

Spiralny rozwój umiejętności programowania

Od układania bloków po zapis programu

Uwagi ogólne

Poniższe propozycje zajęć informatycznych mają na celu spiralny rozwój umiejętności uczniów w zakresie programowania. W tych propozycjach korzysta się z różnych środowiskach, w których może przebiegać nauka programowania. Zamieszczam także wskazówki metodyczne i organizacyjne zajęć, jednak ostateczna decyzja w jakim zakresie i jak przeprowadzić takie zajęcia należy do nauczyciela.

Przedstawiona propozycja może przerodzić w plany zajęć, elastycznie dostosowane do potrzeb programowych przedmiotu (nauczyciela), potrzeb i zainteresowań uczniów. Zajęcia mogą być rozciągnięte na kilka jednostek lekcyjnych, uczniowie mogą wykonywać polecane im zadania w dowolnym czasie, sprawdzenie aktywności może odbywać się na podstawie przedstawionych przez nich rozwiązań w określonych przez nauczyciela terminach, rozwiązania uczniów powinny być wynikiem ich zaangażowania i kreatywności, mogą być także wynikiem współpracy między uczniami.

O **stronie technicznej zajęć** piszę w punkcie 2 blogu. Przeprowadzenie proponowanych zajęć nie wymaga szczególnych rozwiązań technologicznych, obecnie dostępne są wystarczające.

Przygotowanie do zajęć

1. Na komputerach uczniów powinien być dostęp do Internetu i powinny być zainstalowane środowiska programowania w językach Scratch i Python.
2. W propozycjach poniżej uczniowie korzystają także ze środowiska Godziny Kodowania. Tutaj <http://mmsyslo.pl/2019/11/28/godzina-kodowania-2019/> jest krótkie wprowadzenie do tego środowiska.

Przygotowanie uczniów

Zakres tematyczny proponowanych zajęć obejmuje programowanie w różnych środowiskach. Główny nacisk jest położony na kształcenie abstrakcyjnego myślenia ponad konkretnym środowiskiem, czego dowodem ma być umiejętność uczniów przeniesienia rozwiązania z jednego środowiska do drugiego.

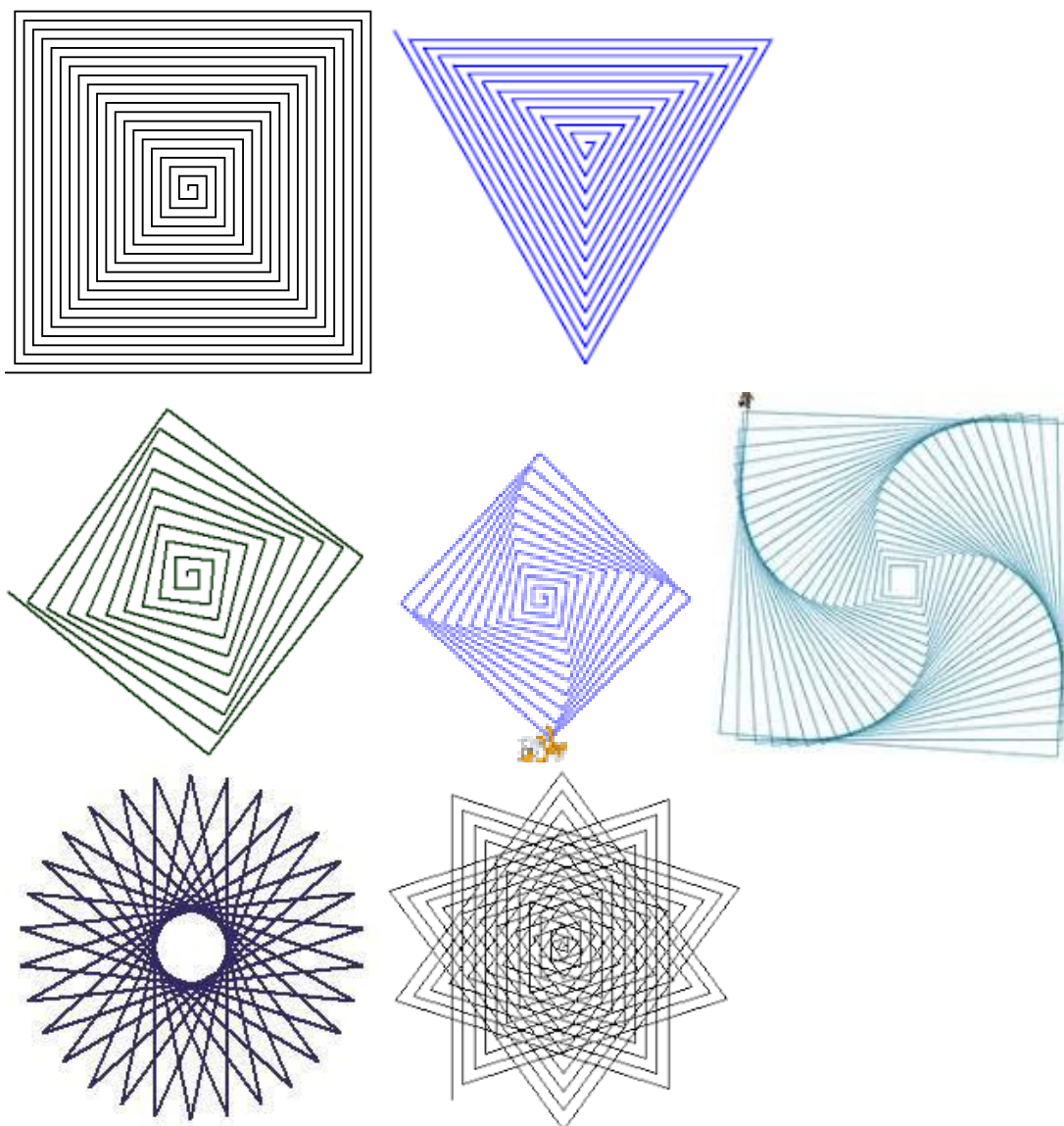
Konkretne plany lekcji – scenariusze zajęć utworzone przez nauczycieli powinny uwzględniać przygotowanie uczniów do pracy w środowiskach, których dotyczą scenariusze.

Dwa moduły

Każdy z modułów składa się z dwóch części, każda w innym środowisku, powiązanych zadaniem, jakie uczeń ma wykonać.

A. Od Scratcha do Pythona

Scratch. Utwórz program w Scratchu, w którym (widoczny lub niewidoczny) Duszek będzie rysował jedną ze spiral przedstawionych na rysunku.



Python. Korzystając z biblioteki `turtle`, napisz program w języku Python, który będzie tworzył spiralę, którą wcześniej tworzyłeś w Scratchu.

Wskazówka. Zauważ, że w blokach Scratcha są słownie zapisane polecenia. Widzisz je zapewne napisane po polsku. Możesz jednak przełączyć język na angielski, by zobaczyć te napisy po angielsku, a więc w języku, w którym są formułowane polecenia w języku Python. To może Ci pomóc.

B. Od Godziny Kodowania do Pythona

Godzina Kodowania. Wybierz łamigłówkę, w której Anna i Elsa rysują na lodzie różne figury. Możesz ją wybrać ze strony: <https://code.org/learn>. Postaraj się poprawnie wykonać kolejne etapy tej łamigłówki, zwłaszcza etap 12.

Python. Korzystając z biblioteki `turtle`, napisz program w języku Python tworzący płatek śniegu, który wcześniej tworzyłeś wraz z Anną i Elszą na etapie 12 – płatek jest tworzony z rombów.

Wskazówka. Zauważ, że w blokach tej łamigłówki są słownie zapisane polecenia. Widzisz je zapewne napisane po polsku. Możesz jednak przełączyć język na angielski, by zobaczyć te

napisy po angielsku, a więc w języku, w którym są formułowane polecenia w języku Python. To może Ci pomóc.

Uwagi dla nauczyciela

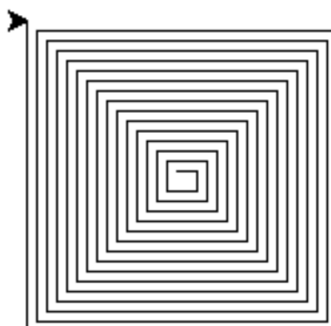
1. Powyższe propozycje dotyczą programowania w różnych środowiskach, z którymi uczniowie spotykają się na zajęciach z informatyki w szkole.
2. Główny nacisk jest położony na kształcenie abstrakcyjnego myślenia ponad konkretnym środowiskiem, czego dowodem ma być umiejętność uczniów przeniesienia rozwiązania z jednego środowiska do drugiego, co wymaga najpierw zrozumienia programu i efektów jego działania w tym pierwszym środowisku.
3. W tych propozycjach zakłada się, że uczniowie dość biegle tworzą programy w środowisku wizualno-blokowym, takim jak: Scratch, Blockly, Godzina Kodowania, czy w podobnych środowiskach związanych z robotami.
4. Jeśli uczniowie mają niewielkie doświadczenie w programowaniu w języku Python, to jako wprowadzenie można polecić im pochylenie się nad gotowymi rozwiązaniami (dwa przykładowe zamieszczamy poniżej) i poprosić ich o:
 - najpierw zrozumienie tych rozwiązań, np. z ilu kwadratów składa się płatek;
 - uruchomienie przykładowych rozwiązań dla różnych parametrów;
 - następnie – o własną modyfikację, można przy tym zasugerować, na czym miałyby polegać ta modyfikacja;
 - na końcu – o wykonanie podobnych konstrukcji według własnego pomysłu, nie wzorując się na podanym rozwiązaniu.
5. Ważne, by propozycje w zadaniach inicjowały aktywność uczniów według ich własnych pomysłów, pobudzały ich kreatywność.

Spirala w języku Python (bok=50, n=30):

```
from turtle import *

def dwa_boki(size):
    forward(size)
    right(90)
    forward(size)
    right(90)

def prosta_spirala(bok,n):
    bok=10
    for i in range(n):
        dwa_boki(bok)
        bok=bok+5
```



Płatek z kwadratów w języku Python (n=50):

```
from turtle import *  
  
def kwadrat(size):  
    for n in range(4):  
        forward(size)  
        right(90)  
  
def płek_kwadratów(n):  
    for i in range(n):  
        right(360/n)  
        kwadrat(100)
```

