratch umieść w obszarze roboczym polecenia pokazane na rysunku 4.
2. Zapisz program w pliku pod nazwą *warunek 1*.
3. Uruchom program. Odpowiedz na pytanie: *Jaką figurę narysował duszek?*
4. Naciśnij i przytrzymaj klawisz strzałki w lewo i uruchom ponownie program. Odpowiedz na pytanie: *Jaką figurę teraz narysował duszek?*



**Aby wybrane polecenia były wykonywane nieskończenie wiele razy,** można zastosować polecenie **zawsze** z grupy **Kontrola**.

**Aby zakończyć działanie programu,** należy kliknąć przycisk .



**Ćwiczenie 7.** Stosujemy polecenie warunkowe i wielokrotnie powtarzamy polecenia

1. Zmodyfikuj program zapisany w ćwiczeniu 6., aby wykonanie poleceń **przesuń o 30 kroków** i **obróć (w prawo) o 90 stopni** zależało od naciśnięcia klawisza strzałki w prawo.
2. Zastąp polecenie **powtarzaj (powtórz)** poleceniem **zawsze**.
3. Zapisz program w pliku pod nazwą *warunek 2*.
4. Uruchom program. Naciskaj i puszczaj (np. naprzemiennie) klawisze strzałek w prawo i w lewo. Obserwuj efekty działania programu. Odpowiedz na pytanie: *Czym różni się rysowanie figur w programach warunek 1 i warunek 2?*

## Klasa 7

Temat: Tworzenie programu i potarzenie poleceń

Uruchom przeglądarkę platformę [www.scratch.mit.edu](http://www.scratch.mit.edu). Zapoznaj się z fragmentami podręcznika zamieszczonymi poniżej i wykonaj ćwiczenia 1, 3, 4 i 5.



**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**).

Rys. 2. Przykład skryptu



### 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 przesunąć 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.
- Niektóre polecenia możemy wydawać duszkowi, klikając wybrany element w panelu poleceń, m.in. z grup **Ruch**, **Wygląd** i **Dźwięk**.
- Aby wyczyścić scenę, należy użyć polecenia **wyczyść** 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. 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

powtórz 10 razy

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.

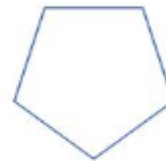


### Ćwiczenie 3. Realizujemy powtórzenia (iterację)

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?*



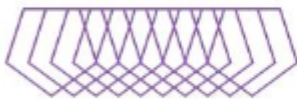
Rys. 3a. Przykład realizacji pętli prostej – ćwiczenie 3.



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



Rys. 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 *Pięciokąty*.
3. Odpowiedz na pytania: *Ile razy zostanie wykonana instrukcja przesuni o 50 kroków, a ile razy – przesuni 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*.

**Zadanie dodatkowe dla chętnych** w celu powtórzenia zrealizowanego materiału:

- Utwórz program, w którym duszek przejdzie z lewego górnego rogu sceny do prawego dolnego rogu, rysując kolorowy wzorek.
- Utwórz program, w którym duszek narysuje cztery jednakowe sześciokąty foremne – każdy umieszczony w innym rogu sceny. wymiary dobierz tak, aby po narysowaniu figury nie zachodziły na siebie.

Termin realizacji prac dodatkowych: 27.03.2020

## **Klasa 8**

Temat: Środowisko programowania Scratch

Uruchom przeglądarkę platformę [www.scratch.mit.edu](http://www.scratch.mit.edu). Zapoznaj się z fragmentami podręcznika zamieszczonym poniżej i wykonaj ćwiczenie 1, 2, 3 i 4.



**Aby utworzyć program w języku Scratch**, należy umieścić odpowiednie polecenia w obszarze tworzenia programu. Połączone ze sobą polecenia tworzą skrypt. Program może zawierać więcej niż jeden skrypt.

Skrypt uruchamiamy, klikając jego dowolny element. Jeśli na początku skryptu umieścimy polecenie **kiedy kliknięto**, będziemy mogli uruchomić skrypt (skrypty), klikając ikonę zielonej flagi.

Utworzony program możemy zapisać w pliku, następnie otworzyć go, poprawić lub uzupełnić i ponownie zapisać pod tą samą lub inną nazwą.

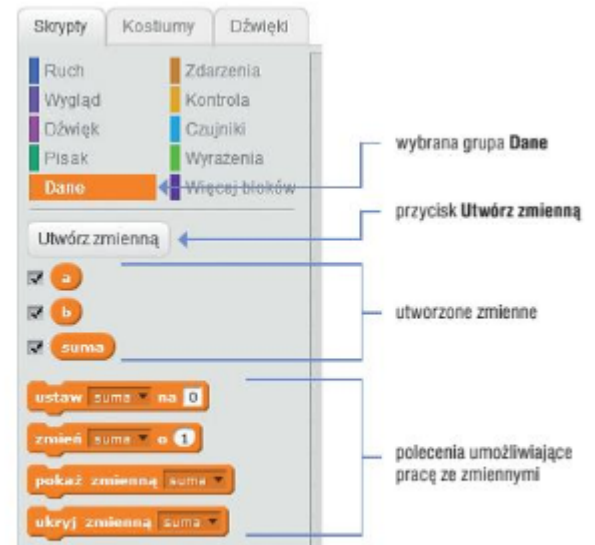
## 2.1. Stosowanie zmiennych i wykonywanie obliczeń

W języku Scratch możemy tworzyć **zmiennne**, których będziemy używali w obliczeniach. Zmiennej możemy przypisać wartość bezpośrednio w programie lub wprowadzić ją z klawiatury po uruchomieniu programu. W pierwszych ćwiczeniach będziemy wprowadzać wartość zmiennej z klawiatury, a wynik obliczeń – wyświetlać na ekranie.

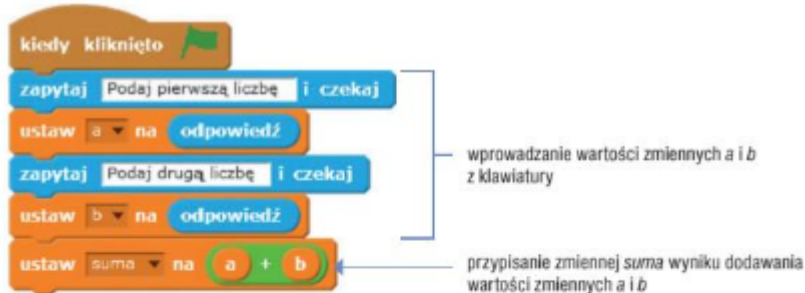


**Aby przypisać zmiennej wartość**, należy w grupie **Dane** wybrać polecenie **ustaw nazwa\_zmiennej na wartość\_zmiennej**.

**Aby wprowadzać wartość zmiennej z klawiatury**, za słowem **na** należy umieścić element **odpowiedź** z grupy **Czujniki**, a polecenie **ustaw** poprzedzić poleceniem **zapytaj**.



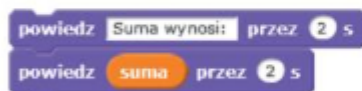
Rys. 1. Tworzenie zmiennych (Scratch) – ćwiczenie 1.



Rys. 2. Program obliczający sumę dwóch liczb (Scratch) – ćwiczenie 1.



**Aby wyprowadzić na ekran wartość zmiennej**, należy w poleceniu **powiedz** użyć jej nazwy.



Rys. 3. Polecenia wyświetlające na ekranie komunikat i wartość zmiennej suma – ćwiczenie 1.



### Ćwiczenie 1. Tworzymy zmienne i stosujemy je w obliczeniach

1. Utwórz trzy zmienne o nazwach *a*, *b* i *suma* (rys. 1.).
2. Utwórz program obliczający sumę dwóch liczb wprowadzanych z klawiatury (rys. 2.).
3. Wynik obliczeń wyprowadź na ekran. Program powinien wyświetlać również odpowiedni komunikat (rys. 3.).
4. Zapisz program w pliku pod nazwą *Obliczenia1*.

## 2.2. Stosowanie instrukcji warunkowej i iteracyjnych

Po uruchomieniu programu utworzonego w ćwiczeniu 1. otrzymujemy wynik dla jednej pary liczb. Jak zmodyfikować program z ćwiczenia 1., aby obliczał sumę dla kilku par liczb?

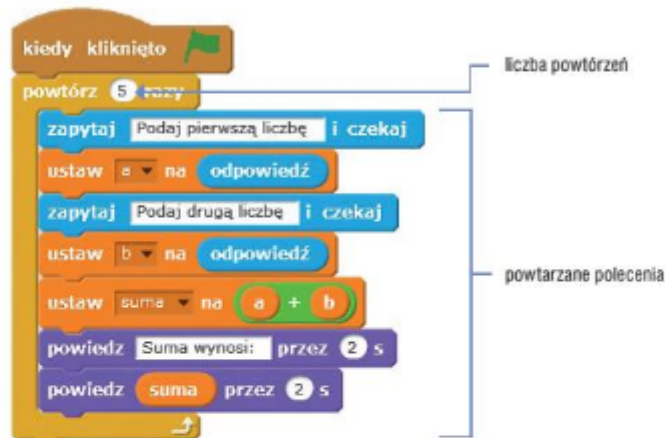


Aby w języku Scratch zapisać powtarzające się polecenia, można zastosować polecenie (instrukcję iteracyjną) **powtórz**, reprezentowane przez element **powtórz** (rys. 4.).



Rys. 4. Polecenie **powtórz** (Scratch)

Polecenia, które chcemy powtarzać, umieszczamy wewnątrz elementu **powtórz**. Liczbę powtórzeń określamy, zmieniając wartość w polu tekstowym elementu **powtórz** (rys. 5.). W polu tym możemy również umieścić zmienną, a jej wartość wprowadzać z klawiatury po każdorazowym uruchomieniu programu.



Rys. 5. Stosowanie polecenia **powtórz** (Scratch) – ćwiczenie 2.



### Ćwiczenie 2. Stosujemy polecenie **powtórz**

1. Otwórz program *Obliczenia1* zapisany w ćwiczeniu 1.
2. Zmodyfikuj program tak, aby po uruchomieniu programu obliczenia powtarzały się dla pięciu par liczb  $a$  i  $b$ .
3. Utwórz nową zmienną o nazwie *różnica*.
4. Uzupełnij skrypt pokazany na rysunku 5. o obliczanie różnicy liczb  $a$  i  $b$  oraz wyświetlanie wyniku na ekranie.
5. Zapisz program w pliku pod tą samą nazwą.

**Wskazówka:** Wewnątrz elementu **powtórz** należy dodać polecenie **ustaw** z wyrażeniem obliczającym różnicę liczb  $a$  i  $b$  oraz polecenie **powiedz** z odpowiednim komunikatem i nazwą zmiennej *różnica*.

Po uruchomieniu programu utworzonego w ćwiczeniu 2. otrzymujemy sumę i różnicę pary liczb. Jak zmodyfikować program, aby działanie było wybierane zależnie od naciśniętego klawisza?



Aby w języku Scratch zrealizować sytuację warunkową, należy zastosować polecenie warunkowe (instrukcję warunkową) **jeżeli** (rys. 6a i 6b) reprezentowane przez element **jeżeli** (rys. 7.).

### Scratch

```
jeżeli warunek to  
  lista_poleceń1  
w przeciwnym razie  
  lista_poleceń2  
kolejne_polecenie
```

Rys. 6a. Ogólna postać polecenia warunkowego (Scratch)

### Scratch

```
jeżeli warunek to  
  lista_poleceń1  
kolejne_polecenie
```

Rys. 6b. Uproszczona postać polecenia warunkowego (Scratch)



Działanie instrukcji warunkowej jest podobne w większości języków programowania. Sprawdzany jest warunek logiczny – jeśli warunek jest spełniony (jest prawdziwy), wykonywana jest instrukcja (*lista\_poleceń1*) po słowie **to**. Jeśli warunek nie jest spełniony (jest fałszywy) – wykonywana jest instrukcja (*lista\_poleceń2*) po słowach **w przeciwnym razie** (rys. 6a). Jeśli brak tej części instrukcji – wykonywana jest od razu kolejna instrukcja (*kolejne\_polecenie*) po instrukcji warunkowej (rys. 6a i 6b).

W języku Scratch instrukcję warunkową reprezentuje element **jeżeli** (rys. 7). Warunek tworzymy, korzystając z elementów z grupy **Wyrażenia**.

Rys. 7. Obliczenie sumy – tylko, gdy naciśnięto klawisz „s” (Scratch) – ćwiczenie 3.



### Ćwiczenie 3. Stosujemy polecenie warunkowe **jeżeli** w uproszczonej wersji

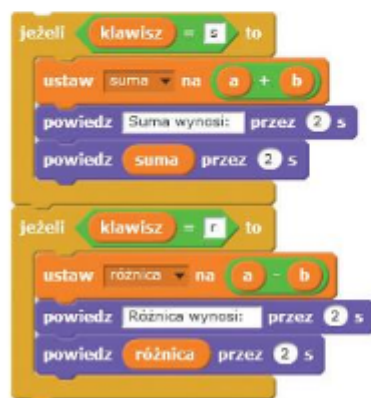
1. Otwórz program *Obliczenia1* zapisany w ćwiczeniu 2.
2. Utwórz nową zmienną o nazwie *klawisz*. Znajdź w panelu poleceń element reprezentujący polecenie **jeżeli**. Zmodyfikuj program, tak aby wykonał odpowiednie obliczenia zależnie od wprowadzonej wartości zmiennej *klawisz*: dodawanie – jeśli *klawisz* = s (rys. 7.), a odejmowanie – jeśli *klawisz* = r (rys. 8a).
3. Zapisz program w pliku pod tą samą nazwą.
4. Uruchom program i sprawdź jego działanie dla różnych danych. Wprowadzaj również litery inne niż s i r. Odpowiedz na pytanie: *Dlaczego wprowadzenie litery innej niż s i r powoduje przejście do wprowadzenia kolejnych wartości zmiennych a i b?*

Działanie pełnej wersji polecenia warunkowego (instrukcji warunkowej) **jeżeli** sprawdzimy w ćwiczeniu 4.



#### Ćwiczenie 4. Stosujemy polecenie warunkowe **jeżeli**

1. Otwórz program *Obliczenia1* zapisany w ćwiczeniu 3.
2. Zamień dwa polecenia **jeżeli** (rys. 8a) na jedno (rys. 8b).
3. Zapisz program pod nazwą *Obliczenia2*.
4. Uruchom program i sprawdź jego działanie dla różnych danych. Wprowadzaj kolejno wartości zmiennej *klawisz*: s, r i inne litery. Objaśnij działanie programu *Obliczenia2*, odpowiadając na pytania: *Jakie polecenia program wykonuje, gdy warunek jest spełniony, a jakie – gdy nie jest spełniony? Dlaczego program oblicza różnicę dla dowolnej wartości zmiennej klawisz różnej od s? Czym różni się działanie programów Obliczenia1 i Obliczenia2?*
5. Zmień komunikat w polu tekstowym polecenia **zapytaj** umieszczonego przed wprowadzaniem wartości zmiennej *klawisz* (rys. 7.), aby odpowiadał warunkom zadania.



**Rys. 8a.** Stosowanie polecenia warunkowego **jeżeli** w uproszczonej wersji (Scratch) – ćwiczenie 3.



**Rys. 8b.** Stosowanie polecenia warunkowego **jeżeli** (Scratch) – ćwiczenie 4.