

3.4 Zakręt za zakrętem

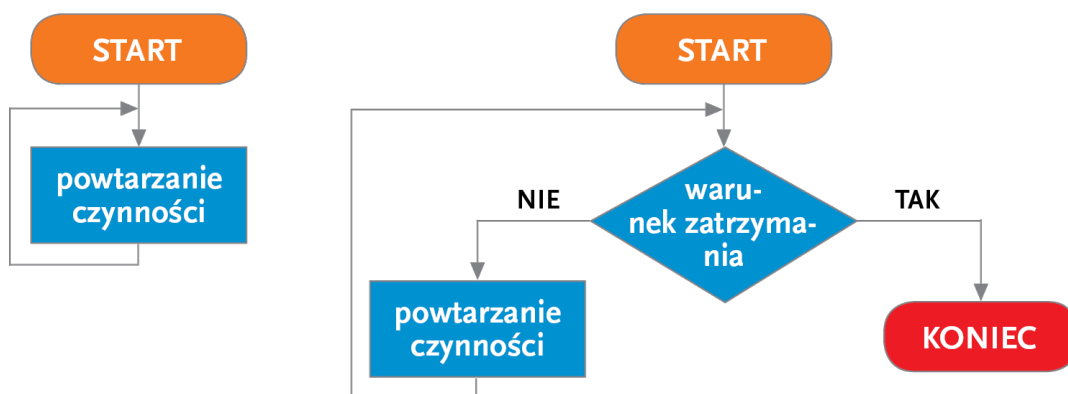
DOWIESZ SIĘ

- czym jest rekurencja,
- w jaki sposób tworzy się skrypty rekurencyjne,
- w jaki sposób wykorzystywać algorytmy rekurencyjne.

Rekurencja umożliwia łatwe rozwiązywanie trudnych zadań. Będziesz z niej korzystał wielokrotnie. Teraz zapoznasz się z działaniem pierwszego skryptu rekurencyjnego.

CO TO JEST REKURENCJA?

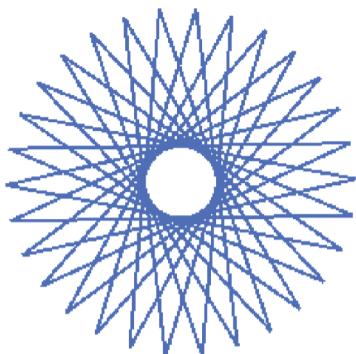
Spróbuj wytłumaczyć, jak wykonywać jakąś czynność wymagającą wielu powtórzeń, np. jak tańczyć tango. Dwa długie wolne kroki, dwa krótkie szybkie kroki. Potem cały proces się powtarza. Na tym właśnie polega rekurencja. To sposób opisywania powtarzających się czynności. W gruncie rzeczy można powiedzieć, że cały świat jest w pewnej mierze rekurencyjny – zmywanie naczyń, jedzenie zupy, przechodzenie po pasach przez ulicę, zbieranie rozsypanych kartek... Nie zawsze łatwo da się określić, ile razy powtórzyć polecenia. Opisujesz więc pewną procedurę, nie kończąc jej definiowania. Stosujesz procedury wywołujące same siebie.



Rys. 1. Schemat rekurencji bez końca i z warunkiem zatrzymania

RYSOWANIE GWIAZD

Figura przedstawiona na rys. 2 powstała wskutek wielokrotnego wykonania bloków **przesuń o 200 kroków i obróć w prawo o 156 stopni**. Ile razy? Dostatecznie wiele. Można pracowicie policzyć, ile potrzeba skrętów, aby uzyskać pełną gwiazdę, ale wygodniej jest zostawić tę część pracy komputerowi.



Rys. 2. Gwiazda o kącie skrętu 156°

Najprostszy plan rysowania gwiazdy, zapisany za pomocą języka naturalnego, wygląda następująco:

- przejście naprzód o długość boku,
- skręt w prawo o wartość kąta,
- powtórzenie tych samych czynności dostatecznie wiele razy.

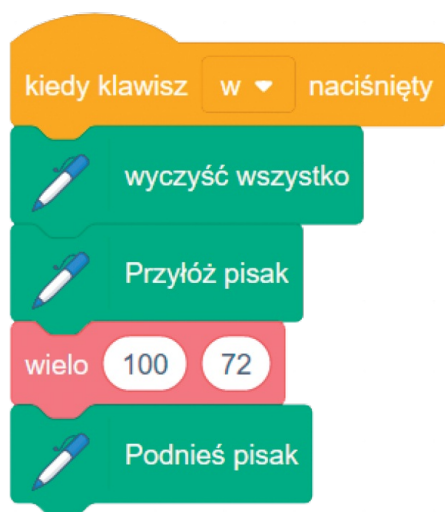
Zrealizuj ten plan w Scratchu.

- Utwórz nowy blok o nazwie **wielo**, kliknij przycisk **Dodaj dane wejściowe liczba lub tekst** i dodaj dwa parametry liczbowe – **bok** i **kąt**.



Rys. 3. Definicja i wywołanie bloku **wielo**

- Utwórz skrypt, który po naciśnięciu klawisza **w** będzie odpowiadać za rysowanie – dodaj grupę bloków **Pióro** i przed wywołaniem bloku **wielo** wstaw bloki **wyczyść wszystko** (w ten sposób przy każdym uruchomieniu programu duszek będzie rysować w pustym oknie) oraz **Przyłóż pisak**. Jeśli chcesz, rozszerz skrypt o ustawienie właściwości pisaka.

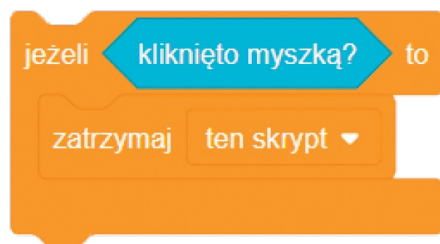


Rys. 4. Skrypt rysowania z blokiem **wiele**

JAK ZATRZYMAĆ REKURENCJĘ?

Istnieje pewne ryzyko stosowania rekurencji: jeśli nie zostanie określony warunek zakończenia działania, to będzie ona działać w nieskończoność. W przypadku tańca warunek jest prosty: należy skończyć, gdy przestanie grać muzyka. Blok **wiele** nie ma określonego warunku zatrzymania – trzeba go zatem wstawić. Niech duszek kończy rysowanie po naciśnięciu lewego przycisku myszy.

- Dodaj do skryptu blok **jeżeli (...) to (...)** opisujący następującą sytuację: jeśli zostanie naciśnięty lewy przycisk myszy, to należy zatrzymać wykonywanie skryptu. Umieść go tuż po nagłówku bloku.



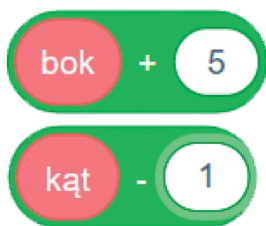
Rys. 5. Blok **jeżeli (...) to (...)**

- Przetestuj działanie skryptu – czy duszek zatrzymał się po naciśnięciu lewego przycisku myszy?
- Czy potrafisz zaplanować takie kąty skrętu, dla których powstaną ładne gwiazdy? Narysuj kilka gwiazd, żeby wypróbować różne kąty skrętu. Podaj również kąt 90 stopni. Co zostanie narysowane?

JAK ZMIENIĆ REKURENCJĘ?

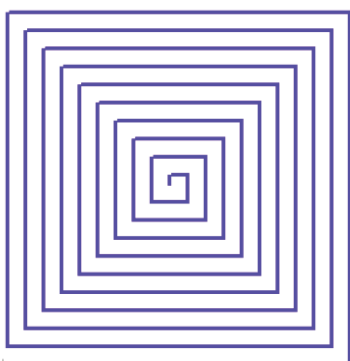
Rekurencję można oczywiście zmodyfikować – wystarczy zmienić wartości parametrów w bloku rekurencyjnym.

- Zmień wartość parametrów w wywołaniu rekurencyjnym bloku **wiele** – niech **bok** zwiększa się o 5, a **kąt** maleje o 1.



Rys. 6. Zmiana parametrów **bok** i **kąć** w skrypcie

Skoro za pomocą bloku **wielo** można narysować kwadrat, to może uda się utworzyć również figurę przedstawioną na rys. 7?



Rys. 7. Labirynt

Ponieważ w tym wypadku długość boku się zmienia, trzeba wprowadzić pewne zmiany w bloku. Rysowanie należy zacząć od najkrótszego boku, np. o długości 5. Następny bok, rysowany po skręcie, powinien być o 5 dłuższy od poprzedniego i tak dalej, aż do narysowania najdłuższego boku, który dotknie krawędzi. Są tu elementy charakterystyczne dla budowy skryptu rekurencyjnego:

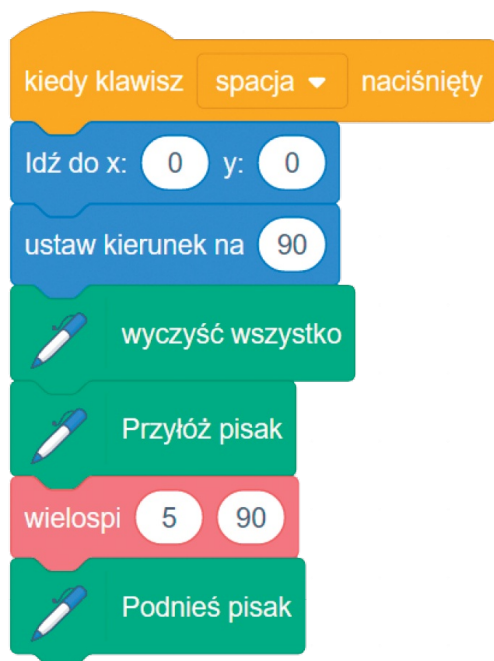
- warunek zakończenia: „aż do zetknięcia z krawędzią”,
- kroki do wykonania powtarzane rekurencyjnie: narysowanie boku i wykonanie skrętu o podany kąt.

Skąd jednak skrypt będzie wiedział, że ma narysować kolejny bok dłuższy od poprzedniego? To proste – blok rysuje bok, którego wartość podaje się w jego wywołaniu. Jeśli przy każdym kolejnym wywołaniu bok ma być dłuższy o 5 od poprzedniego, to należy zwiększać długość boku w wywołaniu rekurencyjnym.

- Utwórz rekurencyjny blok **wielospi** z parametrami **bok**, **kąć**.

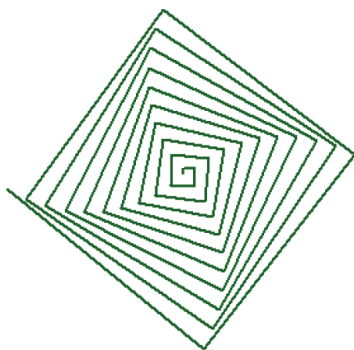
Rys. 8. Blok **wielospi**

- Utwórz skrypt działający po naciśnięciu klawisza spacji – oprócz bloków odpowiadających za rysowanie i wywołania bloku rekurencyjnego dodaj bloki **Idź do x: (...) y: (...)** i **ustaw kierunek na (...)**, tak aby duszek zaczynał rysowanie w tym samym punkcie okna i tak samo ustawiony. Wybierz parametry wywołania: bok **5**, kąć **90**.

Rys. 9. Skrypt uruchamiający blok **wielospi**

- Zwolnij działanie duszka, aby przyjrzeć się jego kolejnym ruchom – dodaj przed wywołaniem rekurencyjnym w bloku **wielospi** blok **czekaj 0.5 sekund**.

- Umieść blok **czekaj** za wywołaniem rekurencyjnym – tym razem duszek znów rysuje bardzo szybko, ale ma kłopoty z zakończeniem działania. Czy to możliwe, że blok dodany po wywołaniu rekurencyjnym zostanie wykonany?
- Przetwórz blok **czekaj** na właściwe miejsce i dodaj za nim blok **zmień rozmiar o 5**. Jak teraz wykonywany jest skrypt? Jak powiększył się duszek?
- Wywołuj blok **wielospi** dla różnych kątów.



Rys. 10. Figura nakreślona po wywołaniu bloku **wielospi**

W projekcie znajdującym się pod adresem <https://scratch.mit.edu/projects/342957512> możesz obejrzeć wiele różnych figur tworzonych przez blok **wielospi**.

ZADANIA

1. Zmodyfikuj blok **wielospi** tak, aby długość boku była niezmienna, a zmieniał się kąt skrętu duszka. Bok nie powinien być zbyt długi. Dla jakich wartości kątów skrypt powinien kończyć rysowanie?
2. Zmodyfikuj blok **wielospi** tak, aby jego parametrami były: **bok**, **kąt** i **skok**. Po wywołaniu rekurencyjnym **bok** powinien zwiększać się o wielkość **skoku**. Jakie ryzyko towarzyszy takiemu rozwiązaniu?