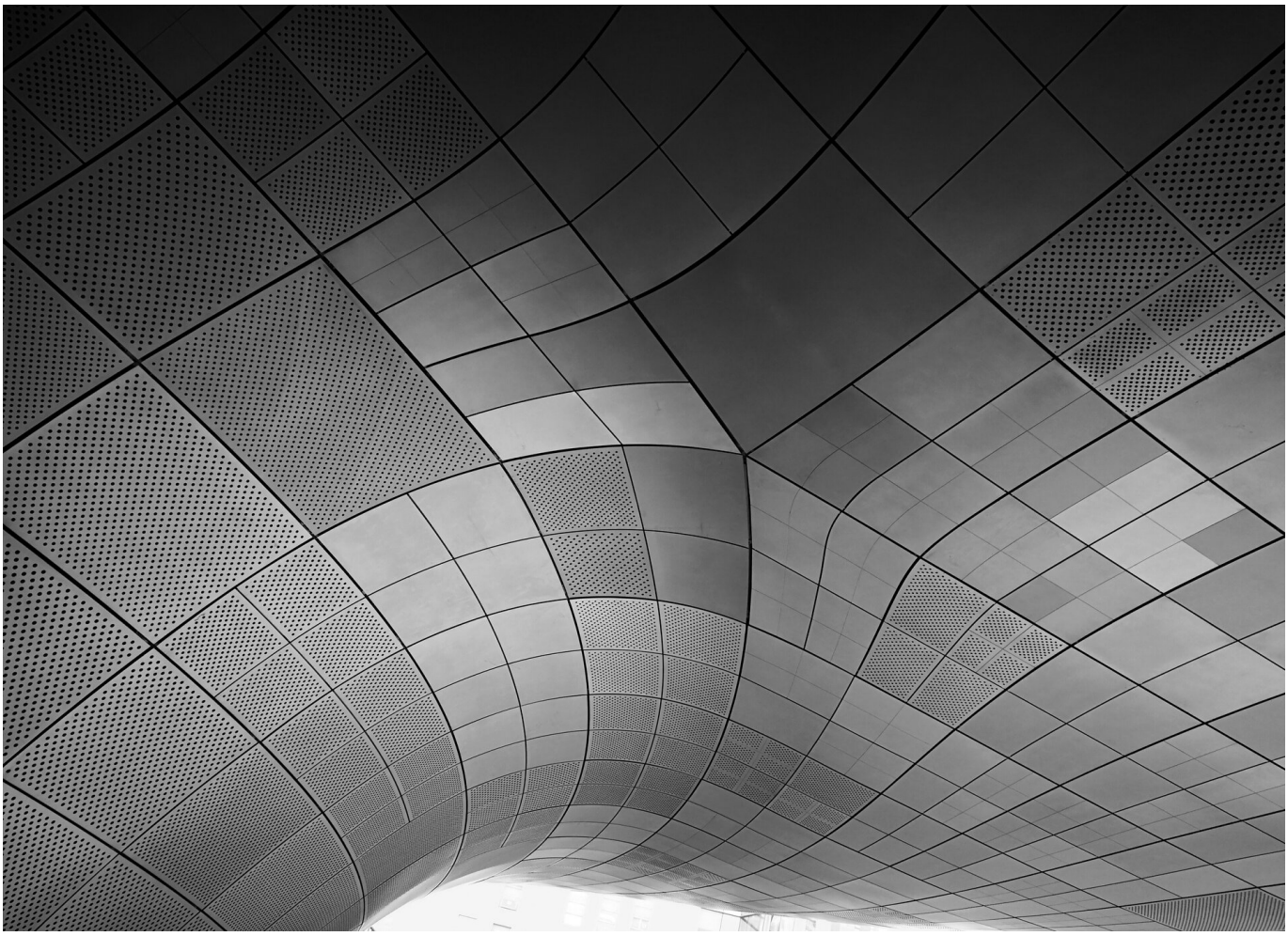
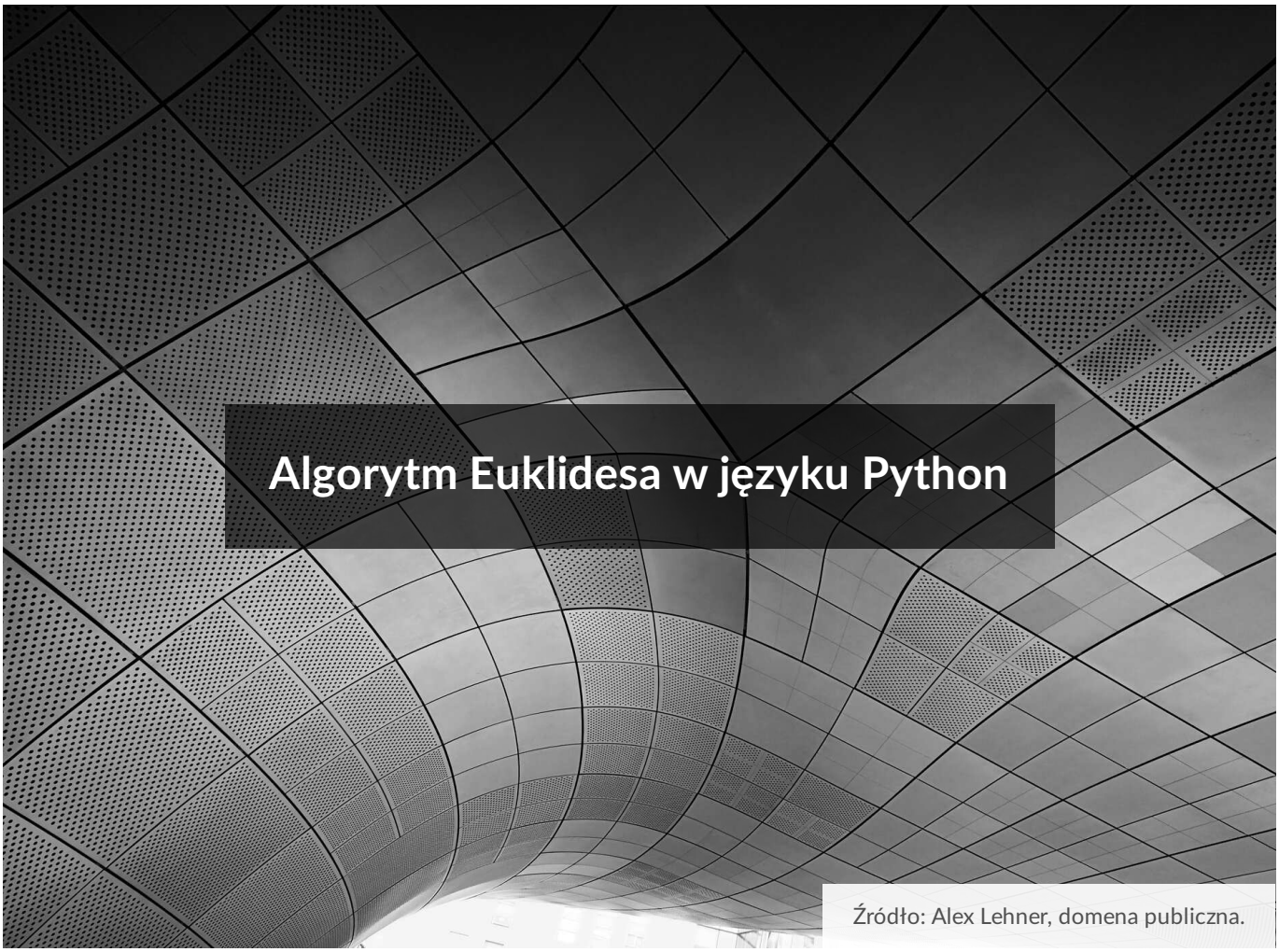


Algorytm Euklidesa w języku Python



Źródło: Alex Lehner, domena publiczna.

> W tym e-materiale powtarzamy wiadomości ze szkoły podstawowej.

Znamy już Algorytm Euklidesa w dwóch wariantach. Stosujemy go jako metodę wyznaczania największego wspólnego dzielnika oraz najmniejszej wspólnej wielokrotności dwóch liczb.

Największy wspólny dzielnik wykorzystujemy np. podczas skracania ułamków.

W tym e-materiale zaimplementujemy go w języku Python.

Ciekawi cię, jak wyglądają implementacje w innych językach programowania? Możesz się z nimi zapoznać w dwóch pozostałych e-materiałach z tej serii:

- Algorytm Euklidesa w języku C++,
- Algorytm Euklidesa w języku Java.

Więcej zadań? Sięgnij do Algorytm Euklidesa – zadania maturalne.

> **Twoje cele**
>
> - Zaimplementujesz algorytm Euklidesa, wykorzystujący metodę odejmowania.
> - Napiszesz program wykorzystujący algorytm Euklidesa w wersji z dzieleniem modulo.
> - Porównasz efektywność obu wersji algorytmu Euklidesa.

Przeczytaj

Przeanalizujmy działanie algorytmu Euklidesa w wersji z odejmowaniem. Oto jego realizacja w języku Python:

```
def NWD_odejmowanie(x, y):
    while x != y:
        if x > y:
            x -= y
        else:
            y -= x

    return x
```

Wyznaczmy przykładowe NWD, używając powyżej zdefiniowanej funkcji:

```
NWD_odejmowanie(720424,4)
4
NWD_odejmowanie(9203523424,23)
1
```

Spróbujmy wykorzystać tę metodę w przypadku dużych liczb. W języku Python istnieje ograniczenie maksymalnej wartości typu `int`. Możemy to sprawdzić, wywołując odpowiednie właściwości modułów `sys` oraz `platform`:

```
import sys
print(sys.maxsize)
9223372036854775807
import platform
print(platform.uname())
uname_result(system='Linux', node='zorin-desktop', release='5.0.0
version='#33~18.04.1-Ubuntu SMP Tue Oct 1 10:20:39 UTC 2019',
machine='x86_64', processor='x86_64')
```

Zatem największą liczbą całkowitą, którą możemy podać w 64-bitowym systemie Linux, jest:

9 223 372 036 854 775 807

Oto wersja programu, która liczy, ile operacji odejmowania należy wykonać podczas wyznaczania NWD:

```
def NWD_odejmowanie_zliczanie(x, y):
    odejmowania = 0
    while x != y:
        if x > y:
            x -= y
        else:
            y -= x
        odejmowania += 1

    print('NWD',x,'; liczba odejmowań', odejmowania)
```

Wyznaczmy NWD dla liczb o wiele mniejszych niż maksymalna wartość `int` zapisana w 64-bitowym systemie Linux:

```
NWD_odejmowanie_zliczanie(920354,22)
NWD 2 ; liczba odejmowań 41839
#
NWD_odejmowanie_zliczanie(9203523424,22)
NWD 2 ; liczba odejmowań 418341979

NWD_odejmowanie_zliczanie(927566801,22)
NWD 1 ; liczba odejmowań 42162136
```

Niewykluczone, że obliczenia zajmą ponad minutę. W przypadku bardzo dużych liczb program uruchomiony na szybkim komputerze mógłby pracować nawet kilkadziesiąt minut (ze względu na dużą liczbę operacji odejmowania). Przedstawiamy obraz z Monitora systemu – pokazuje on, że nawet proste obliczenia potrafią obciążać procesor:

Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Polecenie 1

Możemy zmierzyć czas niezbędny do obliczenia `NWD`. Spróbujmy napisać funkcję `NWD_odejmowanie_zliczanie_czas(x, y)`, która oprócz liczby odejmowań poda czas niezbędny do ich wykonania.

Specyfikacja problemu:

Dane:

- `x, y` – liczby całkowite

Wynik:

- `NWD` – liczba całkowita, największy wspólny dzielnik liczb `x` i `y`
- `odejmowania` – liczba całkowita, liczba wykonanych operacji
- `t` – czas trwania programu; liczba rzeczywista

```
1
```

```
1
```

Na szczęście proces wyznaczania NWD da się zoptymalizować. Przeanalizujmy zmodyfikowaną wersję algorytmu, tzw. **obliczanie NWD metodą dzielenia z resztą**.

Oto implementacja algorytmu w języku Python:

```
def NWD_modulo(x, y):
    while y > 0:
        modulo = x%y
        x = y
        y = modulo
    return x
```

Wersja programu, która zlicza wykonane operacje, może wyglądać następująco:

```
def NWD_modulo_zliczanie(x, y):
    dzielenia = 0
    while y > 0:
        modulo = x % y
        x = y
        y = modulo
        dzielenia += 1

    print('NWD',x,'; liczba dzieleń', dzielenia)
```

Wyznaczmy NWD dla dużych argumentów funkcji. Okaże się, że liczba wykonywanych operacji jest bardzo mała:

```
NWD_modulo_zliczanie(927566801,22)
NWD 1 ; liczba dzieleń 3
```

Polecenie 2

Możemy obliczyć czas niezbędny do wyznaczenia `NWD`. Zdefiniujmy funkcję `NWD_modulo_zliczanie_czas(x, y)`, która oprócz liczby operacji zmierzy także czas ich wykonania.

Specyfikacja problemu:

Dane:

- `x, y` – liczby całkowite

Wynik:

- `NWD` – liczba całkowita, największy wspólny dzielnik liczb `x` i `y`
- `dzielenia` – liczba całkowita, liczba wykonanych operacji
- `t` – czas trwania programu; liczba rzeczywista

```
1
```

```
1
```

Dla zainteresowanych

Możemy obliczyć NWD dla więcej niż dwóch liczb.

Obliczenie NWD dla trzech liczb odbywa się w dwóch etapach. Najpierw wyznaczamy NWD dla liczb pierwszej i drugiej oraz NWD liczb drugiej i trzeciej. Następnie obliczamy NWD dla otrzymanych wyników. Przedstawmy tę zasadę za pomocą pseudokodu:

```
obliczenie NWD(a, b, c):

  do zmiennej tymczasowej x przypisz NWD(a,b)
  do zmiennej tymczasowej y przypisz NWD(b,c)

  wynikiem będzie NWD(x,y)

# inny sposób rozwiązania problemu:
NWD(a,b,c)=NWD(NWD(a,b),c)
```

Problem 1

Algorytm Euklidesa wykorzystujący odejmowanie opiera swoje działanie o spostrzeżenie, że największy wspólny dzielnik jest równy największemu wspólnemu dzielnikowi mniejszej liczby i różnicy liczb większej oraz mniejszej.

Napisz program, który obliczy największy wspólny dzielnik dwóch liczb całkowitych dodatnich. Zaimplementuj algorytm Euklidesa, wykorzystując:

- odejmowanie,
- resztę z dzielenia.

Specyfikacja problemu:

Dane:

- a, b – liczby całkowite

Wynik:

- `NWD` – liczba całkowita, największy wspólny dzielnik liczby a i b

```
# Tutaj dodaj własny kod. Użyj funkcji
 # print()
```

```
1
```

Polecenie 3

Porównaj swoje rozwiązanie z przedstawionym w filmie.

Film dostępny pod adresem /preview/resource/RbNtfLMrgEPh1

Film nawiązujący do treści materiału: Algorytm Euklidesa zastosowanie różnych pętli.

Słownik

modulo

operacja wyznaczania reszty z dzielenia dwóch liczb; działanie takie oznaczane jest w Pythonie operatorem %

optymalizacja

upraszczanie procesu, poszukiwanie najlepszego rozwiązania z punktu widzenia pewnego kryterium, np. liczby wykonywanych operacji przez procesor czy czasu niezbędnego do przeprowadzonych działań lub ilości zajmowanej pamięci

Audiobook

Polecenie 1

Zapoznaj się z audiobookiem opowiadającym o dziele Euklidesa, a następnie dowiedz się, na czym polega szyfr afiniczny.

Audiobook można wysłuchać pod adresem: https://zpe.gov.pl/b/PsWD7mNY0

Elementy i algorytm Euklidesa

Autorstwo Elementów powszechnie przypisuje się Euklidesowi, choć coraz częściej pojawiają się głosy, że dzieło jest kompilacją znanych już wyników badań. Niezależnie od tego, jakiej części autorem jest rzeczywiście Euklides, to dzieło podpisane jego imieniem przetrwało.

Elementy były kanonem nauczania matematyki aż do końca XIX wieku – do tego czasu doczekały się niezliczonej liczby wydań oraz tłumaczeń. Szacuje się, że popularnością w kwestii wydawania ustępują jedynie Biblii.

Pierwszego polskiego tłumaczenia ośmiu ksiąg Elementów pod tytułem Początków geometrii ksiąg ośmioro, to jest sześć pierwszych, jedenasta i dwunasta dokonał na początku XIX wieku Józef Czech, ukazało się ono w roku 1807 w Krzemieńcu.

Dzieło Euklidesa składa się z trzynastu ksiąg. Pierwsze cztery księgi poświęcone są geometrii płaskiej – planimetrii. W kolejnych znajdują się zagadnienia arytmetyki oraz stereometrii. Algorytm Euklidesa pojawia się w księdze VII. Wyróżniamy podstawowy oraz rozszerzony algorytm Euklidesa.

Podstawowa wersja algorytmu Euklidesa pozwala wyznaczyć największy wspólny dzielnik dla dwóch liczb naturalnych. Rozszerzony algorytm Euklidesa wyznacza współczynniki kombinacji dwóch liczb, która w sumie daje ich największy wspólny dzielnik. Tego typu algorytm wykorzystuje się w kryptografii zarówno do szyfrowania, jak również łamania szyfrów. Przykładowo został on wykorzystany w szyfrze afinicznym w celu jak najszybszego znalezienia elementu odwrotnego.

Znasz zastosowanie podstawowej wersji algorytmu. Algorytm rozszerzony pozwala na rozwiązanie zagadki przelewania wody. W wersji ogólnej ma ona następującą postać: dysponujesz dwoma czerpakami o pojemności m litrów oraz n litrów, pustym pojemnikiem o nieograniczonej pojemności oraz nieograniczoną ilością wody. W jaki sposób napełnisz pojemnik k litrami wody, jeśli możesz wodę wlewać do pojemnika i z niego wylewać tylko pełnymi czerpakami?

Nieoczekiwanie wariacja opisanej zagadki matematycznej zrobiła karierę w Hollywood. W filmie z 1995 roku Szklana pułapka 3 bohaterowie muszą ją rozwiązać, jeśli chcą rozbroić bombę. Szczęśliwie bohaterowie odrobili lekcje i udało im się podać poprawną odpowiedź.

Polecenie 2

Znana jest następująca łamigłówka, do rozwiązania której można wykorzystać algorytm Euklidesa. Mamy 8-litrowe naczynie pełne wody. Za pomocą dwóch pustych naczyń o pojemnościach 3 oraz 5 litrów chcemy odmierzyć 4 litry wody. Spróbuj znaleźć rozwiązanie tej łamigłówki

Polecenie 3

Zapisz notatkę, w której podsumujesz najważniejsze informacje przedstawione w audiobooku.

Sprawdź się

Pokaż ćwiczenia: 

Ćwiczenie 1

Napisz program, który dla trzech liczb naturalnych: `a`, `b`, `c` wypisuje dwie z nich, które mają największy wspólny dzielnik.

Specyfikacja problemu:

Dane:

- `a`, `b`, `c` – liczby naturalne dodatnie

Wynik:

Na standardowym wyjściu program wypisuje dwie spośród trzech liczb, mające maksymalny NWD.

Jeśli wszystkie pary mają taki sam NWD, program wypisuje komunikat:

```
Pary mają taki sam NWD
```

Przykładowe wyjście dla `a = 7`, `b = 100`, `c = 150`:

```
100 i 150
```

Twoje zadania

1. Program wypisuje dwie spośród trzech liczb, mające maksymalny NWD.

Ćwiczenie 2

Firma *Warzywa Inc* produkuje włoszczyznę. Proces produkcji polega na umieszczeniu na plastikowej tacce marchewek, pietruszek, porów i selerów, owinięcie ich folią spożywczą i zapakowaniu w kartony zbiorcze. Firmie udało się zakontraktować u rolników a sztuk marchewek, b sztuk pietruszki, c sztuk porów i d sztuk selerów. Ile jednakowych paczek włoszczyzny może przygotować do sprzedaży filma *Warzywa Inc*, tak aby było ich jak najwięcej?

Prezes firmy, Jan Bajtek nie lubi marnowania żywności – zawsze zamawia warzywa w taki sposób, aby produkcja odbywała się bez strat i wykorzsytane były **wszystkie** warzywa.

Swoje rozwiązanie przetestuj dla `a = 210, b = 150, c = 90, d = 60`.

Specyfikacja problemu:

Dane:

- a – liczba marchewek; liczba naturalna dodatnia
- b – liczba pietruszek; liczba naturalna dodatnia
- c – liczba porów; liczba naturalna dodatnia
- d – liczba selerów; liczba naturalna dodatnia

Wynik:

Na standardowym wyjściu program wyświetla liczbę paczek włoszczyzny.

Twoje zadania

1. Program wypisuje liczbę paczek włoszczyzny, które uda się wyprodukować z zadanej liczby warzyw.

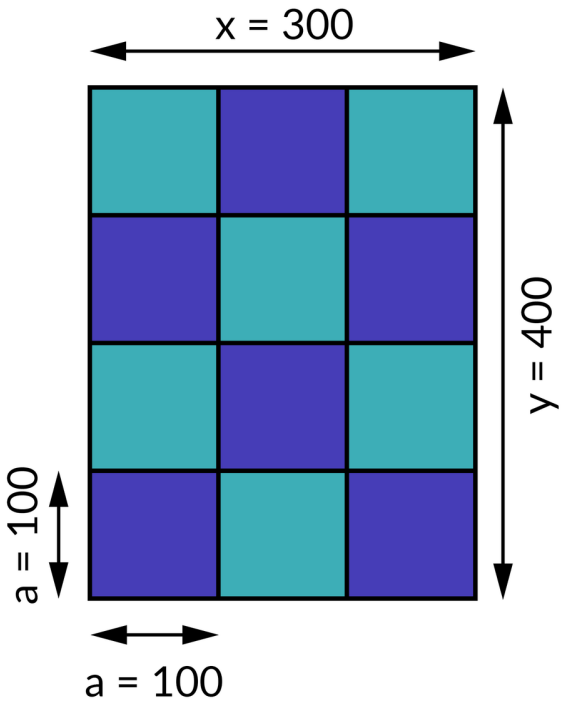
Ćwiczenie 3

Bajtazar chce wyremontować posadzkę w swojej kuchni o wymiarach x na y. W tym celu chce zakupić kwadratowe kafelki o boku długości a (jednostki są ustandaryzowane). Bajtazar chce kupić możliwie jak najmniej kafelek – zakładamy, że w sklepie będzie mógł dobrać dowolny rozmiar, który idealnie wpasuje się w podłogę kuchni (nie będzie konieczności docinania ani kupowania na zapas). Bajtazar nie lubi monotonni – kafelki będą w dwóch kolorach: morskim oraz fioletowym, ułożone naprzemiennie.

Pomóż Bajtazarowi przy remoncie: policz jaki wymiar a powinny mieć kafelki oraz ile powinien ich kupić w każdym z kolorów. Bajtazar chciałby rozpocząć układanie kafelek od lewego górnego rogu kuchni. W tym miejscu wymarzył sobie morski kafelek.

Przykład:

Podłoga o wymiarach `x=300` i `y=400`. Długość boku kafelka to `a=100`. Musi zakupić 6 kafelków morskich i 6 fioletowych.



Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Swoje rozwiązanie przetestuj dla x wynoszącego 640, y równego 400.

Specyfikacja problemu:

Dane:

- x, y – szerokość i długość podłogi; liczby naturalne dodatnie

Wynik:

Na standardowym wyjściu program wyświetla w kolejnych wierszach: długość boku kafelka (a), liczbę kafelków morskich oraz liczbę kafelków fioletowych.

Przykładowe rozwiązania:

Podłoga o wymiarach `x=300` i `y=400`. Długość boku kafelka to `a=100`. Musi zakupić 6 kafelków morskich i 6 kafelków fioletowych.

Podłoga o wymiarach `x=100` i `y=100`. Długość boku kafelka to `a=100`. Musi zakupić 1 kafelek morski.

Twoje zadania

1. Program liczy długość pojedynczej kafelki oraz wskazuje liczbę kafelek fioletowych i morskich.

Dla nauczyciela

Autor: Adam Jurkiewicz
Przedmiot: Informatyka

Temat: Algorytm Euklidesa w języku Python

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres podstawowy i rozszerzony

Podstawa programowa:

Cele kształcenia – wymagania ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Treści nauczania – wymagania szczegółowe

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

3) sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

I + II. Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

1) zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy poznane na wcześniejszych etapach oraz algorytmy:

a) algorytm Euklidesa w wersji iteracyjnej i rekurencyjnej wraz z zastosowaniami,

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Zaimplementujesz algorytm Euklidesa, wykorzystujący metodę odejmowania.
- Napiszesz program wykorzystujący algorytm Euklidesa w wersji z dzieleniem modulo.
- Porównasz efektywność obu wersji algorytmu Euklidesa.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimedium i ćwiczeń interaktywnych;
- ćwiczenia praktyczne.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda;
- oprogramowanie dla języka Python 3 (lub nowszej wersji), w tym PyCharm lub IDLE.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Algorytm Euklidesa w języku Python". Uczniowie mają zapoznać się z treściami w sekcji „Przeczytaj".
   Chętny lub wybrany uczeń przygotowuje rozwiązanie polecenia nr 1 z sekcji „Przeczytaj". Będzie pełnił rolę eksperta podczas zajęć.

Faza wstępna:

1. Nauczyciel wyświetla temat i cele zajęć. Prosi uczniów, by na podstawie wiadomości zdobytych przed lekcją zaproponowali kryteria sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Nauczyciel prosi wybranego ucznia lub uczniów o przedstawienie sytuacji problemowej związanej z tematem lekcji.

Faza realizacyjna:

1. **Praca z tekstem.** Nauczyciel wyświetla zawartość sekcji „Przeczytaj". Uczniowie wykonują polecenie nr 1: „Możemy zmierzyć czas niezbędny do obliczenia `NWD`. Spróbujmy napisać funkcję `NWD_odejmowanie_zliczanie_czas(x, y)`, która oprócz liczby odejmowań poda czas niezbędny do ich wykonania" oraz polecenie nr 2: „Możemy obliczyć czas niezbędny do wyznaczenia NWD. Zdefiniujmy funkcję `NWD_modulo_zliczanie_czas(x, y)`, która oprócz liczby operacji zmierzy także czas ich wykonania."
2. **Praca z multimedium.** Nauczyciel wyświetla zawartość sekcji „Audiobook". Uczniowie wspólnie zapoznają się z poleceniem 1 i analizują prezentację. W kolejnym kroku dzielą się swoimi spostrzeżeniami na forum klasy.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują indywidualnie ćwiczenie nr 1, a następnie porównują swoje odpowiedzi z kolegą lub koleżanką.
4. W kolejnym etapie uczniowie dobierają się w pary i wykonują ćwiczenia nr 2–3. Następnie konsultują swoje rozwiązania z inną parą uczniów i ustalają jedną wersję odpowiedzi.

Faza podsumowująca:

1. Na koniec zajęć z programowania w Pythonie nauczyciel prosi uczniów o rozwinięcie zdania: „Na dzisiejszych zajęciach nauczyłam/łem się jak…".

Praca domowa:

1. Uczniowie wykonują polecenie nr 2 z sekcji „Audiobook".

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla języka Python 3 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania PyCharm lub IDLE.

Wskazówki metodyczne:

- Treści w sekcji „Przeczytaj" można wykorzystać jako podsumowanie i utrwalenie wiedzy uczniów.