

gi lub produktu i czy jego oczekiwaną są spełnione. Design thinking to sposób na wypracowanie innowacji małymi krokami. Dzięki tej metodzie powstały nowe produkty, opakowania, usługi, projekty przestrzeni publicznej, koncepty marketingowe oraz przełomowe technologie. Cieszy fakt, że Design thinking staje się coraz bardziej popularny w edukacji.

Przytisył

1. Bazyńska W., Lis A., Szwedak M., Kowal K., Szlachetka P. *Przewodnik przewodnik metod ek. Creative Learning 2018*.

2. Masłowski J. *Przewodnik ek. Ek. Creative Learning 2018*.

3. *Ek. Creative Learning 2018*.

4. *Ek. Creative Learning 2018*.

5. *Ek. Creative Learning 2018*.

6. *Ek. Creative Learning 2018*.

7. *Ek. Creative Learning 2018*.

Marek Wróblewski

Lekcja programowania: Algorytm Euklidesa w Python



Podstawa programowa przedmiotu informatyka II etap edukacyjny: klasy IV-VIII. Treści nauczania – wymagania szczegółowe – klasy VII i VIII

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

1. **Stosuje różne sposoby przedstawiania algorytmów, w tym w języku naturalnym, w postaci schematów blokowych, listy kroków;**
2. **stosuje przy rozwiązywaniu problemów podstawowe algorytmy:**
 - 1) **na liczbach naturalnych: bada podzielność liczb, wyodrębnia cyfry danej liczby, przedstawia działanie algorytmu Euklidesa w obu wersjach iteracyjnych (z odejmowaniem i z resztą z dzielenia),**
 - 2) **wyszukiwania i porządkowania: wyszukuje element w zbiorze uporządkowanym i nieuporządkowanym oraz porządkuje elementy w zbiorze metodą przez proste wybieranie i zliczanie;**

7. Design thinking. *Ek. Creative Learning 2018*. <http://www.creativelearning.pl/>

Manuskr. 2018

Podstawa programowa z informatyki podkreśla znaczenie algorytmicznego myślenia. Algorytm Euklidesa służy do poszukiwania największego wspólnego dzielnika dwóch liczb naturalnych (NWD), jako jedyny jest wymieniony z nazwy w podstawie programowej klas VII i VIII. Jego realizacja w obu wersjach iteracyjnych, metodą przez odejmowanie i metodą przez dzielenie, jest obowiązkowa. Z zapisów podstawy programowej wynika, że uczeń powinien stosować różne sposoby przedstawiania algorytmów w tym w postaci schematów blokowych – to zagadnienie zostało przeze mnie omówione w artykułach zamieszczonych w poprzednich wydaniach „Informatora Oświatowego” (nr 2/2021, s. 48-49; nr 3/2021, s. 62).

Docelową formą prezentacji algorytmu jest język wysokiego poziomu. Wybrałem Pythona ze względu na prostotę i rosnącą popularność zarówno na rynku, jak i w edukacji. W Python można stworzyć algorytmy na maturze. Do kodu dodałem zmienną licznik, która nie ma bezpośredniego związku z algorytmem Euklidesa, ale jej zadaniem będzie policzenie ilości przebiegu pętli wskazującej NWD. Sprawdźmy w ten sposób wydajność obu wersji algorytmu.

Kod Python algorytmu Euklidesa – wyznaczenie NWD metodą przez odejmowanie.

```

1. liczba1=int(input('podaj pierwszą liczbę: '))
2. liczba2=int(input('podaj drugą liczbę: '))
3. licznik=0
# początek pętli
4. while liczba1!=liczba2:
5.     licznik+=1
6.     if liczba1>liczba2:
7.         liczba1=liczba1-liczba2
8.     else:
9.         liczba2=liczba2-liczba1
# koniec pętli
10.print ("pętla wykonana {} razy".format(licznik))
11.print("NWD to:", liczba1 )

```

Wyjaśnienie poszczególnych linii kodu:

1. Wczytuję z klawiatury zmienną liczba1. Funkcja input wyświetla komunikat i czeka na wprowadzenie wartości. Funkcja int zamienia wprowadzoną wartość (domyślnie string) na liczbę typu integer.
2. Jak w pk.1 ze zmienną liczba2.
3. Tworzę zmienną licznik z wartością początkową 0 po to aby sprawdzić wydajność algorytmu. Zmienna będzie zliczała przebiegi pętli.
4. Początek pętli dopóki liczba1 jest różna od liczba2 (!= operator nie są sobie równe).
5. Dodaję 1 do wartości zmiennej licznik, jest to skrót zapisu licznik = licznik+1.
6. Początek instrukcji warunkowej jeżeli liczba1 jest większa od liczba2.
7. Jeżeli warunek z pk. 6 jest spełniony podstaw do liczba1 wartość liczba1-liczba2.
8. W przeciwnym wypadku (do warunku z pk6).
9. Jeżeli warunek z pk. 6 nie jest spełniony, podstaw do liczba2 wartość liczba2-liczba1. Koniec pętli.
- 10.Wyprowadź na ekran komunikat, zastosowałem funkcję format, która wstawi wartość zmiennej licznik do nawiasu {}.
- 11.Wyprowadź wartość NWD. Można byłoby użyć funkcji format. Kod tej linijki wyglądałby tedy tak: print("NWD to : {}".format(liczba1)).

Kod Python algorytmu Euklidesa – wyznaczenie NWD metodą przez dzielenie.

```

1. liczba1=int(input(,podaj pierwszą liczbę: ,))
2. liczba2=int(input(,podaj drugą liczbę: ,))
3. licznik=0
4. if liczba2==0:
5.     print(„NWD to:”,liczba1)
6.     else:
# początek pętli
7.     while liczba2>0:
8.         licznik+=1
9.         reszta=liczba1%liczba2
10.     liczba1=liczba2
11.     liczba2=reszta
# koniec pętli

```

```

12. print(„pętla wykonana {} razy”.format(licznik))
13. print(„największy wspólny dzielnik to:”,liczba1)

```

Wyjaśnienie poszczególnych linii kodu:

1. Wczytuję z klawiatury zmienną liczba1. Funkcja input wyświetla komunikat i czeka na wprowadzenie wartości. Funkcja int zamienia wprowadzoną wartość (domyślnie string) na liczbę typu integer.
2. Jak w pk.1 ze zmienną liczba2.
3. Tworzę zmienną licznik z wartością początkową 0 po to aby sprawdzić wydajność algorytmu. Zmienna będzie zliczała przebiegi pętli.
4. Jeżeli liczba 2 jest równa 0 to przejdź do pk.5.
5. Wyprowadź komunikat NWD to : wartość liczba1.
6. W przeciwnym wypadku przejdź do pk.7.
7. Początek pętli dopóki liczba2 jest większa od 0.
8. Dodaję 1 do wartości zmiennej licznik, jest to skrót zapisu licznik = licznik+1.
9. Wstawiam do zmiennej reszta, resztę z dzielenia liczba1 przez liczba 2 (% operator modulo)
- 10.Wstawiam wartość zmiennej liczba2 do liczba1
- 11.Wstawiam wartość zmiennej reszta do liczba2
Koniec pętli.
- 12.Wyprowadź na ekran komunikat, zastosowałem funkcję format, która wstawi wartość zmiennej licznik do nawiasu {}.
- 13.Wyprowadź wartość NWD. Można byłoby użyć funkcji format. Kod tej linijki wyglądałby tedy tak: print("NWD to : {}".format(liczba1)).

Algorytm metodą przez odejmowanie wykona dla pary liczb 123456789 i 2 pętlę 61728395 razy, a algorytm metodą przez dzielenie dla tej samej pary liczb, wyświetli wartość 2 licznika pętli. Czas działania algorytmu bazującego na metodzie przez odejmowanie przy dużych liczbach jest znaczący, wykonanie takiej liczby operacji musi trwać.

Zapraszam na warsztaty z cyklu *Jak uczyć programowania?* organizowane przez nasz Ośrodek. Podczas szkoleń wyjaśniam wszystkie pojawiające się niejasności i zawiłości, zarówno pod względem merytorycznym, jak i technicznym.

Marek Wróblewski

Nauczyciel konsultant ds. wspomagania nauczycieli w zakresie stosowania technologii informacyjno-komunikacyjnej ODN w Słupsku. Autor wielu form doskonalenia dla nauczycieli informatyki i przedmiotów nieinformatycznych. Moderator nauczycielskich sieci współpracy i samokształcenia.