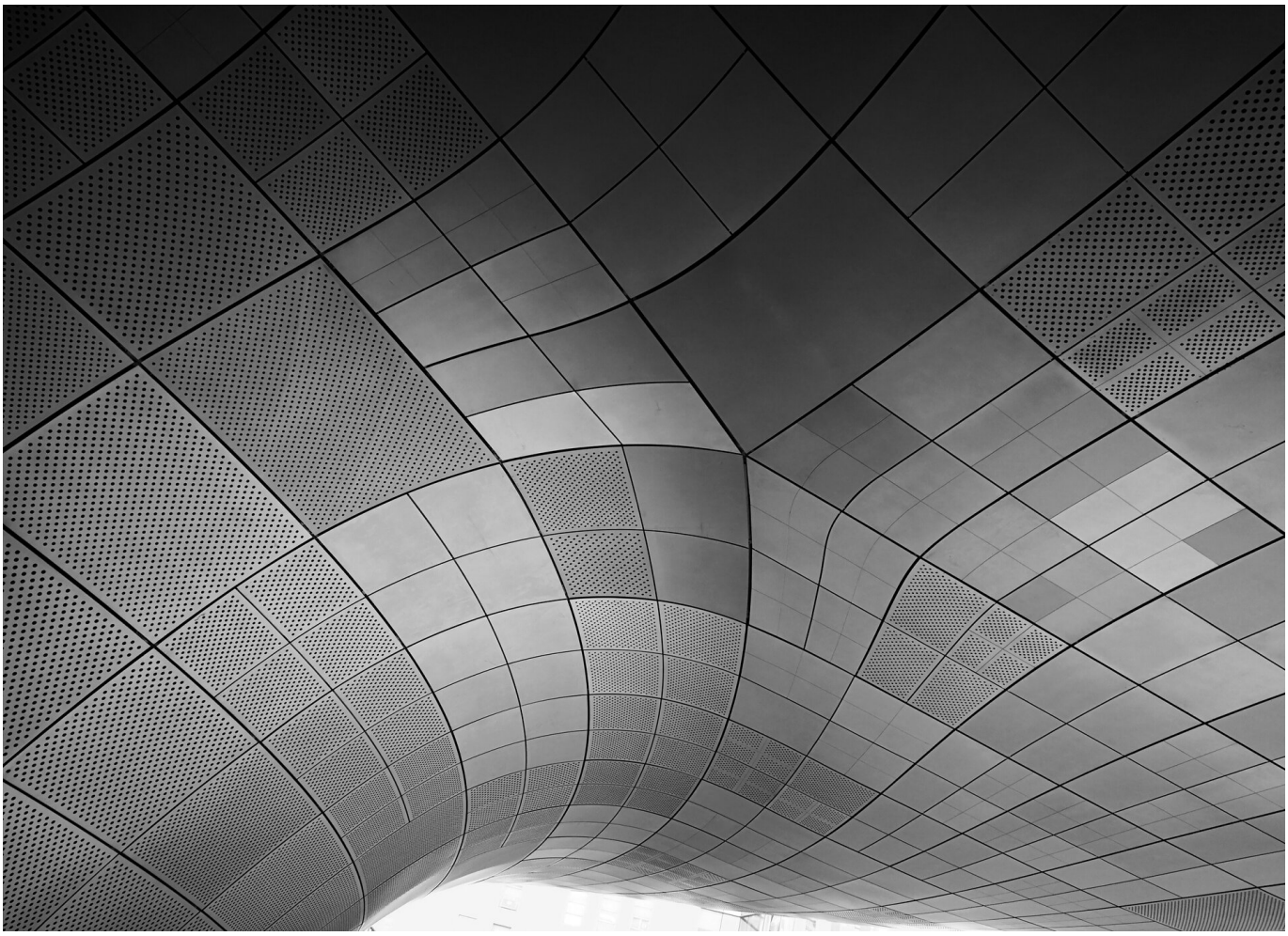


Algorytm Euklidesa w języku Java

- Wprowadzenie
- Przeczytaj
- Gra edukacyjna
- Sprawdź się
- Dla nauczyciela

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 podczas skracania ułamków.

W tym e-materiale zaimplementujemy go w języku Java.

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 Python.

Więcej zadań? Sięgnij do Algorytm Euklidesa – zadania maturalne.

> **Twoje cele**
>
> - Scharakteryzujesz dwie wersje algorytmu Euklidesa.
> - Przeanalizujesz implementacje algorytmu Euklidesa w języku Java.
> - Rozwiążesz zadania związane z algorytmem Euklidesa.

Przeczytaj

Problem 1

Napisz program w języku Java znajdujący NWD dwóch liczb naturalnych *a* i *b*. Wykorzystaj jeden z wariantów algorytmu Euklidesa. Przetestuj jego działanie dla liczb 368 oraz 4.

Specyfikacja problemu:

Dane:

- a, b – dodatnie liczby naturalne

Wynik:

- NWD podanych liczb.

```
public class Main {
    public static void main(String [] args) {
        int a = 110;
        int b = 5;    // dla testów zmień liczby a i b

        // tutaj dopisz kod

    }
}
```

```
1
```

Polecenie 1

Przeanalizuj prezentację. Porównaj swoje rozwiązanie z kodem przedstawionym w prezentacji.

1

Zacznijmy od algorytmu Euklidesa, wykorzystującego operacje odejmowania.

2

Użytkownik wprowadza liczby a oraz b:

```
int a = 0;
int b = 0;
```

3

Zależy nam na pobraniu danych od użytkownika, tworzymy więc instancje klasy `Scanner`. Wykorzystując stworzony obiekt, zapisujemy do zmiennych a i b liczby, dla których wyznaczymy NWD.

```
int a = 0;
int b = 0;
```

```
3
4 Scanner sc = new
  Scanner(System.in);
5 a = sc.nextInt();
6 b = sc.nextInt();
```

4

Teraz przygotujemy pętlę, która zakończy działanie w momencie, gdy a oraz b będą mieć tę samą wartość:

```
1  int a = 0;
2  int b = 0;
3
4  Scanner sc = new
   Scanner(System.in);
5  a = sc.nextInt();
6  b = sc.nextInt();
7
8  while (a != b) {
9      // Odejmowanie
10 }
```

5

Musimy ustalić, która liczba jest większa, aby później odjąć od niej mniejszą. Użyjemy instrukcji warunkowej:

```
1  int a = 0;
2  int b = 0;
3
4  Scanner sc = new
   Scanner(System.in);
5  a = sc.nextInt();
6  b = sc.nextInt();
7
8  while (a != b) {
9      if (a > b) {
```

```
10          a = a - b;
11      } else {
12          b = b - a;
13      }
14  }
```

6

Na koniec wyświetlimy NWD. Oto cały program:

```
1  import java.util.Scanner;
2
3  public class Main {
4      public static void main(String [] args) {
5          int a = 0;
6          int b = 0;
7
8          Scanner sc = new Scanner(System.in);
9          a = sc.nextInt();
10         b = sc.nextInt();
11
12         while (a != b) {
13             if (a > b) {
14                 a = a - b;
15             } else {
16                 b = b - a;
17             }
18         }
19
20         System.out.println("NWD: " + a);
21
22     }
23 }
```

7

Przejdźmy do drugiego wariantu algorytmu. Ponownie zaczynamy od stworzenia zmiennych oraz instancji klasy `Scanner`.

```
Scanner sc = new Scanner(System.in);
int a = sc.nextInt();
int b = sc.nextInt();
```

8

I znowu posłużymy się pętlą; będzie ona działać dopóty, dopóki zmienna b nie osiągnie wartości 0:

```
Scanner sc = new Scanner(System.in);
int a = sc.nextInt();
int b = sc.nextInt();

while (b != 0) {

}
```

9

Kolejna czynność polega na obliczaniu reszty z dzielenia zmiennej a przez zmienną b. Najpierw użyjemy zmiennej pomocniczej c. Następnie zmiennej a przypiszemy wartość zmiennej b, a zmienna b przyjmie wartość zmiennej pomocniczej c.

```
Scanner sc = new Scanner(System.in);
int a = sc.nextInt();
int b = sc.nextInt();

while (b != 0) {
    int c = a % b;
```

```
7     a = b;
8     b = c;
9 }
```

10

Po zakończeniu działania pętli zmienna a będzie przechowywać wartość NWD. Wypiszmy ją na ekranie:

```
1  Scanner sc = new
   Scanner(System.in);
2  int a = sc.nextInt();
3  int b = sc.nextInt();
4
5  while (b != 0) {
6      int c = a % b;
7      a = b;
8      b = c;
9  }
10
11 System.out.println("NWD: "
   + a);
```

11

Pora przedstawić pełny kod programu:

```
1 import java.util.Scanner;
2 public class Main {
3     public static void
  main(String [] args) {
4         Scanner sc = new
  Scanner(System.in);
5         int a =
  sc.nextInt();
6         int b =
  sc.nextInt();
7
8         while (b != 0) {
```

```
9                int c = a % b;
10               a = b;
11               b = c;
12           }
13
14
System.out.println("NWD: "
+ a);
15
16       }
17 }
```

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

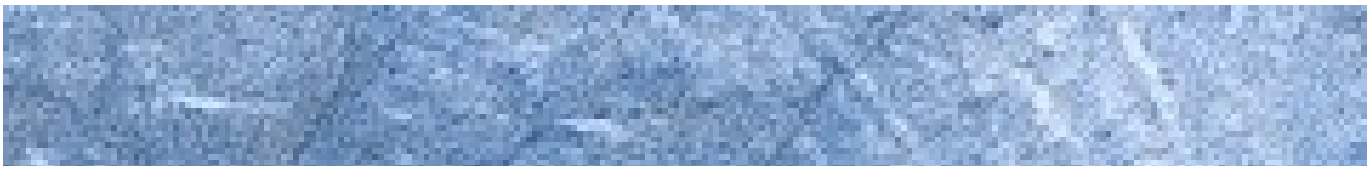
Słownik

> **NWD**
>
> akronim pojęcia „największy wspólny dzielnik”

Gra edukacyjna

Polecenie 1

Sprawdź swoją wiedzę o algorytmie Euklidesa, biorąc udział w grze.



Test

Algorytm Euklidesa. Quiz

Poziom trudności:

łatwy

Limit czasu:

5 min

Twój ostatni wynik:

-

Uruchom

Polecenie 2

Zastanów się, które pytania sprawiły ci trudność. Wróć do fragmentów, których dotyczą.

Polecenie 3

Zaproponuj trzy pytania, które mogłyby pojawić się w grze edukacyjnej.

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. Przetestuj jego działanie dla liczb 51, 34, 45.

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

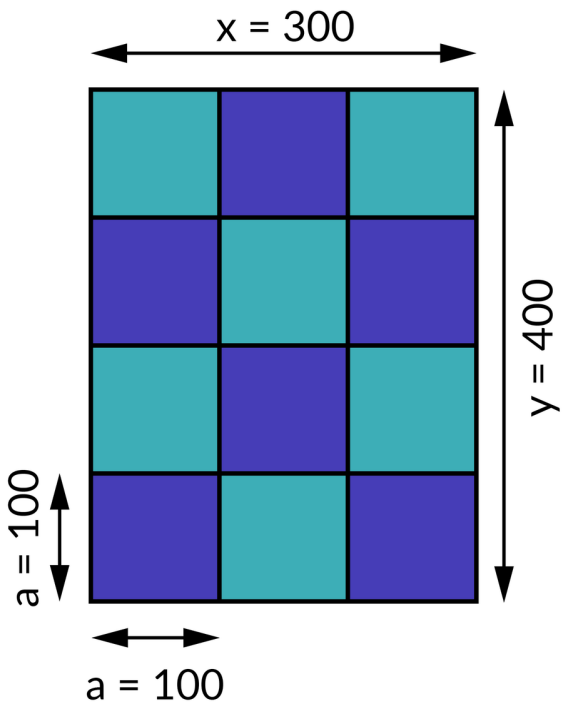
Bajtazar chce wyremontować posadzkę w swojej kuchni o wymiarach x na y. W tym celu chce zakupić kwadratowe kafelki o boku długości a (jednostki są ustandaryzowane). Bajtazar chce kupić możliwie jak najmniej kafelek – zakładamy, że w sklepie będzie mógł dobrać dowolny rozmiar, który idealnie wpasuje się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

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.

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.



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.

Twoje zadania

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

Dla nauczyciela

Autor: Maurycy Gast
Przedmiot: Informatyka

Temat: Algorytm Euklidesa w języku Java

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres 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

I. Rozumienie, analizowanie i rozwiązywanie problemów.

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

2) do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych;

3) objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;

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):

- Scharakteryzujesz dwie wersje algorytmu Euklidesa.
- Przeanalizujesz implementacje algorytmu Euklidesa w języku Java.
- Rozwiążesz zadania związane z algorytmem Euklidesa.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

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

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 Java SE 8 (lub nowszej wersji), w tym Eclipse 4.4 (lub nowszej wersji).

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Algorytm Euklidesa w języku Java". Uczniowie mają zapoznać się z treściami w sekcji „Przeczytaj" i wykonać obliczenia na podstawie dołączonych danych.

Faza wstępna:

1. Nauczyciel inicjuje rozmowę wprowadzającą w temat lekcji. Przedstawia cele zajęć oraz kryteria sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Nauczyciel prosi wybranego ucznia lub uczniów o przedstawienie sytuacji problemowej związanej z tematem lekcji.

Faza realizacyjna:

1. Uczniowie indywidualnie rozwiązują test w postaci Gry edukacyjnej.
2. Nauczyciel dzieli klasę na grupy. Każda z nich przygotowuje 5 pytań podobnych do tych, jakie pojawiły się w grze. Grupy odpowiadają na pytania przygotowane przez inną grupę.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują ćwiczenia nr 1 i 3 z sekcji „Sprawdź się". Nauczyciel sprawdza poprawność wykonanych zadań, omawiając je wraz z uczniami.

Faza podsumowująca:

1. Nauczyciel ponownie wyświetla na tablicy temat lekcji zawarty w sekcji „Wprowadzenie" i inicjuje krótką rozmowę na temat zrealizowanych celów (czego uczniowie się nauczyli).
2. Wybrany uczeń podsumowuje zajęcia z programowania w Javie, zwracając uwagę na nabyte umiejętności.

Praca domowa:

1. Uczniowie wykonują ćwiczenie 2 z sekcji „Sprawdź się".
2. Uczniowie wykonują polecenie 2 z sekcji „Gra edukacyjna".

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla języka Java SE 8 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Eclipse 4.4 (lub nowszej wersji).

Wskazówki metodyczne:

- Uczniowie mogą wykorzystać treści w sekcjach: „Przeczytaj”, „Gra edukacyjna”, „Sprawdź się” jako materiał do lekcji powtórkowej.