

## Kompozycje, historyjki i gry w języku Scratch

1. Programujemy w języku Scratch
2. Umieszczamy duszka w określonym miejscu sceny
3. Stosujemy powtarzanie poleceń
4. Programujemy historyjkę
5. Tworzymy grę dla jednego gracza
6. Stosujemy zmienne – zliczamy punkty



### Warto powtórzyć

1. Na czym polega tworzenie programu w języku Scratch?
2. W jaki sposób programujemy powtarzające się polecenia?
3. Jak można zmienić kierunek poruszania się duszka na scenie?

## 1. Programujemy w języku Scratch

### Uwaga !

Niektóre polecenia (m.in. z grupy **Ruch** i **Dźwięk**) możemy wydawać duszkowi, klikając wybrany element w wykazie poleceń.

Polecenia języka Scratch (podobnie jak w programie Baltie) są reprezentowane przez elementy graficzne, umieszczone w dziesięciu grupach. Każda z grup jest wyróżniona kolorystycznie i dotyczy innego rodzaju czynności.

Tworzenie programu w języku Scratch polega na umieszczaniu odpowiednich poleceń (reprezentowanych przez elementy graficzne) w obszarze tworzenia programu (po wybraniu karty **Skrypty**). 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**).



**Aby uruchomić skrypt**, wystarczy kliknąć jego dowolny element.


Jeśli na początku skryptu (skryptów) umieścimy polecenie **kiedy kliknięto**

**kiedy kliknięto**



z grupy **Zdarzenia**, będziemy mogli uruchomić program zawierający ten skrypt (skrypty), klikając przycisk **Rozpocznij skrypty**

**zielonej flagi** , umieszczony w prawym górnym rogu sceny.

Działanie programu można przerwać, klikając przycisk **Stop** , umieszczony w prawym górnym rogu sceny.



## Ćwiczenie 1. Tworzymy program w języku Scratch

1. Uruchom program Scratch.
2. Ułóż polecenia pokazane na rysunku 1.
3. Pod ułożonymi poleceniami dodaj polecenie, które spowoduje, że kot zamiauczy.
4. Uruchom program.
5. Zapisz program w pliku pod nazwą *dom*.

### Wskazówki:

- W poleceniach **przesuń** i **obróć** można zmienić domyślnie wyświetloną liczbę kroków i stopni, klikając pole tekstowe i wpisując inne wartości.
- Po umieszczeniu dwóch pierwszych poleceń **obróć** i **przesuń**, warto je zduplikować, a następnie zduplikować powstałe w ten sposób cztery polecenia i... program jest prawie gotowy – wystarczy tylko w niektórych poleceniach odpowiednio zmienić liczby kroków oraz wielkości kątów.



Rys. 1. Przykład skryptu – ćwiczenie 1.



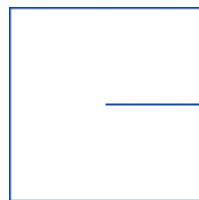
Rys. 2. Schematycznie pokazany przykład użycia poleceń obrotów: w lewo i w prawo o kąt 90°

W grupie poleceń **Ruch** umieszczono dwa rodzaje poleceń obrotów: w lewo i w prawo. Zależnie od użytego polecenia, duszek obróci się odpowiednio w lewo lub w prawo o podany kąt (rys. 2.).



## Ćwiczenie 2. Rysujemy literę

1. Utwórz program rysujący figurę pokazaną na rysunku 3., używając tylko obrotów w prawo i przesunięć z grupy **Ruch**. Jaką literę przypomina ta figura?
2. Zapisz program w pliku pod nazwą *litera*.



Rys. 3. Litera – ćwiczenie 2.

### Dobra rada

Umieszczone w obszarze tworzenia programu polecenie można zduplikować, wybierając z menu kontekstowego danego polecenia opcję **Duplikuj**. Polecenie zostanie zduplikowane razem ze wszystkim poleceniami, które są do niego dołączone.

Program Scratch umożliwia zmianę koloru i rozmiaru pisaka. Odpowiednie polecenia umieszczono w grupie **Pisak**. Do zmiany koloru pisaka możemy na przykład wykorzystać barwę występującą w oknie programu. Kolor pobieramy, stosując narzędzie **Selektor kolorów (Pobieranie kolorów)**, w podobny sposób jak w programie Paint.



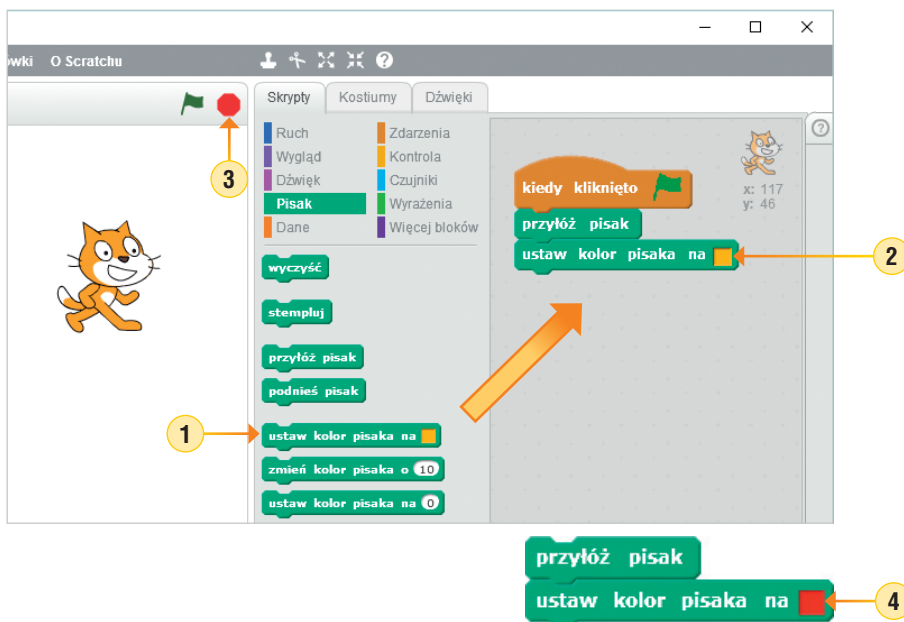
### Przykład 1. Pobieranie koloru

1. Z grupy **Pisak** wybierz polecenie **ustaw kolor pisaka na** i umieść je w obszarze tworzenia programu.
2. Kliknij pole koloru.
3. Przesuń kursor w miejsce na ekranie, z którego chcesz pobrać kolor, i kliknij.
4. W polu koloru pojawi się pobrany kolor.

**Uwaga:** Kolor pisaka możesz również zmienić, wykonując czynności opisane

w przykładzie bez umieszczania polecenia

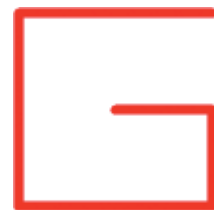
**ustaw kolor pisaka na**



### Ćwiczenie 3. Zmieniamy kolor i rozmiar pisaka

1. Otwórz plik zapisany w ćwiczeniu 2.
2. Zmodyfikuj program, aby narysować figurę pokazaną na rysunku 4. Dodaj polecenia ustawiające kolor pisaka na czerwony, a rozmiar – na 4.
3. Zapisz plik pod tą samą nazwą.

**Wskazówka:** Rozmiar pisaka możemy ustawić, stosując polecenie z grupy **Pisak**. W polu tekstowym należy zmienić odpowiednio liczbę określającą nowy rozmiar.



**Rys. 4.** Litera  
– ćwiczenie 3.

## 2. Umieszczamy duszka w określonym miejscu sceny

Położenie duszka możemy zmienić, chwytając go i upuszczając na scenie. Możemy również użyć polecenia **przesuń**. Jak jeszcze można przenieść duszka w inne miejsce sceny?

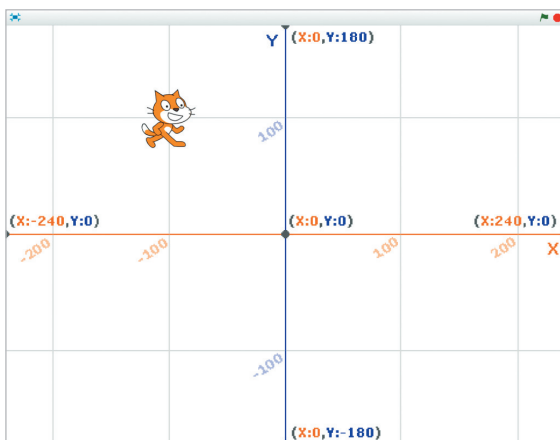
Pozycję duszka można zmienić, korzystając z poleceń określających współrzędne danego miejsca sceny. W programie Scratch współrzędna  $x$  przyjmuje wartości od -240 do 240, a współrzędna  $y$  – wartości od -180 do 180.



**Aby zmienić położenie duszka na scenie, należy użyć**

polecenia **idź do x: 0 y: 0** z grupy **Ruch**.

Na przykład polecenie **idź do x: -100 y: 100** przeniesie duszka do miejsca o współrzędnych (-100, 100). Liczba przed przecinkiem oznacza współrzędną  $x$ , a po przecinku – współrzędną  $y$  (rys. 5).



*Rys. 5. Określanie pozycji duszka z wykorzystaniem układu współrzędnych*



### Ćwiczenie 4. Zmieniamy pozycję duszka na scenie

1. Otwórz plik zapisany w ćwiczeniu 3.
2. Zmień program tak, aby duszek zawsze zaczynał rysowanie ze środka sceny i był zwrócony w prawo.
3. Zapisz plik pod tą samą nazwą.

#### Wskazówki:

- Nowe polecenia należy dodać na początku programu, po poleceniu **kiedy kliknięto**.
- Aby duszek podczas zmieniania pozycji nie rysował linii, należy użyć polecenia **podnieś pisak**.
- Jeśli chcemy, aby duszek odwrócił się w prawo, należy odpowiednio ustawić jego

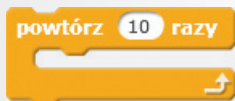
kierunek, używając polecenia **ustaw kierunek na 90** z grupy **Ruch**.

### 3. Stosujemy powtarzanie poleceń

Zamierzamy utworzyć program, w którym duśzek narysuje kompozycję z wielu kwadratów. W jaki sposób zapisać w języku Scratch powtarzanie poleceń?



Aby w języku Scratch zapisać powtarzanie poleceń, stosujemy polecenie **powtórz** (z grupy **Kontrola**).



Polecenia, które mają być powtarzane, umieszczamy wewnątrz elementu **powtórz**. W szczególnym przypadku można powtarzać polecenie **powtórz**. Liczbę powtórzeń określamy, zmieniając wartość w polu tekstowym polecenia **powtórz** (domyślnie jest w nim wpisana wartość 10).

Aby narysować kwadrat, wykonujemy czterokrotnie dwa polecenia (rysowanie boku i obrót o 90°). Dlatego w programie zastosujemy polecenie **powtórz 4 razy** (rys. 6).



Rys. 6. Polecenia rysujące kwadrat – stosujemy polecenie **powtórz**



**Ćwiczenie 5.** Rysujemy kwadrat – stosujemy polecenie **powtórz**

1. Utwórz program rysujący kwadrat o boku 100 kroków (rys. 6.).
2. Zapisz program pod nazwą *kwadrat*.



**Ćwiczenie 6.** Stosujemy zagnieżdżone polecenia **powtórz**

1. Otwórz plik *kwadrat* zapisany w ćwiczeniu 5.
2. Umieść polecenia rysujące kwadrat wewnątrz polecenia **powtórz**, tak jak pokazano na rysunku 7. Dodaj pozostałe polecenia.
3. Uruchom program. Objaśnij działanie poszczególnych poleceń programu. Odpowiedz na pytania: *Ile jest narysowanych kwadratów o boku 100 kroków? Ile razy jest wykonywane polecenie **przesuń o 100 kroków**, a ile razy **przesuń o 25 kroków**?*
4. Zapisz program pod nazwą *kwadraty1*.



Rys. 7. Program rysujący kompozycję z kwadratów – ćwiczenie 6.



## Ćwiczenie 7. Modyfikujemy program

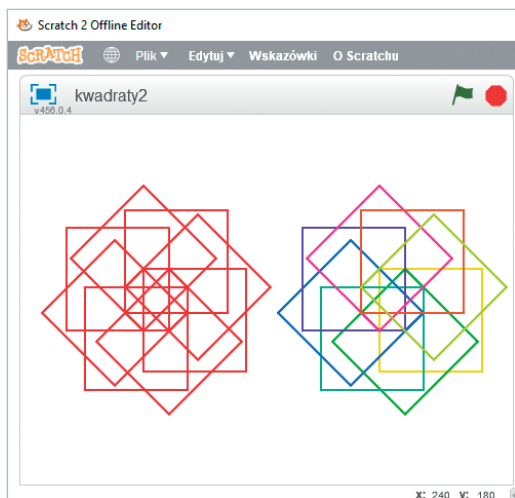
1. Otwórz plik *kwadrat* zapisany w ćwiczeniu 5.
2. Zmodyfikuj program tak, aby otrzymać efekt podobny do pokazanego na rysunku 8. W kompozycji po prawej stronie każdy kwadrat narysowano innym kolorem.
3. Zapisz program pod nazwą *kwadraty2*.

### Wskazówki:

- Na początku programu umieść polecenie **wyczyść**.
- Pamiętaj o możliwości duplikowania poleceń.
- Do narysowania kompozycji wyświetlonej po prawej stronie zastosuj w odpowiednim

miejscu programu polecenie **zmień kolor pisaka o 10**. Aby uzyskać wyraźny efekt zmiany koloru, zwiększ liczbę w polu tekstowym polecenia.

- Zmień odpowiednio położenie duszka przed rysowaniem każdej z kompozycji, stosując polecenie **idź do**.
- Użyj poleceń **podnieś pisak** i **przyłóż pisak**, aby duszek nie rysował niepotrzebnych linii.

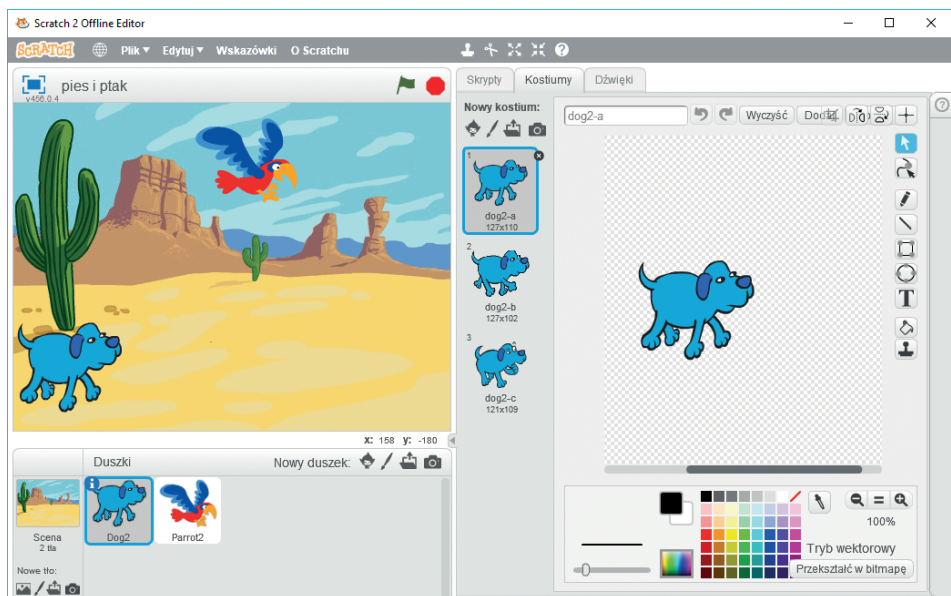


*Rys. 8. Efekt wykonania programu – ćwiczenie 7.*

## 4. Programujemy historijkę


Chcemy utworzyć program, w którym scena zostanie zamieniona na obraz pustyni, po której będzie biegał tam i z powrotem niebieski pies, a po niebie będzie latał kolorowy ptak (rys. 9.). W jaki sposób dodać nowe duszki, wyświetlić je na nowym tle, a następnie wprowadzić w ruch?


W programie Scratch duszek może przyjmować różne postacie. Można wybrać postać z przygotowanych w programie, zmodyfikować istniejącą postać lub narysować nową, korzystając z edytora grafiki dostępnego na karcie **Kostiumy** (rys. 9.). Można również wprowadzić na scenę więcej niż jednego duszka i dla każdego ułożyć oddzielne polecenia. Istnieje też możliwość zmiany tła sceny.



**Rys. 9.** Przykład sceny ze zmienionym tłem i dwoma duszkami. Po prawej stronie odsłonięta karta **Kostiumy** z widocznymi kostiumami duszka-psy i edytorem grafiki wbudowanym do programu Scratch



**Aby wybrać nowe tło**, należy kliknąć przycisk  – otworzy się okno **Biblioteki tła**, z której można wybierać tła.

**Aby dodać nowego duszka**, należy kliknąć przycisk  – otworzy się okno **Biblioteki duszków**, z której można wybierać postacie.

**Aby duzek zmieniał kostiumy**, należy umieścić w programie odpowiednie polecenie z grupy **Wygląd**:

następny kostium

lub

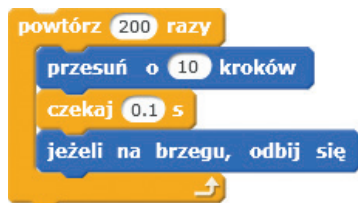
zmień kostium na

kostium2



## Ćwiczenie 8. Zmienianie tła i dodawanie postaci duszka

1. Utwórz nowy program. Zmień tło na krajobraz pustynny.
2. Dodaj duszka-psy. Usuń duszka-kota.
3. Ustaw początkowe współrzędne x i y pozycji duszka-psy tak, aby zaczynał ruch w miejscu pokazanym na rysunku 9.
4. Duszek-psy ma poruszać się po scenie od lewej do prawej strony i z powrotem.
5. Powtarzaj 200 razy przesuwanie się duszka-psy o 10 kroków, przy czym ruchy duszka nie powinny być zbyt szybkie i nie powinien on „uciekać” poza scenę.
6. Zapisz program w pliku pod nazwą *na pustyni*.



**Rys. 10.** Zastosowanie poleceń **czekaj i jeżeli na brzegu, odbij się** – ćwiczenie 8.



### Wskazówki:


- Aby ruchy duszka były wolniejsze, zastosuj polecenie **czekaj** z grupy **Kontrola** (rys. 10.). W polu tekstowym możesz wpisać liczbę dziesiętną, którą w programie Scratch zapisuje się z kropką (np. 0.3, 0.2). Im mniejsza liczba, tym szybsze będą ruchy.
- Aby pies „nie uciekał” za scenę, zastosuj polecenie **jeżeli na brzegu, odbij się** z grupy **Ruch** (rys. 10.).



### Dobra rada

Po umieszczeniu duszka na scenie możemy zwiększyć jego rozmiar

– kliknąć przycisk **Zwiększ** , a następnie klikać postać duszka.

Kliknięcie przycisku **Zmniejsz** , a następnie klikanie postaci duszka, zmniejsza jego rozmiar.

Żeby duszek-pies nie chodził do góry nogami, można zmienić styl obrotów duszka.



### Przykład 2. Ustawianie stylu obrotu duszka

1. Uaktywnij (kliknij) duszka w obszarze **Duszki** – postać powinna być otoczona niebieską ramką.
2. Kliknij przycisk umieszczony w górnym lewym rogu ikony aktywnego duszka.
3. Wybierz odpowiedni styl obrotów, np. dwustronny (prawo-lewo).
4. Sprawdź ustawiony kierunek i ewentualnie popraw go.



### Uwagi:

- Aby duszek poruszał się w poziomie, kierunek powinien być ustawiony na 90°.
- Ustawienia kierunku ruchu duszka można zmieniać, obracając (po kliknięciu) niebieski wskaźnik na tarczy kierunku.
- Aktualnie wybrany styl obrotów jest wyróżniony kolorem niebieskim.



### Ćwiczenie 9. Zmieniamy kostiumy i ustawiamy styl obrotów duszka

1. Otwórz plik zapisany w ćwiczeniu 8. Zmodyfikuj program:
  - a) usuń kostium oznaczony literą c,
  - b) spraw, aby duszek nie chodził do góry nogami – korzystając z przykładu 2., ustaw styl obrotów dwustronny (prawo-lewo).
  - c) zmieniaj kostiumy duszka-psa, tak aby duszek w trakcie przejścia przebierał nogami – umieść polecenie **następny kostium** wewnątrz polecenia **powtórz**.
2. Zapisz plik pod tą samą nazwą.



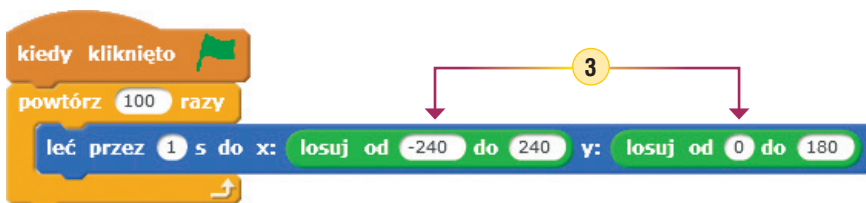
Chcemy dodać duszka, który będzie przemieszczał się w sposób losowy po wybranym obszarze sceny.



### Przykład 3. Losowe przemieszczanie się duszka po scenie

1. Z grupy **Ruch** wybierz polecenie **leć przez 1 s do x: 0 y: 0**.
2. Z grupy **Wyrażenia** wybierz element **losuj od 1 do 10**.
3. Do pól tekstowych (**x**: i **y**): polecenia **leć** wstaw polecenia **losuj** i wpisz odpowiednie wartości, które mają być generowane losowo przez program (**od... do...**).

**Uwaga:** Jeśli dla współrzędnej  $x$  wpiszemy wartości od -240 do 240, a dla współrzędnej  $y$  – od 0 do 180, to duszek będzie poruszał się po całej szerokości górnej części sceny.



#### Uwaga



W obszarze tworzenia programu układamy skrypt dla aktywnego duszka. Aby rozpocząć tworzenie skryptu dla danego Duszki, należy w obszarze **Duszki** uaktywnić (kliknąć) tego duszka.



### Ćwiczenie 10. Dodajemy drugiego duszka, poruszającego się losowo po scenie

1. Otwórz plik zapisany w ćwiczeniu 9.
2. Dodaj duszka-papugę. Zmniejsz rozmiar postaci.
3. Korzystając z przykładu 3., utwórz skrypt, w którym duszek-papuga będzie latał losowo po górnej części sceny. Ustaw prędkość lotu na pół sekundy (wpisz liczbę 0.5). Wszystkie polecenia powtarzaj 200 razy.
4. Duszek-papuga w trakcie lotu powinien zmieniać kostiumy. Dodaj odpowiednie polecenia wewnątrz polecenia **powtórz**.
5. Zapisz plik pod tą samą nazwą.

## 5. Tworzymy grę dla jednego gracza

Zamierzamy utworzyć grę, w której wezmą udział dwa duszki: ballerina i piłka. Duszek-piłka będzie poruszał się losowo po całej scenie, a duszkiem-balleriną będziemy sterować, naciskając klawisze strzałek. Jak zaprogramować możliwość decydowania, w jakim kierunku ma się przemieścić duszek?

W języku Scratch kolejność wykonywania poleceń może zależeć od zaistnienia określonych zdarzeń, np. naciśnięcia wybranego klawisza na klawiaturze. Ta możliwość jest przydatna podczas tworzenia gier komputerowych.

W naszej grze duszek-ballerina będzie się poruszał po scenie zależnie od naciśniętego klawisza. Mamy więc do czynienia z **sytuacjami warunkowymi**.



Aby w języku Scratch zrealizować sytuację warunkową, możemy użyć polecenia **jeżeli** z grupy **Kontrola**.



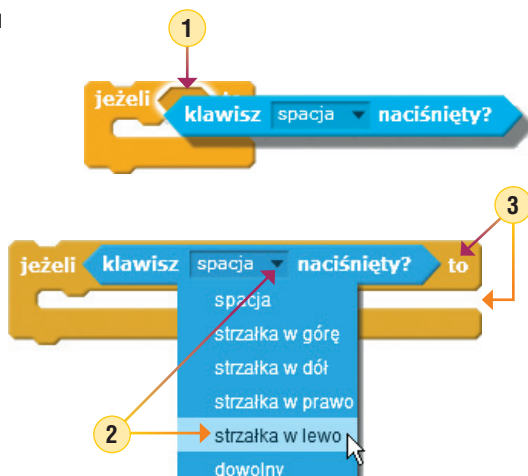
Jeśli **warunek** jest **spełniony**, realizowana jest **lista poleceń** programu umieszczona po słowie **to**. Następnie (niezależnie od spełnienia **warunku**) realizowane jest **kolejne polecenie** programu.

**Uwaga:** Jest to uproszczona wersja polecenia warunkowego.



#### Przykład 4. Stosowanie polecenia warunkowego w wersji uproszczonej

1. Umieść **warunek** w polu warunku – **warunkiem** może być wybrane zdarzenie z grupy **Czujniki**, np. **klawisz spacja naciśnięty?**
2. Rozwiń listę wyboru polecenia (naciskając strzałkę przy domyślnym klawiszu **spacja**) i wybierz np. **strzałka w lewo**.
3. Po słowie **to** umieść polecenia, które mają być wykonywane, jeśli **warunek** jest spełniony.



#### Ćwiczenie 11. Sterujemy duszkiem w czterech kierunkach

1. Utwórz program, w którym scena zostanie zamieniona na wybrany obraz, a duszek-ballerina będzie poruszał się po scenie w prawo, w lewo, w górę i w dół, zależnie od naciśniętych klawiszy strzałek.
2. Wszystkie polecenia powtarzaj nieskończenie wiele razy.
3. Zapisz program w pliku pod nazwą *ballerina z piłką*.

##### Wskazówki:

- Tło sceny pokazane na rysunku 11. zostało wybrane z gotowych tła, a następnie inaczej pokolorowane w edytorze grafiki wbudowanym do programu Scratch.
- Na rysunku 12. pokazana jest część programu, w której określono, że duszek-ballerina porusza się w górę i w dół po naciśnięciu odpowiednich klawiszy strzałek. Aby dodać kolejne polecenia, skorzystaj z możliwości duplikowania poleceń, a następnie zmień odpowiednio **warunki** i ustawienia kierunków.
- Aby duszek-ballerina nie chodził bokiem, ani do góry nogami, pamiętaj o odpowiednim ustawieniu stylu obrotów duszka (przykład 2.).



Rys. 11. Przykładowa scena gry – ćwiczenia 11. i 12.



Rys. 12. Pierwsza część skryptu – ćwiczenie 11.



**Ćwiczenie 12.** Piszemy skrypt dla duszka poruszającego się losowo po całej scenie

1. Otwórz plik zapisany w ćwiczeniu 11.
2. Dodaj duszka-piłkę i utwórz dla niego skrypt, w którym duszek będzie poruszał się nieskończenie wiele razy w losowy sposób po całej scenie.
3. Zapisz plik pod tą samą nazwą.

**Wskazówka:** Zastosuj polecenie **leć** i umieść je w pętli **zawsze**.

## 6. Stosujemy zmienne – zliczamy punkty

{ Chcemy zliczać punkty, gdy duszki się dotkną. W jaki sposób wprowadzić do gry zliczanie punktów? }

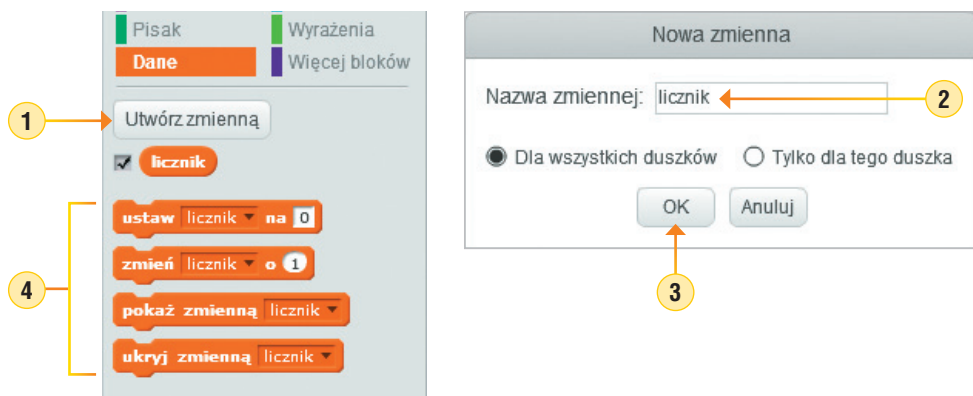
W programie Scratch można korzystać ze zmiennych. W naszej grze utworzymy zmienną o nazwie *licznik*, dzięki której będziemy pamiętać liczbę zdobytych punktów.

Liczba zdobytych punktów będzie zależała od spełnienia warunku (tu: dotknięcia się duszków), dlatego zastosujemy również polecenie warunkowe.



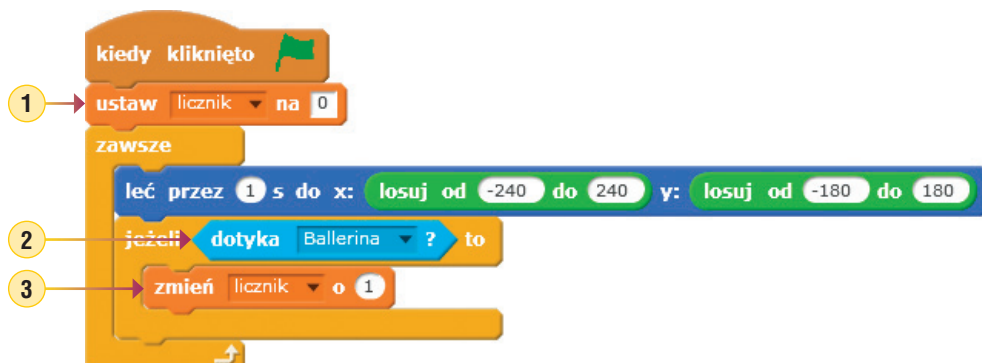
#### Przykład 5. Tworzenie nowej zmiennej w programie

1. W grupie **Dane** na karcie **Skrypty** kliknij przycisk **Utwórz zmienną** – otworzy się okno dialogowe **Nowa zmienna**.
2. W polu tekstowym **Nazwa zmiennej** wpisz nazwę zmiennej, np. *licznik*.
3. Kliknij przycisk **OK**.
4. Utworzona zmienna i polecenia dotyczące jej stosowania pojawią się w panelu poleceń.



#### Przykład 6. Zliczanie punktów zależne od warunku

1. Aby punkty w grze zliczały się od zera, wyzeruj na początku programu zmienną *licznik*, stosując polecenie **ustaw licznik na 0**.
2. Aby zliczać punkty, gdy duszki się dotkną, do polecenia warunkowego wstaw warunek **dotyka ?** z grupy **Czujniki** i z listy wyboru wybierz nazwę danego duszka.
3. Jeśli warunek jest spełniony, powinny zliczać się punkty. Dodaj polecenie **zmień licznik o 1**.





### Ćwiczenie 13. Zliczamy punkty w grze

1. Otwórz plik zapisany w ćwiczeniu 12. Zmodyfikuj program tak, aby gracz otrzymywał jeden punkt, gdy duszki się dotkną.
2. Korzystając z przykładu 5., utwórz zmienną o nazwie *licznik*.
3. Korzystając z przykładu 6., uzupełnij skrypt dla duszka-piłki.
4. Zapisz plik pod tą samą nazwą.
5. Uruchom program i obserwuj na scenie zmieniającą się wartość zmiennej *licznik*.

**Uwaga:** *Licznik* po wykonaniu polecenia **ustaw licznik na 0** jest równy zero.

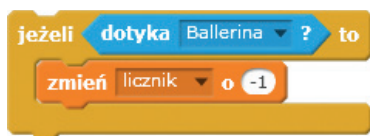
Wykonanie pierwszy raz polecenia **zmień licznik o 1** spowoduje, że *licznik* przyjmie wartość 1. Kolejne wykonanie polecenia **zmień licznik o 1** powoduje, że *licznik* zwiększa się o jeden, czyli będzie przyjmował wartości: 2, 3, 4 itd.



### Ćwiczenie 14. Dodajemy trzeciego duszka i wprowadzamy odejmowanie punktów



**Rys. 13a.** Przykładowy trzeci duszek – ćwiczenie 14.



**Rys. 13b.** Polecenie warunkowe dla duszka-nietoperza – ćwiczenie 14.

1. Otwórz plik zapisany w ćwiczeniu 13.
2. Dodaj duszka-nietoperza (rys. 13a), który ma się poruszać losowo po całej scenie, podobnie jak piłka.
3. Wprowadź warunek odejmowania punktów w grze – gdy duszek-ballerina i duszek-nietoperz dotkną się, licznik ma zmniejszać się o 1 (rys. 13b).
4. Zapisz program w pliku pod nazwą *gra trzy duszki*.

**Wskazówka:** Polecenia dla duszka-nietoperza są bardzo podobne do poleceń dla duszka-piłki, dlatego możesz zdublować polecenia duszka-piłki i kopię przeciągnąć na duszka-nietoperza w obszarze **Duszki**. Następnie wprowadź niezbędne poprawki i usuń zerowanie zmiennej *licznik* w skrypcie duszka-nietoperza.



### Warto zapamiętać

- W języku Scratch program tworzymy, układając polecenia w obszarze tworzenia programu (na karcie **Skrypty**). Polecenia wykonuje postać zwana duszkiem. Polecenia reprezentowane są przez elementy graficzne umieszczone w dziesięciu grupach. Polecenia można łączyć, tworząc tzw. skrypty.
- W języku Scratch możemy zapisywać powtarzające się polecenia, stosując polecenie **powtórz**.
- W języku Scratch możemy umieścić na scenie więcej niż jednego duszka i dla każdego napisać oddzielny skrypt.
- W języku Scratch kolejność wykonywania poleceń może zależeć od zaistnienia określonych zdarzeń, np. naciśnięcia wybranego klawisza na klawiaturze.
- W języku Scratch sytuację warunkową realizujemy, stosując polecenie warunkowe **jeżeli**.
- W języku Scratch można stosować zmienne. Zmienną należy wcześniej utworzyć.



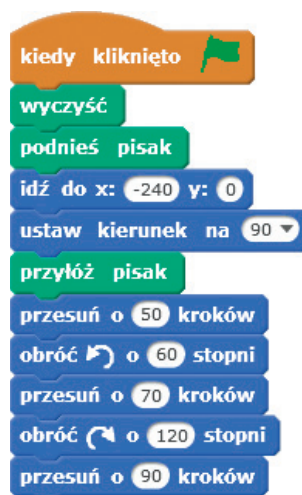
## Pytania

1. Wyjaśnij na przykładzie, na czym polega tworzenie programu w języku Scratch.
2. W jaki sposób można uruchomić program utworzony w języku Scratch?
3. Wyjaśnij na przykładzie sposób zmiany koloru i rozmiaru pisaka.
4. W jaki sposób można umieścić duszka w pozycji określonej współrzędnymi (120, -100)?
5. Wyjaśnij na przykładzie, w jaki sposób w języku Scratch można zapisać powtarzające się polecenia.
6. W jaki sposób można zmienić tło sceny?
7. W jaki sposób można zmienić postać duszka?
8. Na czym polega zmiana kostiumu duszka? Wyjaśnij na przykładzie.
9. Wyjaśnij na przykładzie, w jaki sposób realizuje się sytuację warunkową w języku Scratch. Podaj przykłady zdarzeń, które mogą być warunkami w poleceniu **jeżeli**.
10. W jaki sposób w języku Scratch tworzy się zmienną? Omów na przykładzie.

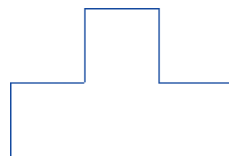


## Zadania

1. Ułóż polecenia pokazane na rys. 14. Uruchom skrypt. Zmień wartości kąta i liczby kroków, aby otrzymać inny obraz. Dodaj dodatkowe polecenia **przesuń** i **obróć** – według własnego pomysłu. Uruchom program. Zapisz program w pliku pod nazwą *łamana*.
2. Utwórz program, w którym duszek narysuje figurę pokazaną na rysunku 15. Zapisz program w pliku pod nazwą *figura*.
3. Utwórz program, w którym duszek narysuje (schematycznie) pierwszą literę twojego imienia. Zastosuj swój ulubiony kolor i pogrubioną linię. Zapisz program w pliku, który nazwiesz swoim imieniem.
4. Utwórz program, w którym duszek narysuje figurę według twojego pomysłu. Zastosuj kolory i zmianę grubości linii. Zapisz program w pliku pod nazwą *figura1*.
5. Utwórz program rysujący trójkąt równoboczny o boku długości 80 kroków w kolorze zielonym. Zapisz program w pliku pod nazwą *trójkąt*.
6. Utwórz program rysujący kompozycję składającą się z trójkątów równobocznych według własnego pomysłu (możesz wzorować się na ćwiczeniu 7.). Zapisz program w pliku pod nazwą *trójkąty*.
7. Otwórz plik *dom* zapisany w ćwiczeniu 1. Dodaj polecenia rysujące płotek składający się z takich samych sześciu sztachet. Zapisz program w pliku pod nazwą *dom z płotem*.



Rys. 14. Program – zadanie 1.



Rys. 15. Efekt wykonania programu – zadanie 2.





**Rys. 16.** Przykładowa scena – zadanie 10.

8. Otwórz plik *na pustyni* zapisany w ćwiczeniu 8. Dodaj po prawej stronie postać lwa, który będzie poruszał się powyżej psa i w przeciwnym niż on kierunku. Zapisz plik pod tą samą nazwą.
9. Otwórz plik *na pustyni* zapisany w zadaniu 8. Pokoloruj postacię psa i lwa. Zapisz plik pod tą samą nazwą.
10. Utwórz program, w którym po scenie zamienionej na las będzie galopował tam i z powrotem koń i latał (poruszając się losowo) nietoperz. Na rysunku 16. pokazana jest przykładowa scena. Ruch duszka-konia powinien zacząć się z lewej strony. Zmieniaj kostiumy postaci i ewentualnie rozmiary duszków. Ruchy postaci powtarzaj po 200 razy. Zapisz program w pliku pod nazwą *koń i nietoperz*.
11. Utwórz program, w którym po scenie podobnej do pokazanej na rysunku 17a będą poruszać się losowo cztery duszki: trzy kolorowe piłki w dolnej części sceny i motyl w górnej części sceny. Zmieniaj kostiumy piłek (rys. 17b). Zapisz program w pliku pod nazwą *piłki i motyl*.

**Wskazówki:**

- Aby dodać trzy takie same duszki-piłki, dla których ma być ułożony taki sam skrypt, wystarczy dwukrotnie zduplikować pierwszego duszka-piłkę (z menu kontekstowego duszka wybrać polecenie **duplikuj**).
  - Aby duszki-piłki zmieniały kostiumy w innej kolejności, można zmienić kolejność kostiumów (rys. 17b), przeciągając i upuszczając dany kostium w inne miejsce.
12. Utwórz grę, w której wezmą udział dwa duszki. Jeden z duszków ma poruszać się losowo po całej scenie. Drugim duszkiem będziemy sterować (w lewo, w prawo, w górę, w dół), naciskając wybrane klawisze. Jeśli duszki się dotkną, otrzymujemy 10 punktów. Wybierz samodzielnie tło i postacię duszków (możesz również zmodyfikować istniejące postaci lub narysować nowe). Zapisz program w pliku pod nazwą określającą tematykę gry.

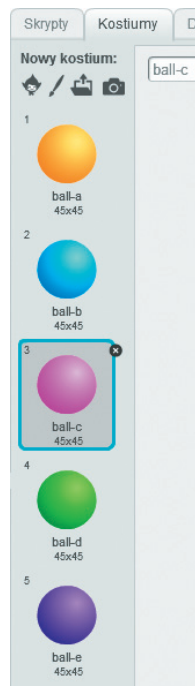
**Wskazówka:** Własne sceny, tworzone np. w programie Paint, zapisz w plikach z rozszerzeniem *png*.

13. Otwórz plik zapisany w zadaniu 12. Dodaj trzeciego duszka, który ma poruszać się losowo po całej scenie. Jeśli nowy duszek i duszek sterowany klawiszami dotkną się, tracimy 10 punktów. Zapisz plik pod tą samą nazwą.





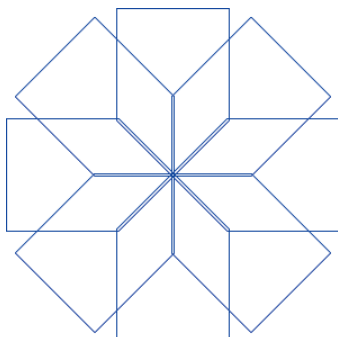
**Rys. 17a.** Przykładowa scena – zadanie 11.



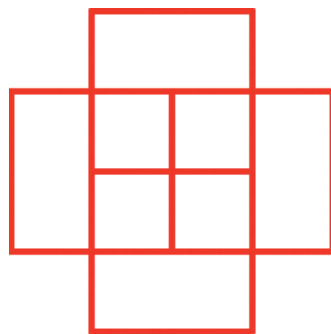
**Rys. 17b.** Przykładowe kostiumy duszka-piłki – zadanie 11.

### Dla zainteresowanych

14. Stosując polecenie **powtórz**, zmodyfikuj program zapisany w ćwiczeniu 1., aby powstała kompozycja pokazana na rysunku 18.
15. Stosując polecenie **powtórz**, zmodyfikuj program zapisany w ćwiczeniu 5., aby powstała kompozycja pokazana na rysunku 19.



**Rys. 18.** Kompozycja – zadanie 14.



**Rys. 19.** Kompozycja – zadanie 15.

16. Utwórz program, w którym duszek narysuje wymyśloną przez siebie kompozycję. Zapisz program w pliku pod nazwą *moja kompozycja*.

17. Utwórz program, w którym na scenie podobnej do pokazanej na rysunkach 20a i 20b po chodniku będzie chodził tam i z powrotem duszek-chłopiec, a ulicą będzie jeździł duszek-samochód, zachowując zasadę ruchu prawostronnego. Ruch obydwu duszków powinien zacząć się z lewej strony sceny. Powtarzaj ruchy wszystkich postaci nieskończenie wiele razy. Zapisz program w pliku pod nazwą *droga*.

**Wskazówki:**

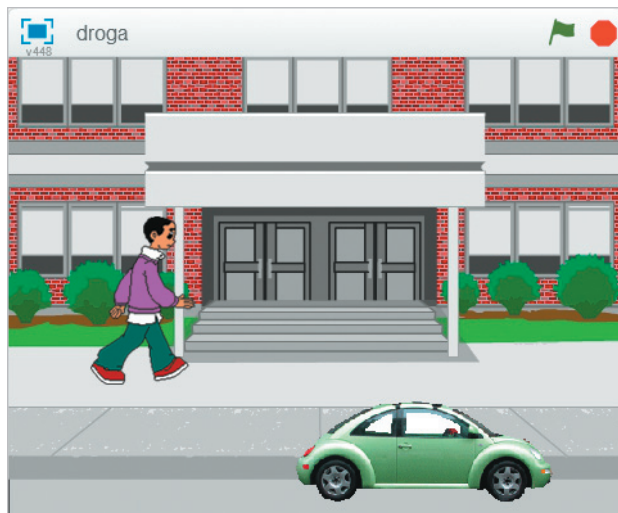
- Dla duszka-samochodu zastosuj do poruszania się po jezdni polecenie

leć przez 1 s do x: 0 y: 0, ustalając odpowiednio współrzędne x i y.

- Do zmiany kierunku jazdy zastosuj polecenie

ustaw kierunek na 90

**Rys. 20a.** Przykładowa scena (ruch w prawo) – zadanie 17.



**Rys. 20b.** Przykładowa scena (ruch w lewo) – zadanie 17.



18. Utwórz grę według własnego pomysłu. Korzystając z **Pomocy**, zapoznaj się samodzielnie z innymi możliwościami programu Scratch. Uwzględnij nowe funkcje w swojej grze. Zapisz program w pliku pod nazwą *moja gra*.