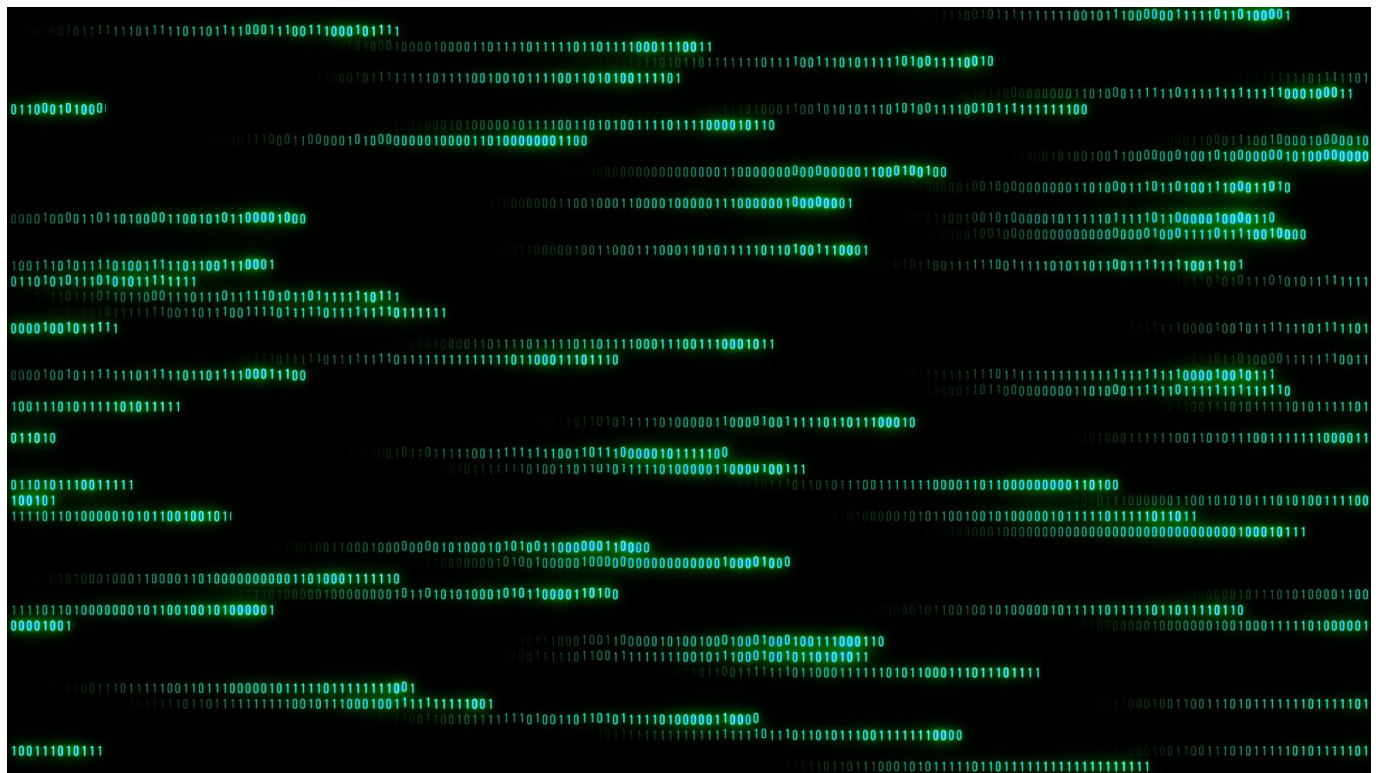
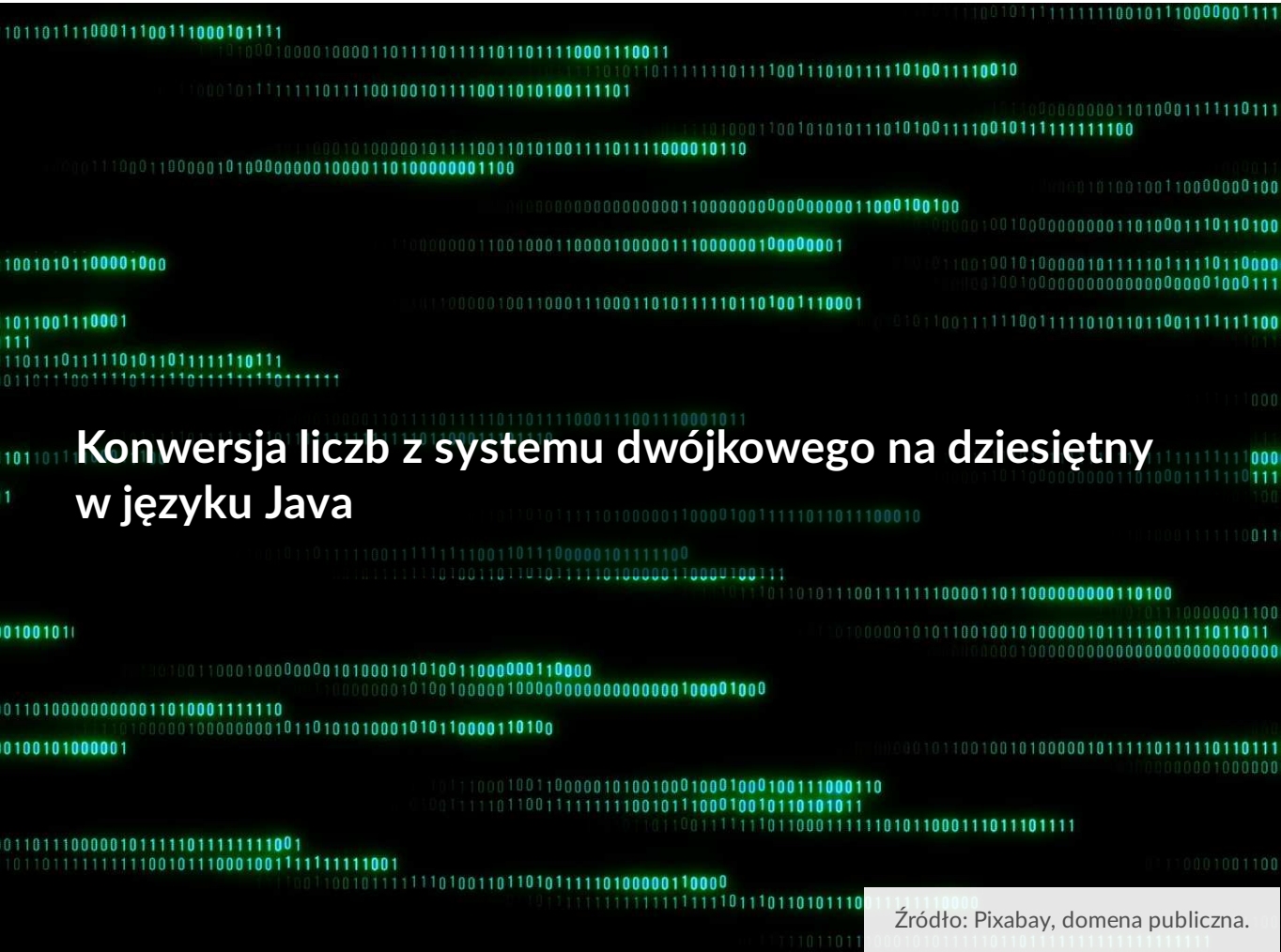




# Konwersja liczb z systemu dwójkowego na dziesiętny w języku Java

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Prezentacja multimedialna](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



## Konwersja liczb z systemu dwójkowego na dziesiętny w języku Java

Źródło: Pixabay, domena publiczna.

W tym e-materiale powtarzamy wiadomości ze szkoły podstawowej.

Dziesiętny system liczbowy jest wygodny dla nas w codziennym użytkowaniu, a zarazem kłopotliwy do zastosowania we współczesnych układach cyfrowych, takich jak procesory lub pamięci. Działanie wielu urządzeń oparte jest na systemie binarnym (dwójkowym).

Poznaliśmy już algorytm [konwersji liczb z systemu dwójkowego na dziesiętny](#). Teraz 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:

- [Konwersja liczb z systemu dwójkowego na dziesiętny w języku C++](#),
- [Konwersja liczb z systemu dwójkowego na dziesiętny w języku Python](#).

Więcej zadań? Sięgnij do [Konwersja liczb z systemu dwójkowego na dziesiętny – zadania naturalne](#).

### Twoje cele

- Zastosujesz w praktyce wiedzę na temat pozycyjnych systemów liczbowych.

- Prześledzisz, jak w języku Java zapisać program, służący do konwersji liczb całkowitych, zapisanych w systemie dwójkowym do postaci dziesiętnej.
- Zaimplementujesz w języku Java program przekształcający część ułamkową liczby dwójkowej do postaci dziesiętnej.

# Przeczytaj

---

Przypomnijmy metodę konwersji całkowitych liczb binarnych do postaci dziesiętnej. Posłużymy się przykładem, dzięki któremu łatwiej nam będzie przełożyć algorytm na program w języku Java.

Zapiszemy liczbę  $10111_{(2)}$  w systemie dziesiętnym:

$$10111_{(2)} \rightarrow ( )_{(10)}$$

Oto rozwiązanie zadania:

$$10111_{(2)} = ( )_{(10)}$$

$$\begin{aligned} 1 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 = \\ 1 + 2 + 4 + 0 + 16 = 23 \end{aligned}$$

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

Kolejne bity liczby dwójkowej mnożyliśmy przez odpowiadające im wagi, zaczynając od najmniej znaczącej cyfry (znajdującej się skrajnie po prawej stronie). Poszczególne wagi wynosiły:  $2^0, 2^1, 2^2, 2^3, 2^4 \dots$  (czyli były zwiększającymi się o 1 potęgami liczby 2). Następnie zsumowaliśmy otrzymane iloczyny i uzyskaliśmy dziesiętną postać liczby binarnej:

$$10111_{(2)} \rightarrow 23_{(10)}$$

## Realizacja algorytmu w języku Java

Rozpocznijmy od instrukcji, dzięki którym użytkownik poda liczbę do przekształcenia. Skorzystamy z wbudowanej klasy `Scanner` oraz strumienia wejścia `System.in`:

```
1 String liczbaTest;  
2 Scanner Scanner = new Scanner(System.in);  
3  
4 System.out.println("Podaj liczbę do przekonwertowania na system d  
5 liczbaTest = Scanner.nextLine();
```

Następnie zadeklarujemy funkcję `konwertujBinDec()`. Właśnie ona będzie odpowiadać za realizację omówionego algorytmu.

Funkcja zwróci wartość typu `int`, a jako `parametr` przyjmie ciąg znaków (`String`) `liczbaBin` – jest to liczba dwójkowa traktowana jako tekst.

Nie bez powodu podajemy liczbę do przekształcenia w postaci ciągu znaków, a nie zmiennej typu całkowitego. Tak jest po prostu wygodniej, nie musimy dokonywać jej konwersji do łańcucha znaków. Dodatkowo, podczas konwersji chcemy znać poszczególne cyfry (bity), a nie wartość całej liczby dwójkowej:

```
1 public static int konwertujBinDec(String liczbaBin) {
2     int wynik = 0;
3     int waga = 1;
4     int bit = 0;
5
6     for (int i = liczbaBin.length() - 1; i >= 0; i--) {
7         bit = Character.getNumericValue(liczbaBin.charAt(i));
8         wynik = wynik + bit * waga;
9         waga = waga * 2;
10    }
11
12    return wynik;
13 }
```

Omówmy poszczególne instrukcje.

W liniach 2., 3. oraz 4. deklarujemy trzy zmienne typu całkowitego: `wynik`, `waga` oraz `bit`:

- `wynik` – rezultat przekształcenia liczby dwójkowej zwracany przez funkcję `konwertujBinDec()`; przechowuje ona wartość konwertowanej liczby w systemie dziesiętnym;
- `waga` – przyjmuje wartości kolejnych wag bitów, czyli potęg liczby 2 (jako wartość początkową przypisujemy jej 1, ponieważ  $2^0 = 1$ );
- `bit` – przechowuje wartości kolejnych bitów zmiennej `liczbaBin`, uzyskane dzięki funkcji `getNumericValue()`, która służy do rzutowania zmiennej typu `char` na `int`.

Wewnątrz pętli `for` zapisujemy instrukcje służące do wyznaczenia wartości dziesiętnej liczby:

- `bit = Character.getNumericValue(liczbaBin.charAt(i))` – zamiana tekstu na wartość liczbową;
- `wynik = wynik + bit * waga` – sumowanie kolejnych iloczynów częściowych;

- $waga = waga * 2$  – obliczanie wagi następnego elementu (w wyniku mnożenia bieżącej wartości wagi przez 2).

Ostatnim krokiem jest wywołanie funkcji `konwertujBinDec()` i wyświetlenie wyniku:

```
1 System.out.println("Wynik: " + konwertujBinDec(liczbaTest));
```

A oto cały kod programu:

```
1 import Java.util.Scanner;
2
3 public class Bin2Dec {
4     public static int konwertujBinDec(String liczbaBin) {
5         int wynik = 0;
6         int waga = 1;
7         int bit = 0;
8
9         for (int i = liczbaBin.length() - 1; i >= 0; i--) {
10             bit = Character.getNumericValue(liczbaBin.charAt(i));
11             wynik = wynik + bit * waga;
12             waga = waga * 2;
13         }
14
15         return wynik;
16     }
17
18     public static void main(String[] args) {
19         String liczbaTest;
20
21         Scanner Scanner = new Scanner(System.in);
22         System.out.println("Podaj liczbę do przekonwertowania na
23         liczbaTest = Scanner.nextLine();
24         System.out.println("Wynik: " + konwertujBinDec(liczbaTest
25     }
26 }
```

## Konwersja części ułamkowej z systemu dwójkowego do systemu dziesiętnego

Sposób przekształcania części ułamkowej liczby binarnej do postaci dziesiętnej jest bardzo podobny do tego, który przed chwilą zastosowaliśmy. Zmieniają się wartości wag odpowiadających każdej cyfrze dwójkowej. Są nimi:  $2^{-1}$ ,  $2^{-2}$ ,  $2^{-3}$ ..., czyli odwrotności kolejnych potęg liczby 2. Przeciwny jest ponadto kierunek odczytywania bitów i obliczania iloczynów cząstkowych: posuwamy się od strony lewej do prawej.

Jeszcze raz dokonajmy konwersji, zajmując się częścią ułamkową liczby dwójkowej:

$$0,1011_{(2)} = ( \quad )_{(10)}$$

$$1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} =$$

$$0,5 + 0 + 0,125 + 0,0625 = 0,6875$$

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

Mnożyliśmy bity (zaczynając po przecinku od strony lewej i kierując się ku prawej) przez wagi przypisane poszczególnym pozycjom, czyli kolejne odwrotności potęgi liczby 2 (pierwszą była  $2^{-1}$ ). Po zsumowaniu iloczynów cząstkowych otrzymaliśmy ułamek zapisany w systemie dziesiętnym.

W kolejnej sekcji zaimplementujemy algorytm konwersji części ułamkowej z systemu dwójkowego do postaci dziesiętnej w języku Java.

## Polecenie 1

Zapoznaj się z filmem. Przedstawiono w nim dwie metody konwersji liczby całkowitej zapisanej w systemie dwójkowym na postać dziesiętną. Pierwszą z nich już znasz, druga opiera się na schemacie Hornera.



# Algorytm zamiany liczby bin→dec

Konwersja liczby dwójkowej (binarnej) na dziesiętną w języku Java



Film dostępny pod adresem </preview/resource/R7RpCNWMHH1na>

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

Film nawiązujący do algorytmu zmiany liczby bin na dec.

---

Plik o rozmiarze 899.00 B w języku polskim

## Słownik

### argument

element składni w określonym języku programowania, który w wyniku wywołania podprogramu zostaje utożsamiony (skojarzony) z określonym parametrem podprogramu

### najmłodszy bit

bit o najmniejszej wadze (najmniej istotny), zajmujący miejsce najdalej po prawej stronie liczby

### parametr

element składni języka, który jest definiowany przy deklaracji metody lub funkcji; deklarujemy go w nagłówku podprogramu; umożliwia komunikację pomiędzy podprogramem (funkcją/metodą), a programem go wywołującym

# Prezentacja multimedialna

---

## Polecenie 1

Napisz w języku Java algorytm konwersji części ułamkowej liczby dwójkowej do postaci dziesiętnej. Przetestuj go dla wybranych wartości.

### Specyfikacja problemu:

*Dane:*

- `ulamekTest` – zmienna typu `string`, przedstawiająca część ułamkową liczby w zapisie dwójkowym

*Wynik:*

- Na standardowym wyjściu wyświetlana jest wartość ułamka dwójkowego w systemie dziesiętnym.

### Twoje zadania

1. Przekonwertuj ułamek zapisany w systemie dwójkowym na system dziesiętny.

```
1 public class Bin2DecFraction {
2     public static double konwertujUlamek(String ulamekBin) {
3
4         // tutaj dopisz kod
5     }
6
7     public static void main(String[] args) {
8         String ulamekTest = "0,1011";
9         System.out.println(konwertujUlamek(ulamekTest));
10    }
11 }
```

## Polecenie 2

Porównaj swoje rozwiązanie z tym przedstawionym w prezentacji.



System dwójkowy to system liczbowy, który używany jest w systemach komputerowych, procesorach, pamięciach RAM, twardych dyskach etc.

Źródło: unsplash.com, CC 0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PeUBmOoeC>

Pisząc program przekształcający ułamek dwójkowy do postaci dziesiętnej, wykorzystamy algorytm przedstawiony w poprzedniej sekcji.

Zakładamy też, że liczba poddawana konwersji jest mniejsza od 1 (innymi słowy, nie zajmujemy się jej częścią całkowitą).

2

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PeUBmOoeC>

Zacznijemy od zbudowania szkieletu programu, czyli zadeklarowania funkcji głównej – `main()`:

```
1 public class
  Bin2DecFraction {
2     public static void
  main(String[] args) {
3
4     }
5 }
```

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PeUBmOoeC>

Następnie zapiszemy deklarację funkcji `konwertujUlamke()`. Zwróci ona zmienną typu `double`. Będzie to ułamek zapisany w systemie dziesiętnym:

```
1 public class
  Bin2DecFraction {
2     public static double
  konwertujUlamke(String
  ulamekBin) {
3
4     }
```

3

```

5
6     public static void
main(String[] args) {
7
8     }
9 }

```

4

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PeUBmOoeC>

Funkcja `konwertujUlamek()` przyjmie jako parametr zmienną `ulamekBin`, czyli liczbę dwójkową, która ma zostać przekształcona. Parametr będzie ciągiem znaków (typ `String`):

```

1 public class
  Bin2DecFraction {
2     public static double
konwertujUlamek(String
ulamekBin) {
3
4     }
5
6     public static void
main(String[] args) {
7
8     }
9 }

```

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PeUBmOoeC>

Zdefiniujemy następnie dwie zmienne liczbowe. Pierwsza z nich jest typu całkowitego (`int`) i nosi nazwę `bit`. Wstępnie przypiszemy jej

5

wartość 0. Będzie się ona później zmieniać na 1 lub ponownie na 0, zależnie od wartości bitu odczytanego ze zmiennej `ulamekBin`.

Druga zmienna jest typu `double`

. Jej nazwa to `waga`. W kolejnych etapach konwersji przyjmie ona wartość odpowiadającą potęgom odwrotności liczby 2. Wartością początkową jest 0.5, czyli  $2^{-1}$ :

```
1 public class
  Bin2DecFraction {
2     public static double
  konwertujUlamek(String
  ulamekBin) {
3         int bit = 0;
4         double waga = 0.5;
5     }
6
7     public static void
  main(String[] args) {
8
9     }
10 }
```

6

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PeUBmOoeC>

Potrzebna będzie także zmienna służąca do zapisywania wyniku. Musi być ona typu `double`

. Nadamy jej nazwę `wynik` i przypiszemy wartość początkową równą 0:

```
1 public class
  Bin2DecFraction {
2     public static double
  konwertujUlamek(String
  ulamekBin) {
```

```

3         int bit = 0;
4         double waga = 0.5;
5         double wynik = 0;

6     }

7
8     public static void
main(String[] args) {
9
10    }
11 }

```

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PeUBmOoeC>

Następnie otworzymy pętlę for

, dzięki której odczytamy kolejne bity  
konwertowanej części ułamkowej:

```

1 public class
  Bin2DecFraction {
2     public static double
konwertujUlamiek(String
ulamekBin) {
3         int bit = 0;
4         double waga = 0.5;
5         double wynik = 0;
6
7         for (int i = 2; i
< ulamekBin.length(); i++)
8         {
9             }
10    }
11
12    public static void
main(String[] args) {
13
14    }

```

Zauważ, że odczytywanie znaków ciągu `ulamekBin` zaczynamy od elementu o indeksie 2. Dzieje się tak dlatego, że pierwsze dwa znaki zmiennej to "0," (elementy `ulamekBin.charAt(0)` i `ulamekBin.charAt(1)` to - odpowiednio - "0" oraz ",").

8

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PeUBmOoeC>

Po odczytaniu wartości kolejnych bitów obliczamy cząstkowe iloczyny cyfr dwójkowych i odpowiadających im wag, a następnie sumujemy je ze sobą (zgodnie z algorytmem omówionym w poprzedniej sekcji):

```

1 public class
  Bin2DecFraction {
2     public static double
    konwertujUlamek(String
      ulamekBin) {
3         int bit = 0;
4         double waga = 0.5;
5         double wynik = 0;
6
7         for (int i = 2; i
          < ulamekBin.length(); i++)
8         {
9             bit =
              Character.getNumericValue(
                ulamekBin.charAt(i));
10            wynik = wynik
              + bit * waga;
11            waga = waga *
              0.5;
12        }

```

```

13         return wynik;
14     }
15
16     public static void
main(String[] args) {
17
18     }
19 }

```

Definicję funkcji kończy polecenie zwrócenia zmiennej wynik, czyli ułamka przekształconego do postaci dziesiętnej.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PeUBmOoeC>

9

Przedostatnim etapem pracy nad programem jest dopisanie kodu odpowiedzialnego za wymianę danych z użytkownikiem. Dodamy polecenia, dzięki którym będzie można wprowadzać liczbę do przekształcenia:

```

1 import java.util.Scanner;
2
3 public class
Bin2DecFraction {
4     public static double
konwertujUlamek(String
ulamekBin) {
5         int bit = 0;
6         double waga = 0.5;
7         double wynik = 0;
8
9         for (int i = 2; i
< ulamekBin.length(); i++)
10 {
            bit =
Character.getNumericValue(
ulamekBin.charAt(i));

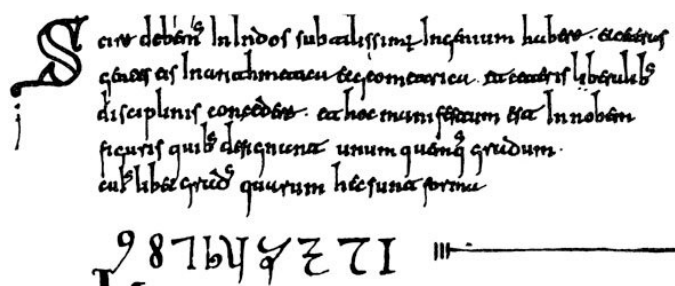
```

```

11         wynik = wynik
+ bit * waga;
12         waga = waga *
0.5;
13     }
14
15     return wynik;
16 }
17
18     public static void
main(String[] args) {
19         Scanner scanner =
new Scanner(System.in);
20
21
22         System.out.println("Podaj
ułamek w systemie
binarnym, aby dokonać jego
konwersji do systemu
dziesiętnego");
23         String ulamekTest
= scanner.nextLine();
24     }

```

10



Najstarsze znane wystąpienie liczb systemu dziesiętnego to hiszpański *Codex Albelensis*.

Źródło: en.wikipedia.org, domena publiczna.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PeUBmOoeC>

Na koniec dodajemy wywołanie funkcji `konwertujUlamke()` z argumentem `ulamekTest` i wyświetlamy wynik.

Oto końcowa wersja kodu:

```
1 import java.util.Scanner;
2
3 public class
4 Bin2DecFraction {
5     public static double
6     konwertujUlamek(String
7     ulamekBin) {
8
9         int bit = 0;
10        double waga = 0.5;
11        double wynik = 0;
12
13        for (int i = 2; i
14        < ulamekBin.length(); i++)
15        {
16            bit =
17            Character.getNumericValue(
18            ulamekBin.charAt(i));
19            wynik = wynik
20            + bit * waga;
21            waga = waga *
22            0.5;
23        }
24
25        return wynik;
26    }
27
28    public static void
29    main(String[] args) {
30        Scanner scanner =
31        new Scanner(System.in);
32
33        System.out.println("Podaj
34        ułamek w systemie
35        binarnym, aby dokonać jego
36        konwersji do systemu
37        dziesiętnego");
38
39        String ulamekTest
40        = scanner.nextLine();
41
42        System.out.println("Wynik:
43        " +
```

```
konwertujUlamiek(ulamiekTest
));
24     }
25 }
```

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

# Sprawdź się

---

Pokaż ćwiczenia:   

## Ćwiczenie 1



Przedstawiony program powinien konwertować dwójkową liczbę całkowitą do postaci dziesiętnej. Kod jest jednak niepełny i program nie działa. Dopisz brakujące instrukcje wewnątrz pętli `for` i uruchom program dla wprowadzonych danych.

### Specyfikacja problemu:

*Dane:*

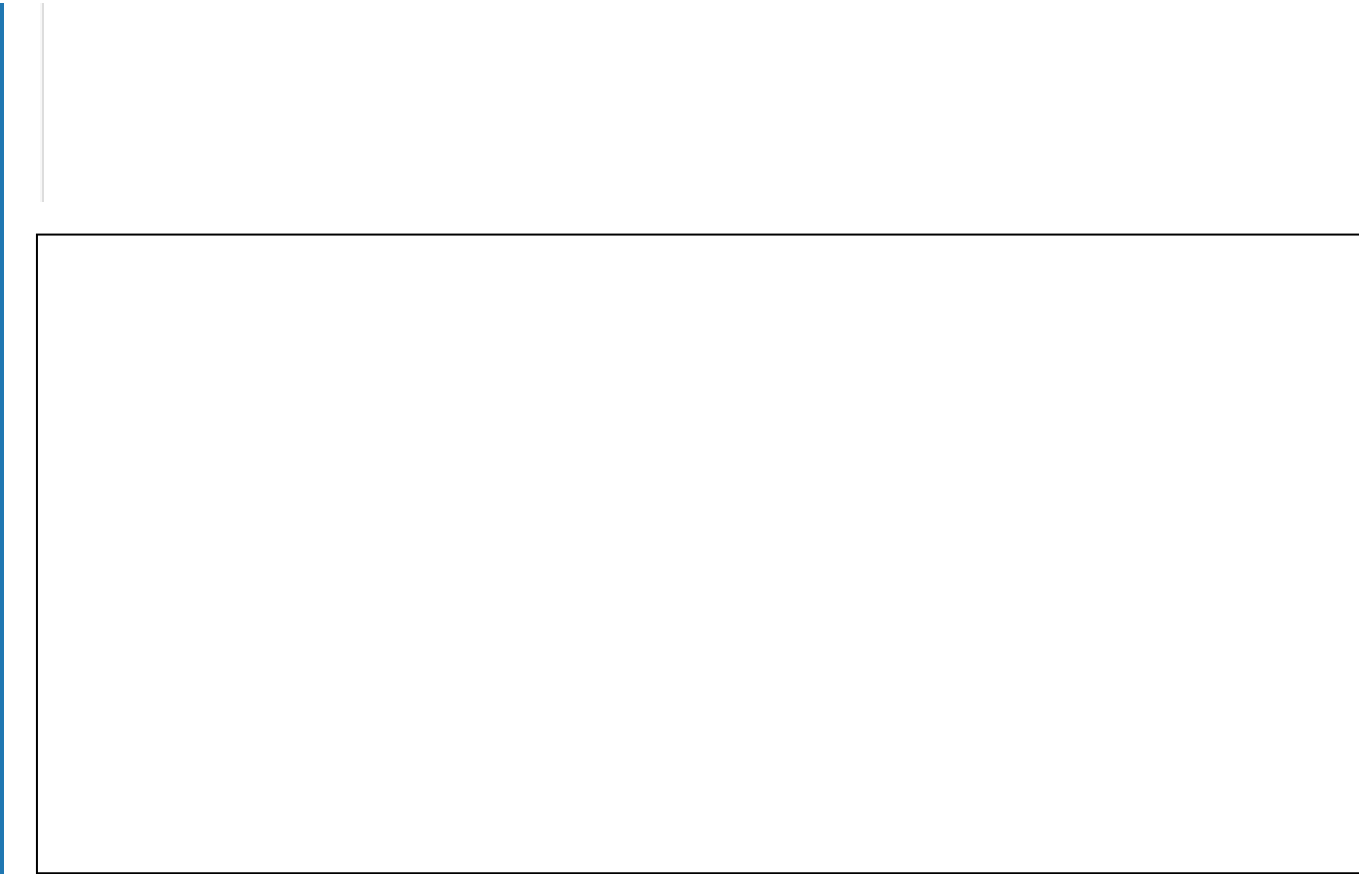
- `liczba` – zmienna typu `string`, zawierająca liczbę całkowitą zapisaną w systemie binarnym

*Wynik:*

- Na standardowym wyjściu wyświetlana jest `liczba` przekonwertowana do postaci dziesiętnej.

### Twoje zadania

1. Program ma dokonać konwersji liczby  $1101001_{(2)}$  do postaci dziesiętnej.





Przedstawiony program powinien konwertować część ułamkową liczby binarnej do postaci dziesiętnej. Kod jest jednak niepełny i program nie działa. Dopisz brakujące instrukcje wewnątrz pętli `for` i uruchom program dla wprowadzonych danych.

### Specyfikacja problemu:

#### *Dane:*

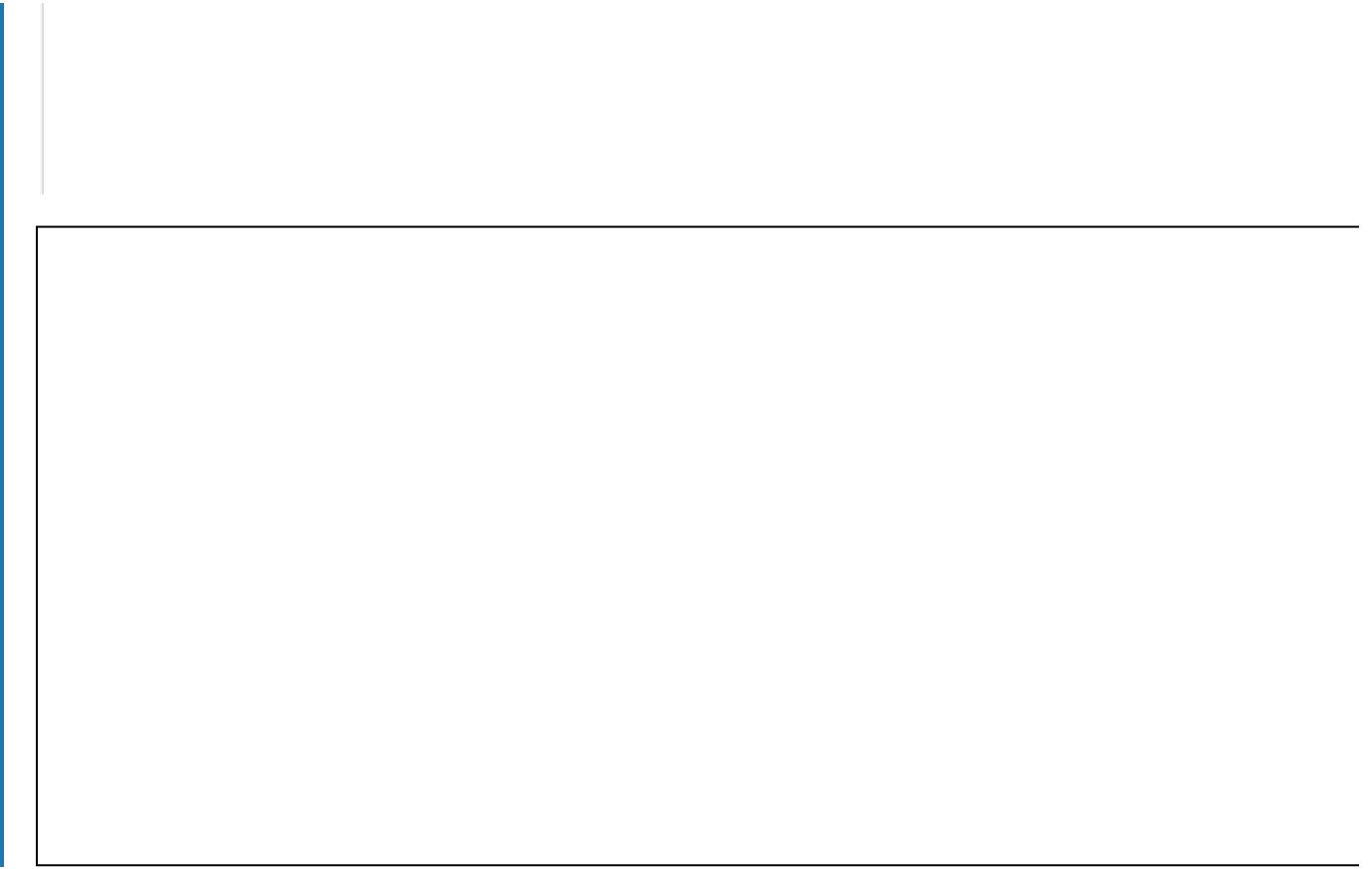
- `ułamek` – zmienna typu `string`, zawierająca część ułamkową liczby w systemie binarnym

#### *Wynik:*

- Na standardowym wyjściu wyświetlana jest wartość ułamka binarnego w systemie dziesiętnym.

### Twoje zadania

1. Program ma dokonać konwersji liczby  $0,11011_{(2)}$  do postaci dziesiętnej.



## Ćwiczenie 3



Napisz program, który przekształci liczbę dwójkową do postaci dziesiętnej.

### Specyfikacja problemu:

#### *Dane:*

- `liczba` – zmienna typu `string`, zawierająca liczbę rzeczywistą, zapisaną w postaci binarnej; liczba posiada część ułamkową

#### *Wynik:*

- Na standardowym wyjściu wyświetlana jest `liczba`, przekonwertowana do postaci dziesiętnej.

### Twoje zadania

1. Program ma dokonać konwersji liczby  $11010,1101_{(2)}$  do postaci dziesiętnej.



# Dla nauczyciela

---

**Autor:** Maurycy Gast

**Przedmiot:** Informatyka

**Temat:** Konwersja liczb z systemu dwójkowego na dziesiętny w języku Java

**Grupa docelowa:**

Szkoła ponadpodstawowa, liceum ogólnokształcące, technikum, zakres podstawowy

**Podstawa programowa:**

Cele kształcenia – wymagania ogólne

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

Treści nauczania – wymagania szczegółowe

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres podstawowy. Uczeń:

2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy:

a) na liczbach: badania pierwszości liczby, zamiany reprezentacji liczb między pozycyjnymi systemami liczbowymi, działań na ułamkach z wykorzystaniem NWD i NWW,

**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):**

- Zastosujesz w praktyce wiedzę na temat pozycyjnych systemów liczbowych.
- Prześledzisz, jak w języku Java zapisać program, służący do konwersji liczb całkowitych, zapisanych w systemie dwójkowym do postaci dziesiętnej.

- Zaimplementujesz w języku Java program przekształcający część ułamkową liczby dwójkowej do postaci dziesiętnej.

### **Strategie nauczania:**

- konstruktywizm;
- konektywizm.

### **Metody i techniki nauczania:**

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu 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 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: „Konwersja liczb z systemu dwójkowego na dziesiętny w języku Java”. Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj”.

#### **Faza wstępna:**

1. Prowadzący wyświetla na tablicy interaktywnej zawartość sekcji „Wprowadzenie” i omawia cele do osiągnięcia w trakcie lekcji.
2. **Rozpoznanie wiedzy uczniów.** Uczniowie tworzą pytania dotyczące tematu zajęć, na które odpowiedzą w trakcie lekcji.

#### **Faza realizacyjna:**

1. Uczniowie charakteryzują obie poznane metody, służące do konwersji liczb z systemu dwójkowego na dziesiętny.
2. Uczniowie rozwiązują problem postawiony w sekcji „Prezentacja multimedialna” w Poleceniu 1. Chętne lub wybrane osoby prezentują swój kod na forum klasy. Nauczyciel omawia go. Uczniowie porównują swoje rozwiązanie z prezentacją.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują ćwiczenia nr 1–2 z sekcji „Sprawdź się”. Nauczyciel sprawdza poprawność wykonanych zadań, omawiając je wraz z uczniami.

#### **Faza podsumowująca:**

1. Na koniec zajęć nauczyciel raz jeszcze wyświetla na tablicy temat lekcji i cele zawarte w sekcji „Wprowadzenie”. W odniesieniu do ich realizacji dokonuje szczegółowej oceny rozwiązania zastosowanego przez wybranego ucznia.
2. Na koniec zajęć z programowania w Javie nauczyciel prosi uczniów o rozwinięcie zdania: „Na dzisiejszych zajęciach nauczyłam/łem się jak...”.

#### **Praca domowa:**

1. Uczniowie wykonują ćwiczenie 3 z sekcji „Sprawdź się”.

#### **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”, „Prezentacja multimedialna”, „Sprawdź się” jako materiał do lekcji powtórkowej.