

E-NAUCZANIE JAKO METODA PRACY Z UCZNIEM ZDOLNYM

Wanda Jochemczyk, Katarzyna Olędzka, Agnieszka Samulska
Ośrodek Edukacji Informatycznej i Zastosowań Komputerów
02-026 Warszawa, ul. Raszyńska 8/10
{wanda.jochemczyk, katarzyna.oledzka,
agnieszka.samulska}@oeiizk.waw.pl

Abstract. Online courses in programming for students have been organized since 2005. The main goal is to support them in gaining new knowledge in algorithmics and programming. They are conducted on the Moodle. Depart from improving computer literacy, students build new experience on communication with others with similar interests and learning on online courses.

1. Wstęp

Szkolenia online dla uczniów zainteresowanych algorytmiką i programowaniem są prowadzone w Ośrodku Edukacji Informatycznej i Zastosowań Komputerów od 2005 r. Są pomyślane jako wsparcie uczniów w rozwijaniu pasji informatycznych. Uczniowie różnych szkół województwa mazowieckiego uczą się wspólnie, wymieniając się doświadczeniami. Nie tracąc czasu na dojazdy, w dogodnym dla siebie miejscu i czasie, zmagają się z zadaniami programistycznymi. Nauczyciele, którzy towarzyszą uczniom, otrzymują wsparcie i pomysły na zajęcia dodatkowe. Szkolenia przez wiele lat funkcjonowały w oparciu o język Logo, który był wykorzystywany w konkursach informatycznych organizowanych przez OEiizK jako wiodący język do nauki programowania w szkołach podstawowych. Obecnie staramy się propagować programowanie w języku Python.

Python to powszechnie używany język programowania wysokiego poziomu. Charakteryzuje się czytelnością kodu, jego funkcje i metody umożliwiają pisanie jasnych programów zarówno w małej, jak i dużej skali. Stawiamy na programowanie w Pythonie z kilku powodów.

- Po pierwsze, jest rozwijany jako projekt Open Source i jest bezpłatny, a jego interpretery są dostępne dla wielu systemów operacyjnych – na przykład Windows, GNU/Linux, MacOS. To bardzo ważne dla edukacji, w ten sposób za-

chęcemy uczniów i nauczycieli do korzystania z bezpłatnego oprogramowania umożliwiającego pracę w różnych systemach operacyjnych.

- Po drugie, Python nie wymusza jednego stylu programowania, można stosować różne podejścia: strukturalne, obiektowe i funkcjonalne. Cechuje go prosta i przejrzysta składnia, a stosowanie intencacji (wcięć) wymusza dobre praktyki programistyczne.
- Po trzecie, jest szeroko stosowany zarówno w edukacji, nawet uczniów szkół podstawowych, jak i w profesjonalnych systemach.

Szkolenia dla uczniów prowadzimy na platformie <https://konkursy.oeiizk.edu.pl>. W roku szkolnym 2018/19 przeprowadziliśmy następujące szkolenia:

- Pierwsze kroki w Pythonie – otwarte przez cały rok szkolny dla wszystkich zainteresowanych uczniów szkolenie podstaw programowania z wykorzystaniem modułu Turtle.
- Grafika żółwia w Pythonie – obejmuje około 60 godzin pracy własnej,
- Obliczenia i algorytmy – obejmuje około 40 godzin pracy własnej,
- W głąb algorytmiki – obejmuje około 40 godzin pracy własnej,

Za szkoleniami w formie online przemawia wiele argumentów. Kluczowym jest dotarcie do jak największego grona odbiorców. Chcemy dać szansę na możliwość systematycznego kształcenia uczniom z odległych miejsc województwa mazowieckiego. Ponadto masowy udział uczniów wymusza taką formułę, zarówno od strony organizatorów, jak i uczestników. Wymagamy od uczniów biorących udział w naszych szkoleniach otwartości na wiedzę i dociekliwości, systematyczności i wytrwałości w uczeniu się, przestrzegania zasad kultury osobistej oraz uczciwości i szanowania pracy innych.

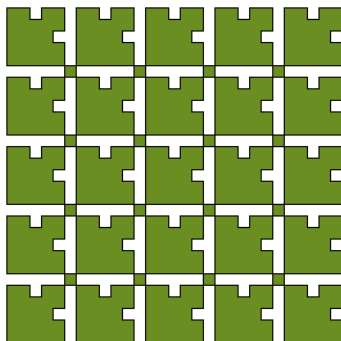
2. Grafika żółwia, czyli moduł Turtle



Rysunek 1 Eklasa w zakresie grafiki żółwia

Podczas kursu Pierwsze kroki w Pythonie młodzi ludzie uczą się programować w języku Python rozwiązując zadania z grafiki żółwia, Poznają podstawowe polecenia

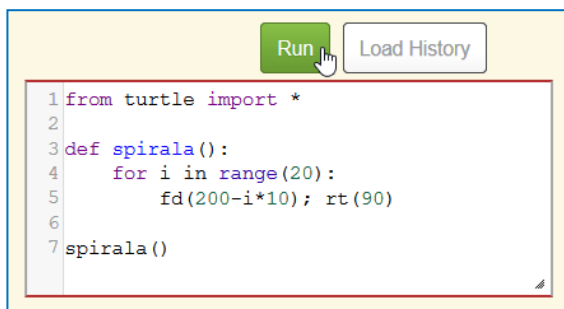
sterujące ruchem żółwia oraz instrukcję iteracji. Mają okazję nauczyć się definiować funkcje. Jest to wstęp do dalszej nauki programowania.



Rysunek 2 Posadzka z parametrem określającym liczbę kolumn

Kontynuacją nauki jest szkolenie Grafika żółwia w Pythonie. Szkolenie zaczyna się od instrukcji iteracji z wykorzystaniem pętli **for** oraz zagnieżdżonego powtarzania. Potem następuje przejście do rysowania posadzek, piramid i innych rysunków, w których liczba elementów i charakterystyka jest opisywana w pętli. Uczniowie rozwiązują zadania poprzez dzielenie ich na mniejsze problemy – definiują funkcje, korzystają ze zmiennych, stosują instrukcje warunkowe i iteracji.

Na platformie Moodle przygotowaliśmy dla uczniów ćwiczenia interaktywne korzystając z technologii Runestone Interactive. Uczeń widzi polecenie, najczęściej fragment kodu, a jego zadaniem jest uzupełnić program, a następnie uruchomić. Dzięki temu uczniowie mogą wykonywać ćwiczenia bezpośrednio na stronie internetowej, widząc od razu efekt wywołania. Zachęca to do eksperymentowania i sprzyja jakości kształcenia.



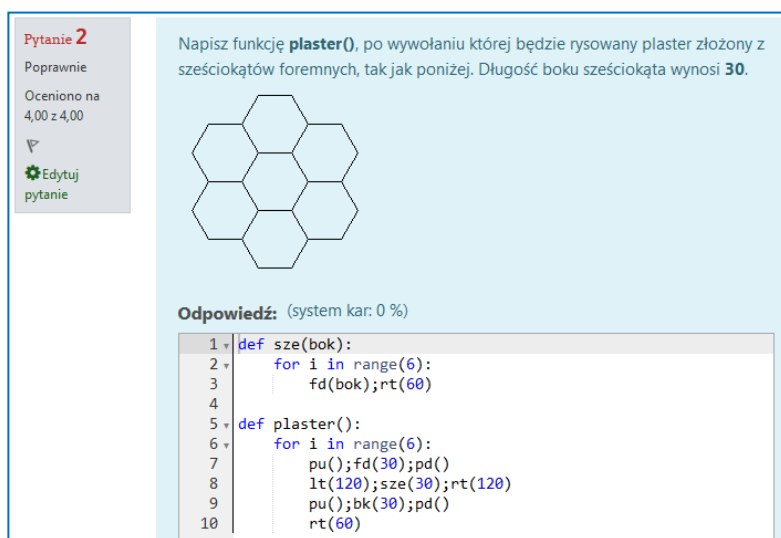
```
1 from turtle import *
2
3 def spirala():
4     for i in range(20):
5         fd(200-i*10); rt(90)
6
7 spirala()
```

Rysunek 3 Ćwiczenie interaktywne z wykorzystaniem Runestone Interactive

Wykorzystujemy też CodeRunner, darmową wtyczkę typu open-source do Moodle, która pozwala sprawdzać kod programu przesłany przez uczniów poprzez jego uru-

chomienie. Na potrzeby szkolenia zostały przygotowane zadania w formie quizów polegające na napisaniu programu, który jest sprawdzany automatycznie. W odpowiedzi na każde pytanie programistyczne uczniowie publikują swój kod i od razu widzą wyniki testów. Mogą wtedy poprawić swój program i ponownie wysłać rozwiązanie.

Warto zauważyć, że automatycznie sprawdzane są także zadania graficzne. Stosowana wtyczka dla rozwiązań z wykorzystaniem modułu Turtle została wykonana w Pracowni Edukacji Informatycznej i Kształcenia na Odległość w Ośrodku przez Katarzynę Olędzką. Uczniom ułatwia to naukę poprzez natychmiastową informację zwrotną dotyczącą wysłanego rozwiązania, a nam prowadzącym to szkolenie pozwala oszczędzić czas.



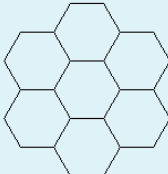
Pytanie 2

Poprawnie

Oceniono na 4,00 z 4,00

Edytuj pytanie

Napisz funkcję `plaster()`, po wywołaniu której będzie rysowany plaster złożony z sześciokątów foremnych, tak jak poniżej. Długość boku sześciokąta wynosi **30**.



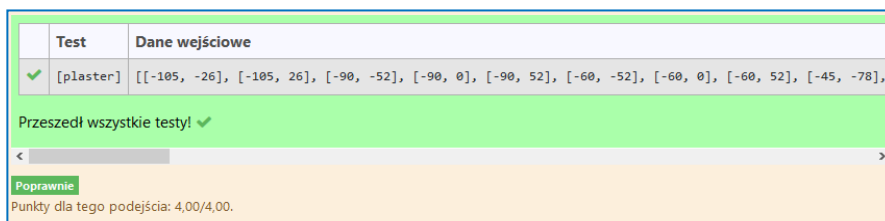
Odpowiedź: (system kar: 0 %)

```

1 def sze(bok):
2     for i in range(6):
3         fd(bok);rt(60)
4
5 def plaster():
6     for i in range(6):
7         pu();fd(30);pd()
8         lt(120);sze(30);rt(120)
9         pu();bk(30);pd()
10        rt(60)

```

Rysunek 4 Pytanie quizu i odpowiedź ucznia.



Test	Dane wejściowe
✓ [plaster]	[[[-105, -26], [-105, 26], [-90, -52], [-90, 0], [-90, 52], [-60, -52], [-60, 0], [-60, 52], [-45, -78],

Przeszedł wszystkie testy! ✓

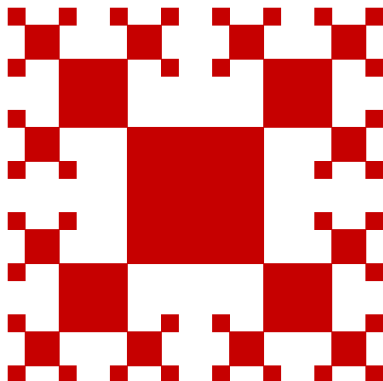
Poprawnie

Punkty dla tego podejścia: 4,00/4,00.

Rysunek 5 Odpowiedź zwrotna na wysłane rozwiązanie

W jednym z modułów szkolenia wprowadzamy rekurencję jako metodę rozwiązywania zadań graficznych. Podpatrując przyrodę, uczeń odkrywa, że jeśli umie naryso-

wać kawałek kalafiora, to może narysować go całego. Stąd łatwiej może zrozumieć, że rekurencja polega na rozwiązywaniu problemu w rozbiciu na mniejsze problemy.



Rysunek 6 Dywan – typowe zadanie rekurencyjne

W każdym module są umieszczane zadania dodatkowe, dla zdolniejszych uczniów, rozwiązanie których nie wpływa na zaliczenie całego szkolenia, ale jest szansą do rozwiązywania nietypowych problemów. Często są to zadania otwarte i nieschematyczne.

Poniższe zadanie wymaga narysowania losowej trasy opisanej w treści zadania. Ponadto należy obliczyć liczbę możliwych tras, co nie jest zadaniem trywialnym.

Zdefiniuj funkcję **trasa(n)**, która dla dowolnej wartości parametru n nie mniejszej niż **1** i nie większej niż **6**:

- rysuje na ekranie mapę rzędu n ,
- wykreśla na niej w sposób losowy trasę **WNESW**.

Funkcję należy napisać w taki sposób, aby każda trasa **WNESW** mogła być efektem jej wykonania. Oblicz, ile jest różnych tras **WNESW** na mapach rzędu **1, 2, ..., 6**? Odpowiedź przedstaw w tabelce i uzasadnij pisemnie. Jako rozwiązanie prześlij plik ***.py** i dowolny plik tekstowy.

Rysunek 7 Zadanie dodatkowe – wykorzystanie losowości

3. Obliczenia i algorytmy

Kolejne szkolenie poświęcone jest rozwiązywaniu zadań obliczeniowych oraz dotyczących przetwarzania napisów i niezagnieżdżonych list. W pierwszym module uczniowie mieli za zadanie definiować proste funkcje obliczeniowe i zamieniać słowo na obrazek. Prawidłowo wykonane obliczenia są niezbędne do skalowania tworzonych rysunków. Wprowadzenie instrukcji warunkowej umożliwiło generowanie grafiki, której wygląd zależy od przetwarzanego napisu lub liczby.



Rysunek 8 Kwadraty – graficzna reprezentacja cyfr liczby 1223333889

Poznanie podstawowych operacji na napisach dało możliwość prostych zabaw związanych ze znajdowaniem elementów lub przetwarzaniem ciągów znaków.

Pytanie 3

Poprawnie

Oceniono na
4,00 z 4,00

🚩

Napisz jednoparametrową funkcję `powiel`, której daną może być dowolne słowo składające się z par litera-cyfra (np. `'a5z3x0y2'`), a wynikiem - słowo składające się wyłącznie z liter, w którym każdej kolejnej parze litera-cyfra danego słowa odpowiada ciąg tylu identycznych liter, jaka jest wartość cyfry występującej bezpośrednio po literze.

Na przykład:

Test	Wynik
<code>powiel('a3n5z1')</code>	<code>aaannnnnz</code>

Odpowiedź: (system kar: 0%)

```

1 def powiel(s):
2     wynik=''
3     i=0
4     while i<len(s):
5         wynik+=s[i]*int(s[i+1])
6         i+=2
7     return wynik

```

Rysunek 9 Funkcja powiel – przetwarzanie ciągu znaków

W tym module sprawdzana była również umiejętność korzystania z dokumentacji języka Python. Zadaniem uczniów jest wyszukać i opisać własnymi słowami ciekawe polecenie wraz z przykładem jego użycia.

Niezwykle ważna jest umiejętność zapisania w języku formalnym tego, co rozumiemy intuicyjnie. Trzeci moduł poświęcony był między innymi logice. Zadania stają się coraz bardziej złożone, dlatego naturalnym wydaje się dzielenie problemu na pod-problemy, czyli stosowanie funkcji pomocniczych. Wprowadzone zostały listy jako struktury danych, które mogą przechowywać różne wartości. Jednym z problemów algorytmicznych było zliczanie elementów.

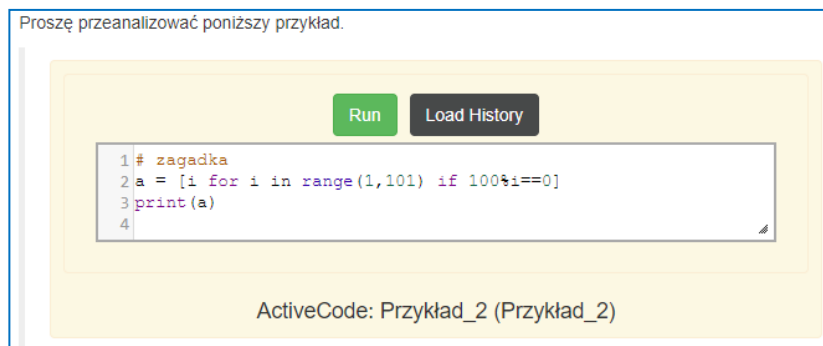
Ostatni moduł szkolenia poświęcony był algorytmowi Euklidesa obliczania największego wspólnego dzielnika dwóch liczb oraz problemowi testowania rozwiązań.

4. W głąb algorytmiki

Ostatnie szkolenie w cyklu poświęcone jest następującym zagadnieniom:

- struktury danych,
- reprezentowania informacji w komputerze,
- szyfrowanie informacji,
- złożoność algorytmów.

Uczniowie, podobnie jak w poprzednich szkoleniach, mają okazję eksperymentować w ramach ćwiczeń interaktywnych.



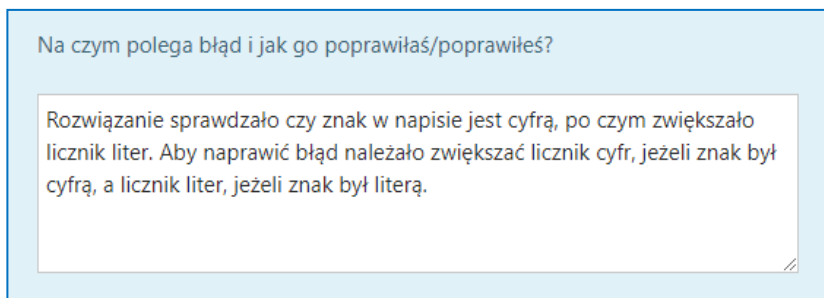
Rysunek 10 Fragment interaktywnego materiału z technologii Runestone Interactive

Na platformie znalazły się też odwołania do innych miejsc w sieci – interaktywnych ćwiczeń na portalu Informatyka dla Jasia i Joasi¹ czy Akademii Khana².

¹ <http://jasijoaasia.edu.pl/>

² <https://pl.khanacademy.org>

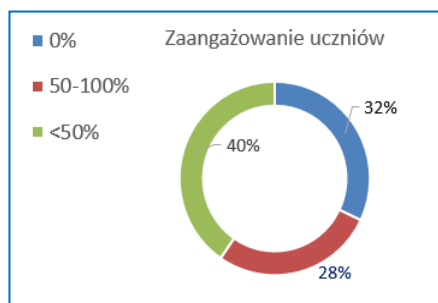
Omawiane algorytmy znalazły odzwierciedlenie w zadaniach do samodzielnego rozwiązania. Nie zabrakło też aktywności związanych z testowaniem rozwiązań w praktyce. Umiejętność analizy kodu, znajdowania błędów i ich poprawiania gra niebagatelną rolę. Zgodność z treścią zadania oraz przetestowanie rozwiązania dla warunków brzegowych jest kluczowe, szczególnie w odniesieniu do zadań konkursowych.



Rysunek 11 Spostrzeżenia ucznia z debugowania kodu

5. Podsumowanie

Zainteresowanie kursem Pierwsze kroki w Pythonie miało odzwierciedlenie w liczbie uczestników – 483. Niestety, tylko 44 uczniów mogło obejrzeć gratulacje, odsłaniane tym osobom, które uzyskały co najmniej 95% możliwych do zdobycia punktów.



Rysunek 12 Oceny uczniów – Pierwsze kroki

W kolejnych szkoleniach brało udział odpowiednio w Grafice żółwia w Pythonie – 92 uczniów, w Obliczeniach i algorytmice – 108, w ostatnim W głąb algorytmiki – 121. Była to okazja do pogłębienia umiejętności informatycznych, interpersonalnych i uczenia się w nowej formie – szkoleń online. W każdym z powyższych szkoleń brali udział jako obserwatorzy zainteresowani nauczyciele informatyki, głównie ci, którzy pracują ze zdolnymi uczniami na zajęciach pozalekcyjnych.

Literatura

1. Jochemczyk W., Olędzka K., Pythonowe wyzwania dla początkujących, Informatyka w Edukacji, Toruń 2017.
2. Programowanie w szkole, <http://programowanie.oeiizk.edu.pl>, ostatni dostęp 31.05.2019 roku.
3. Python w szkole, <http://python.oeiizk.edu.pl>, ostatni dostęp 31.05.2019 roku.