

## Koncepcje dotyczące programowania oraz umiejętności, których uczy Scratch

Tworząc interaktywne historyjki, gry oraz animacje przy użyciu Scratch młodzi ludzie zdobywają umiejętności związane z programowaniem oraz mają szansę zapoznać się z podstawowymi koncepcjami dotyczącymi tworzenia programów.

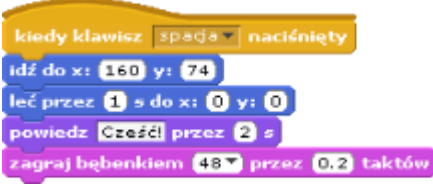

### Jak Scratch uczy rozwiązywać problemy i stworzyć projekt ?

- rozwija umiejętność logicznego rozumowania
- dziecko uczy się jak usunąć błędy z programu (debugging)
- daje możliwość prześledzenia procesu powstawania programu – od pierwszych wyobrażeń aż do zakończenia projektu
- wymaga ciągłego skupienia i wytrwałości

### Podstawowe koncepcje dotyczące komputerów i programowania

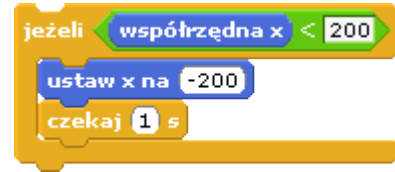
- program 'mówi' komputerowi dokładnie co ma zrobić, krok po kroku
- stworzenie programu komputerowego nie wymaga wybitnych umiejętności, najważniejsze jest logiczne myślenie oraz staranność

### Pozostałe idee dotyczące programowania

Idea	Wyjaśnienie	Przykład
kolejność	Tworząc program w Scratch musisz najpierw uważnie przemyśleć kolejność poszczególnych kroków.	
iteracja (pętle)	Do powtarzania ciągu instrukcji możemy użyć poleceń <b>zawsze</b> oraz <b>powtórz __razy</b> . Powtarzanie fragmentów programu nazywamy iteracją.	

instrukcje warunkowe

Dzięki **jeżeli** oraz **jeżeli-w\_przeciwym\_wypadku** możemy sprawdzić czy warunek jest spełniony.



zmienne



Blok **zmienne** pozwala na stworzenie zmiennych oraz użycie ich w programie. W zmiennych możemy przechowywać liczby oraz ciągi znaków. W Scratch masz do dyspozycji zarówno zmienne globalne jak i zmienne dostępne wyłącznie dla poszczególnych obiektów. Przy użyciu bloku **lista** możesz przechowywać oraz mieć dostęp do listy liczb oraz łańcuchów znaków. Ten rodzaj struktury danych może być uznany za “tablicę dynamiczną”.

listy (tablice)

obsługa zdarzeń

**kiedy\_klawisz\_naciśnięty** oraz **kiedy\_kliknięto\_duszek** są przykładami obsługi zdarzeń – odpowiedzi na zdarzenia, które zostały wywołane przez użytkownika lub inną część programu



wątki (współbieżność)

Uruchomienie dwóch ciągów instrukcji w tym samym czasie sprawia, że dwa wątki są wykonywane jednocześnie.



koordynacja i synchronizacja

**nadaj** i **czekaj** umożliwiają koordynację akcji dla wielu duszków jednocześnie. Synchronizację zapewnia **nadaj\_i\_czekaj**.

Na przykład Duszek1 wysłał wiadomość *zwycięzca*, kiedy został spełniony warunek:

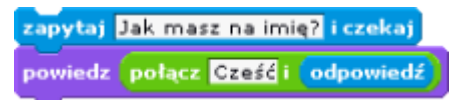


Kiedy Duszek2 dostaje wiadomość zostaje uruchomiony ten fragment skryptu:



dane z klawiatury

**zapytaj\_i\_czekaj** zachęca użytkowników do pisania na klawiaturze. **odpowiedź** przechowuje wpisane dane. **losuj\_liczbę** zwraca losową wartość z podanego zakresu.



losowe liczby



wyrażenia logiczne

Przy użyciu **i**, **lub**, **nie** tworzymy wyrażenia logiczne.



dynamiczna interakcja

**x\_myszy**, **y\_myszy** oraz **głośność** mogą być użyte do stworzenia interakcji w czasie rzeczywistym.



projektowanie interfejsu użytkownika

W Scratch możesz stworzyć interaktywne interfejsy użytkownika – na przykład używając duszków, które można kliknąć lub tworząc przyciski.



**Koncepcje dotyczące programowania, których Scratch obecnie nie uczy:**  
procedury i funkcje; przekazywanie parametrów oraz zwracanie wartości; rekurencja;  
definiowanie klas obiektów; dziedziczenie; obsługa wyjątków; odczyt plików/zapis do pliku.  
<http://scratch.mit.edu>