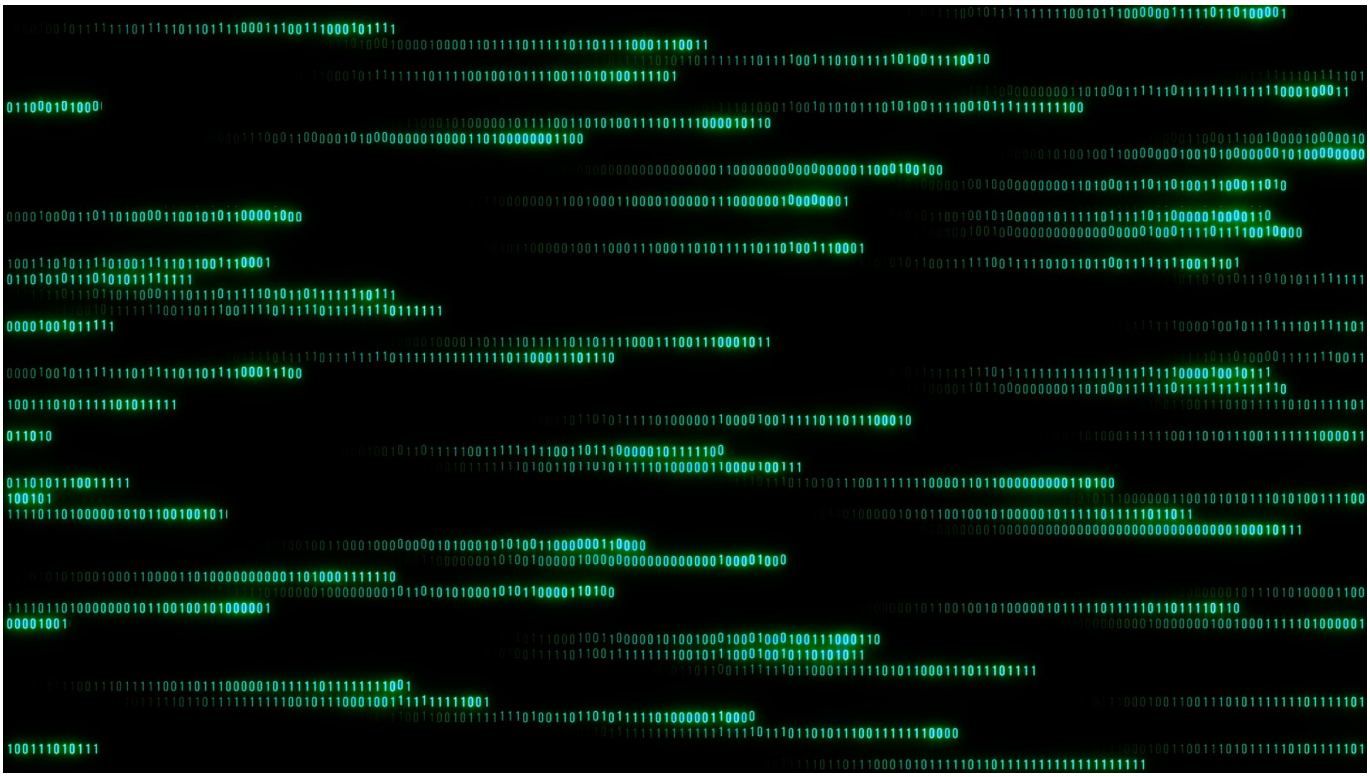
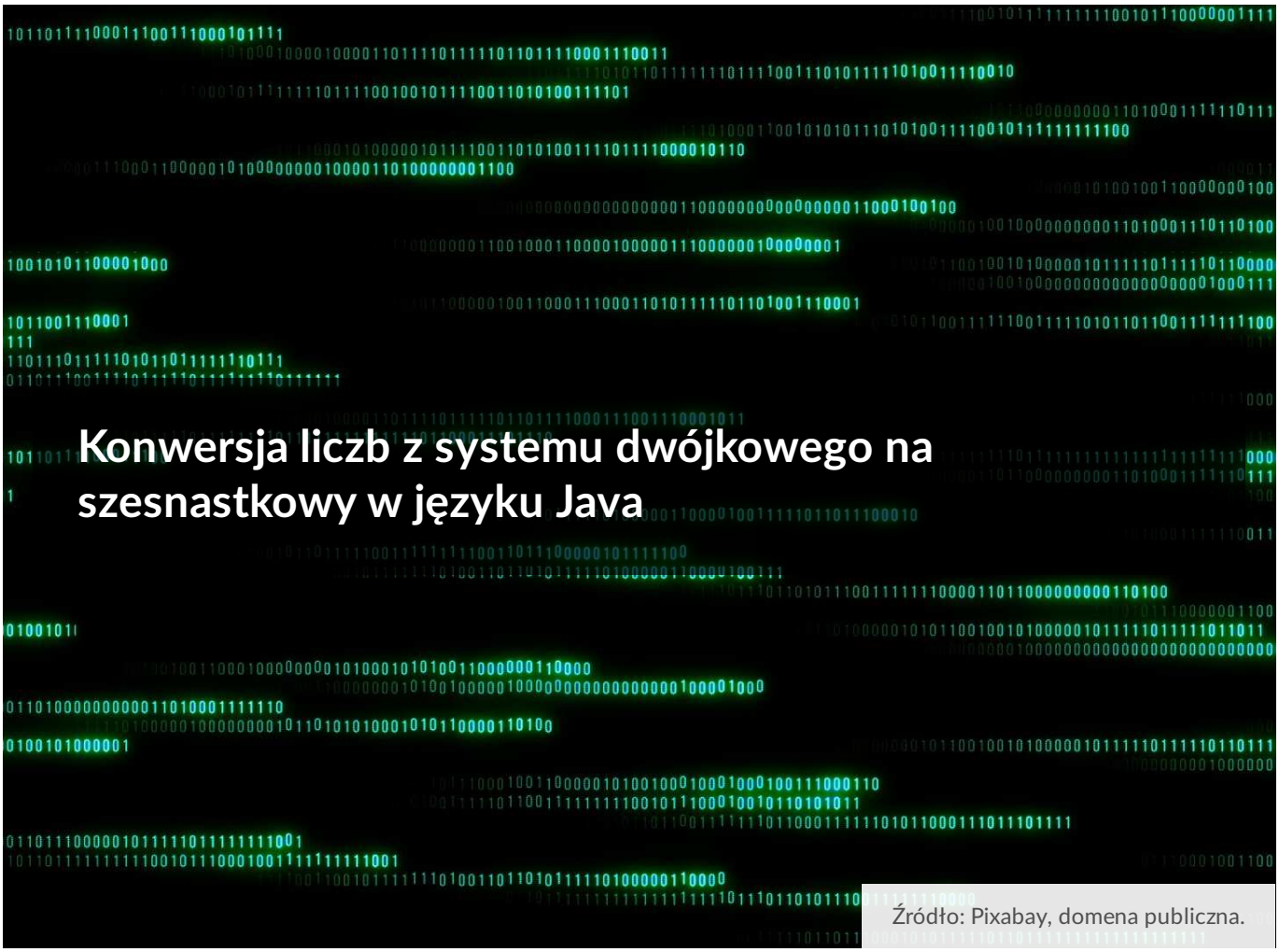


Konwersja liczb z systemu dwójkowego na szesnastkowy w języku Java



Źródło: Pixabay, domena publiczna.

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

System dwójkowy ma – z punktu widzenia człowieka – zasadniczą wadę: przedstawione w nim liczby bywają bardzo długie. Z tego właśnie powodu w pewnych sytuacjach, takich jak choćby podawanie adresów komórek pamięci, wygodniej jest posługiwać się systemem szesnastkowym. W porównaniu z systemem binarnym pozwala on czterokrotnie zmniejszyć długość zapisu liczby. Więcej na ten temat możesz przeczytać w e-materiale Konwersja liczb z systemu dwójkowego na szesnastkowy.

Umiejętność przedstawienia tej samej liczby w systemach o różnych podstawach przydaje się niejednokrotnie w pracy programisty. W tym e-materiale skupimy się na konwersji liczb binarnych do ich odpowiedników w systemie szesnastkowym. Napiszemy także program w języku Java, który będzie realizował to zadanie.

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

- Konwersja liczb z systemu dwójkowego na szesnastkowy w języku C++,
- Konwersja liczb z systemu dwójkowego na szesnastkowy w języku Python.

Więcej zadań? Sięgnij do: Konwersja liczb z systemu dwójkowego na szesnastkowy – zadania maturalne.

> **Twoje cele**
>
> - Przeanalizujesz algorytm zamiany liczby zapisanej w systemie binarnym na liczbę w systemie szesnastkowym.
> - Napiszesz program, który automatycznie dokona konwersji liczby z systemu binarnego na szesnastkowy.
> - Rozwiążesz kilka zadań związanych z tematem tego e-materiału.

Przeczytaj

Algorytm konwersji liczby z systemu binarnego na szesnastkowy

Przeanalizujmy zadanie polegające na konwersji liczb z systemu dwójkowego na szesnastkowy i zaimplementujmy realizujący je algorytm w języku Java.

Ważne!

W systemie o podstawie 16, oprócz cyfr z zakresu <0, 9>, dysponujemy także literami od A do F. Tabela prezentuje ich wartości dziesiętne.

Symbol	Wartość
A	10
B	11
C	12
D	13
E	14
F	15

Przykład 1

Naszym zadaniem jest przekonwertowanie liczby binarnej 10010101 na system szesnastkowy.

$$10010101_{(2)} = (\quad)_{(16)}$$

Gdybyśmy mieli do dyspozycji kartkę i długopis, najpierw przeprowadzilibyśmy konwersję z systemu binarnego na dziesiętny, a następnie na notację szesnastkową. Wyglądałoby to następująco:

$$10010101_{(2)} = 1 \cdot 2^7 + 1 \cdot 2^4 + 1 \cdot 2^2 + 1 \cdot 2^0 =$$

$$= 128 + 16 + 4 + 1 = 149_{(10)}$$

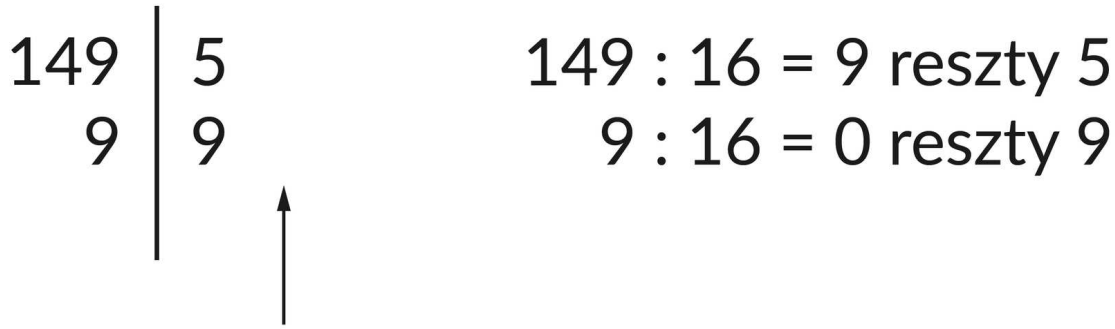
Aby przekonwertować liczbę z systemu dwójkowego do systemu dziesiętnego, obliczamy sumę iloczynów wszystkich cyfr danej liczby i odpowiadających im wag (czyli

w przypadku systemu binarnego kolejnych potęg liczby 2). Możemy w tym celu zastosować również schemat Hornera.

Oto wynik końcowy pierwszej konwersji:

$$10010101_{(2)} = 149_{(10)}$$

Otrzymany wynik konwertujemy następnie do systemu docelowego, czyli szesnastkowego.



Przeanalizujmy listę kroków algorytmu (`liczbaDec` oznacza liczbę, którą chcemy zamienić na system szesnastkowy):

1. Weź `liczbaDec` i podziel ją całkowicie przez podstawę systemu docelowego, czyli 16. W pokazanym wyżej przykładzie bierzemy liczbę 149 i dzielimy ją całkowicie przez 16.
2. Zapisz wynik reszty z dzielenia całkowitego z kroku 1. Reszta z dzielenia całkowitego 149 przez 16 to 5.
3. Zapisz wynik dzielenia całkowitego z kroku 1. do zmiennej `liczbaDec`. Wynik dzielenia całkowitego 149 przez 16 to 9.
4. Powtarzaj kroki 1-3, aż do osiągnięcia przez `liczbaDec` wartości 0. Kolejne dzielenie całkowite, które zostanie wykonane, to 9 przez 16, reszta z tego dzielenia to 9, a wynik dzielenia całkowitego to 0, co kończy powtarzanie dzielenia.
5. Wynikiem jest liczba szesnastkowa, składająca się z reszt z dzielenia zapisanych od końca. W naszym przykładzie jest to liczba 95.

Oto wynik zadania:

$$10010101_{(2)} = 149_{(10)} = 95_{(16)}$$

Realizacja algorytmu w języku Java

Przejdźmy do implementacji przedstawionego algorytmu w języku Java. Program będzie wykonywał następujące operacje:

- pobranie liczby binarnej z klawiatury,
- przekształcenie liczby binarnej na dziesiętną – do tego celu użyjemy wbudowanej funkcji `parseInt()`,
- przekształcenie liczby dziesiętnej na szesnastkową za pomocą funkcji `dec2Hex()`, którą napiszemy,
- wypisanie wyniku.

Zacznijmy od głównej funkcji programu:

```
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String liczbaBin = sc.next();
        if (liczbaBin.length() < 32) {
            int liczbaDec = Integer.parseInt(liczbaBin, 2);
            System.out.println(dec2Hex(liczbaDec));
        } else {
            System.out.println("Podaj liczbę binarną, która zawie
        }
    }
```

Konwertowaną liczbę binarną pobieramy z klawiatury i zapisujemy w zmiennej `liczbaBin`. Zanim użyjemy funkcji `parseInt()` sprawdzamy, czy podana liczba bitów jest mniejsza od 32, ponieważ wspomniana funkcja ma pewne ograniczenia.

Funkcja ta przyjmuje dwa argumenty: liczbę w postaci ciągu znaków oraz podstawę systemu, w którym zapisana jest liczba. Podana liczba musi zawierać tylko cyfry sytemu podanego jako podstawa. Ponieważ zakres wartości typu `Integer` to $<-2^{31}, 2^{31} - 1>$, największa liczba binarna, która zostanie poprawnie zamieniona, może składać się z co najwyżej 31 bitów o wartości 1.

W programie używamy więc instrukcji warunkowej, która sprawdza podaną liczbę bitów. Jeżeli jest ich mniej niż 32, liczbę binarną zamieniamy na dziesiętną przy użyciu funkcji `parseInt`, następnie wywołujemy funkcję `Dec2Hex()` oraz wypisujemy zwrócony rezultat konwersji. W przeciwnym wypadku wypisujemy stosowny komunikat i kończymy program.

Następny krok to utworzenie funkcji `dec2Hex()`, która przekształci podaną liczbę dziesiętną na szesnastkową.

```
public static String dec2Hex(int liczbaDec) {
```

```
2     if(liczbaDec == 0) {
3         return "0";
4     }
5     String liczbaHex = "";
6     int reszta = 0;
7
8     while (liczbaDec > 0) {
9         if (liczbaDec % 16 <= 9) {
10            reszta = liczbaDec % 16;
11            liczbaHex = reszta + liczbaHex;
12        } else {
13            reszta = liczbaDec % 16 - 10 + 'A';
14            liczbaHex = (char)reszta + liczbaHex;
15        }
16
17        liczbaDec = liczbaDec / 16;
18    }
19
20    return liczbaHex;
21 }
```

Funkcja zwraca ciąg znaków typu `string`. Konwersji dokonujemy poprzez dopisywanie („doklejanie”) do zmiennej `liczbaHex` kolejnych reszt z dzielenia przez 16.

Opis działania funkcji

W linii 2 sprawdzamy, czy konwertowana liczba jest równa 0, jeżeli tak, możemy od razu zwrócić 0 i zakończyć działanie funkcji (linia 3).

W przeciwnym wypadku deklarujemy zmienne:

- `String liczbaHex = ""` – w której zapisany jest wynik końcowy konwersji (linia 5),
- `int reszta = 0` – która przechowuje reszty z dzielenia przez 16 (linia 6).

Kolejny krok to wywołanie w linii 8. pętli `while`, która wykonuje instrukcje dopóty, dopóki `liczba_dec` jest większa od 0.

W linii 9. sprawdzamy, czy reszta z dzielenia przez 16 zawiera się w przedziale <0, 9>. Jeżeli warunek jest spełniony, w linii 10. zapisujemy wynik operacji `liczba_dec % 16` do zmiennej `reszta`, po czym w linii 11. dopisujemy („doklejamy”) go do `liczbaHex`.

W przeciwnym wypadku w linii 13 do zmiennej `reszta` wpisujemy wynik operacji `liczba_dec % 16 - 10 + A`. Odejmując `10` i dodając `A`, spowodujemy, że w zmiennej `reszta` będą mogły się znajdować jedynie kody ASCII znaków od `A` do `F`.

Następnie w linii 14. wykonujemy rzutowanie zmiennej `reszta`, która jest liczbą całkowitą, na typ `char`. W wyniku tej operacji liczba zapisana w zmiennej `reszta` zmienia się w odpowiadający jej znak z tablicy ASCII. Następnie zmienną `reszta` doklejamy do `liczbaHex`.

Zgodnie z algorytmem w 17 linii funkcji wykonujemy dzielenie całkowite zmiennej `liczbaDec` przez podstawę systemu.

```
1 liczbaDec = liczbaDec / 16;
```

Na końcu w linii 20. zwracamy wynik – jest on zapisany w zmiennej `liczbaHex`.

Poniżej kod całego programu zawierający omówioną funkcję:

```
1  import java.util.Scanner;
2
3  public class binary2hexadecimal {
4
5      public static String dec2Hex(int liczbaDec) {
6          if (liczbaDec == 0) {
7              return "0";
8          }
9          String liczbaHex = "";
10         int reszta = 0;
11
12         while (liczbaDec > 0) {
13             if (liczbaDec % 16 <= 9) {
14                 reszta = liczbaDec % 16;
15                 liczbaHex = reszta + liczbaHex;
16             } else {
17                 reszta = liczbaDec % 16 - 10 + 'A';
18                 liczbaHex = (char) reszta + liczbaHex;
19             }
20
21             liczbaDec = liczbaDec / 16;
22         }
23
```

```
24          return liczbaHex;
25      }
26
27
28      public static void main(String[] args) {
29          Scanner sc = new Scanner(System.in);
30          String liczbaBin = sc.next();
31          if (liczbaBin.length() < 32) {
32              int liczbaDec = Integer.parseInt(liczbaBin, 2);
33              System.out.println(dec2Hex(liczbaDec));
34          } else {
35              System.out.println("Podaj liczbę binarną, która zawie
36          }
37      }
38  }
```

W przedstawionym programie wykorzystaliśmy gotową wbudowaną funkcję `parseInt()`, która może posłużyć do zamiany liczby binarnej zawierającej nie więcej niż 31 bitów na liczbę dziesiętną. W filmie zamieszczonym w sekcji „Prezentacja multimedialna" zaprezentowane jest rozwiązanie oparte na schemacie Hornera pozbawione omówionych ograniczeń funkcji `parseInt()`.

Słownik

> **dwójkowy system pozycyjny**
>
> pozycyjny system zapisu liczb, którego podstawą jest liczba 2 - w zapisie tym występują wyłącznie cyfry 0 oraz 1
> **szesnastkowy system pozycyjny**
>
> pozycyjny system zapisu liczb, którego podstawą jest liczba 16 – w zapisie tym oprócz cyfr od 0 do 9 występują jeszcze znaki A, B, C, D, E, F, oznaczające odpowiednio cyfry od 10 do 15

Prezentacja multimedialna

Polecenie 1

Przeanalizuj prezentację, która pokazuje kolejne etapy implementacji algorytmu konwersji części ułamkowej liczby z systemu binarnego do systemu o podstawie 16.

1

Dwójkowy zegarek pokazujący godzinę 3:25.
Źródło: pl.wikipedia.org, dostęp: 24.01.2023, CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/PNeN4EI53

W prezentacji omówimy program, który realizuje algorytm konwersji części ułamkowej liczby z systemu binarnego na szesnastkowy w języku Java. W implementacji wykorzystamy bazy skojarzone.

Program ma działać w następujący sposób:

- pobieramy z klawiatury ułamkową część liczby binarnej, przy czym liczba powinna być wprowadzana bez wiodącego zera i separatora,
- z podanej liczby odczytujemy po 4 bity, zaczynając od najbardziej znaczących,

- odczytaną 4-bitową grupę zamieniamy na system szesnastkowy.

Opisana metoda jest poprawna, ponieważ podstawę systemu heksadecymalnego, czyli 16, można uzyskać przez podniesienie podstawy systemu binarnego, czyli 2, do czwartej potęgi.

2

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/PNeN4EI53

Zaczynamy od definicji funkcji `konwertujUlamek()`, która zrealizuje algorytm konwersji między systemami binarnym i szesnastkowym.

Funkcja przyjmuje jeden parametr `liczbaBin`. Jest to część ułamkowa w systemie dwójkowym, zapisana jako łańcuch znaków.

```
1  public static String
konwertujUlamek(String
liczbaBin) {
2
3      }
```

3

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/PNeN4EI53

W funkcji `konwertujUlamek()` deklarujemy zmienną `String liczbaHex = ""`, która posłuży do zapisywania wyniku konwersji.

```
1  public static String
konwertujUlamek(String
liczbaBin) {
```

```
        String liczbaHex =
"";
    }
```

4

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/PNeN4EI53

W kolejnym kroku zadbamy o to, aby liczba cyfr w części ułamkowej liczby binarnej była wielokrotnością 4. Jeżeli tak nie będzie, dodamy odpowiednią liczbę cyfr `0` do liczby od strony najmniej znaczącej.

```
 public static String
konwertujUlamek(String
liczbaBin) {
        String liczbaHex =
"";
        int dlugosc =
liczbaBin.length();

        if (dlugosc % 4 !=
0) {
            int ileBrakuje
= 4 - dlugosc % 4;

            for (int j =
0; j < ileBrakuje; j++) {
                liczbaBin
+= "0";
            }
        }
    }

```

Do zmiennej `dlugosc` przypisujemy liczbę cyfr liczby binarnej `liczbaBin`. Następnie w instrukcji warunkowej sprawdzamy, czy reszta z dzielenia długości liczby binarnej jest różna od zera. Jeżeli tak, do zmiennej `ileBrakuje`

przypisujemy liczbę brakujących znaków, czyli resztę z dzielenia przez 4 odjętą od 4: `int ileBrakuje = 4 - dlugosc % 4`.

Dla przykładu `liczbaBin` = "111100001" zawiera 9 cyfr. Wynikiem dzielenia `9 % 4` będzie 1. Aby liczba znaków była podzielna przez 4, musimy dodać 4 - 1, czyli 3 brakujące cyfry `0`.

Brakujące cyfry `0` dodajemy do liczby binarnej w pętli `for`, która wykonuje się tyle razy, ile wyniesie wartość zmiennej `ileBrakuje`.

5

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/PNeN4EI53

Dodamy teraz pętlę `do while`, w której pierwszą powtarzaną operacją będzie odczytywanie czteroznakowych grup znaków konwertowanej liczby binarnej.

```
1  public static String konwertujUlamek(String liczbaBin) {
2
3          ...
4
5          int i = 0;
6
7          do {
8              String bity = "";
9
10             for (int j = i; j < i + 4; j++) {
11                 bity = bity + liczbaBin.charAt(j);
12             }
13
```

```
14              ...
15
16              i = i + 4;
17          } while (i <
    dlugosc);
18
19          return liczbaHex;
20      }
21
```

Zmienną sterującą pętli jest `i`, której przypisujemy początkową wartość 0. W każdej iteracji jej wartość zwiększamy o 4. Pętla zakończy się, kiedy warunek `i < dlugosc` stanie się fałszywy, czyli po odczytaniu wszystkich znaków.

Do odczytywania czteroznakowych grup znaków liczby binarnej używamy pętli `for`. Zmienna `j` przyjmuje jako początkowe kolejne wartości zmiennej `i`, np. w pierwszym wykonaniu pętli zewnętrznej będzie to 0, w drugim 4. W instrukcji `liczbaBin.charAt(j)` zmienna `j` jako indeks pozwala odczytać kolejne 4 znaki liczby binarnej. Odczytane znaki zapisujemy w zmiennej `bity`.

Po zakończeniu pętli `while` umieszczamy instrukcję `return`, która zwróci zapisaną w zmiennej `liczbaHex` szesnastkową reprezentację liczby binarnej.

6

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/PNeN4EI53

Wewnątrz pętli `do while`, zaraz po pętli `for` dodajemy polecenie `switch`.

```
1
```

```
            switch (bity)
{
                case "0000":
 liczbaHex = liczbaHex + "0";
                    break;
                case "0001":
 liczbaHex = liczbaHex + "1";
                    break;
                case "0010":
 liczbaHex = liczbaHex + "2";
                    break;
                case "0011":
 liczbaHex = liczbaHex + "3";
                    break;
                case "0100":
 liczbaHex = liczbaHex + "4";
                    break;
                case "0101":
 liczbaHex = liczbaHex + "5";
                    break;
                case "0110":
 liczbaHex = liczbaHex + "6";
                    break;
```

```
                case "0111":
                    liczbaHex = liczbaHex + "7";
                    break;
                case "1000":
                    liczbaHex = liczbaHex + "8";
                    break;
                case "1001":
                    liczbaHex = liczbaHex + "9";
                    break;
                case "1010":
                    liczbaHex = liczbaHex + "A";
                    break;
                case "1011":
                    liczbaHex = liczbaHex + "B";
                    break;
                case "1100":
                    liczbaHex = liczbaHex + "C";
                    break;
                case "1101":
                    liczbaHex = liczbaHex + "D";
                    break;
                case "1110":
```

```
46
                liczbaHex = liczbaHex +
"E";
47                          break;
48                      case
"1111":
49
                liczbaHex = liczbaHex +
"F";
50                          break;
51                  }
52
```

Polecenie `switch` pozwala zamienić czteroznakową kombinację bitów zapisaną w zmiennej `bity` na odpowiednią cyfrę szesnastkową, która dopisywana jest do zmiennej `liczbaHex`.

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/PNeN4EI53

7

Przygotujemy teraz funkcję główną.

```
1
2       public static void
main(String[] args) {
3           String liczbaBin;
4
5           Scanner scanner =
new Scanner(System.in);
6           liczbaBin =
scanner.nextLine();
7
8
            System.out.println("0," +
konwertujUlamek(liczbaBin)
);
9       }
10
```

W funkcji `main()` za pomocą obiektu klasy `Scanner` pobieramy z klawiatury ułamkową część liczby binarnej i zapisujemy ją w zmiennej `liczbaBin`. Następnie wywołujemy funkcję `konwertujUlamek()`, przekazując jako argument pobraną liczbę. Wynik konwersji z dodanym wiodącym zerem i separatorem wypisujemy przy użyciu funkcji `println()`.

8

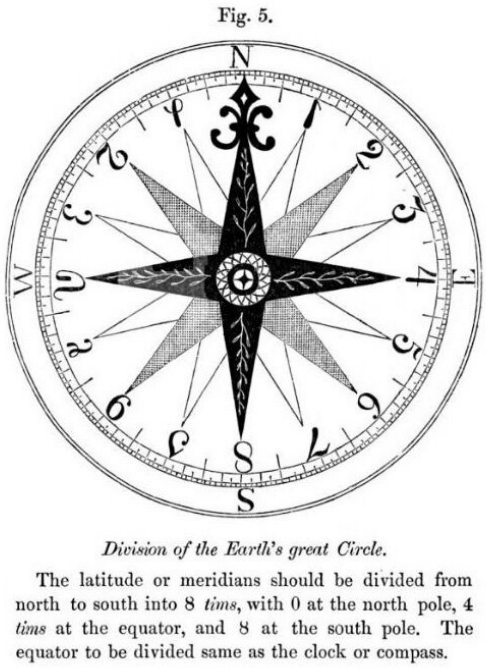


Fig. 5.

Division of the Earth's great Circle.

The latitude or meridians should be divided from north to south into 8 *tims*, with 0 at the north pole, 4 *tims* at the equator, and 8 at the south pole. The equator to be divided same as the clock or compass.

Projekt kompasu zaproponowany w XIX wieku przez Nystroma w systemie pozycyjnym szesnastkowym.

Źródło: John W. Nystrom, pl.wikipedia.org, dostęp: 24.01.2023, CC 0.

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/PNeN4EI53

Kompletny kod programu wraz z wymaganym importem klasy `Scanner` zamieszczamy poniżej:

```
1 import java.util.Scanner;
2
3 public class
  Bin2HexFraction {
```

```
4       public static String
konwertujUlamek(String
liczbaBin) {
5           String liczbaHex =
"";
6           int dlugosc =
liczbaBin.length();
7
8           if (dlugosc % 4 !=
0) {
9               int ileBrakuje
= 4 - dlugosc % 4;
10
11              for (int j =
0; j < ileBrakuje; j++) {
12                  liczbaBin
+= "0";
13              }
14          }
15
16          int i = 0;
17
18          do {
19              String bity =
"";
20
21              for (int j =
i; j < i + 4; j++) {
22                  bity =
bity +
liczbaBin.charAt(j);
23              }
24
25              switch (bity)
{
26                  case
"0000":
27
 liczbaHex = liczbaHex +
"0";
28                      break;
29                  case
"0001":
30
 liczbaHex = liczbaHex +
```

```
"1";
31                      break;
32                  case
"0010":
33
 liczbaHex = liczbaHex +
"2";
34                      break;
35                  case
"0011":
36
 liczbaHex = liczbaHex +
"3";
37                      break;
38                  case
"0100":
39
 liczbaHex = liczbaHex +
"4";
40                      break;
41                  case
"0101":
42
 liczbaHex = liczbaHex +
"5";
43                      break;
44                  case
"0110":
45
 liczbaHex = liczbaHex +
"6";
46                      break;
47                  case
"0111":
48
 liczbaHex = liczbaHex +
"7";
49                      break;
50                  case
"1000":
51
 liczbaHex = liczbaHex +
"8";
52                      break;
```

```
53                 case "1001":
54                     liczbaHex = liczbaHex + "9";
55                     break;
56                 case "1010":
57                     liczbaHex = liczbaHex + "A";
58                     break;
59                 case "1011":
60                     liczbaHex = liczbaHex + "B";
61                     break;
62                 case "1100":
63                     liczbaHex = liczbaHex + "C";
64                     break;
65                 case "1101":
66                     liczbaHex = liczbaHex + "D";
67                     break;
68                 case "1110":
69                     liczbaHex = liczbaHex + "E";
70                     break;
71                 case "1111":
72                     liczbaHex = liczbaHex + "F";
73                     break;
74             }
75 
76             i = i + 4;
```

```
        } while (i < dlugosc);

        return liczbaHex;
    }

    public static void main(String[] args) {
        String liczbaBin;

        Scanner scanner = new Scanner(System.in);
        liczbaBin = scanner.nextLine();


 System.out.println("0," + konwertujUlamek(liczbaBin) );
    }
}

```


Polecenie 2

Opracuj notatkę podsumowującą najważniejsze informacje przedstawione w prezentacji multimedialnej.

Polecenie 3

Porównaj program zapisany w sekcji „Przeczytaj” z przedstawionym w filmie.

Film dostępny pod adresem /preview/resource/RrwEUHq5iy9mn

Film nawiązujący do treści materiału: Algorytmy zamiany liczby bin na hex.

Plik z kodem źródłowym do pobrania:

Plik o rozmiarze 1.08 KB w języku polskim

Sprawdź się

Pokaż ćwiczenia: 

Ćwiczenie 1

Dany jest program przeprowadzający konwersję liczby całkowitej zapisanej w systemie dwójkowym na system szesnastkowy. Blok funkcji głównej jest niekompletny. Dopisz brakujący kod, tak aby program przy użyciu funkcji `parseInt()` oraz `dec2Hex()` konwertował podaną liczbę całkowitą z systemu binarnego na system szesnastkowy.

Działanie programu przetestuj dla liczby $101001010_{(2)}$.

Jeśli długość liczby przekracza 31 znaków, program powinien wypisać następujący komunikat:

```
Podaj liczbę binarną, która zawiera mniej bitów.
```

Specyfikacja problemu:

Dane:

- `liczbaBin` – łańcuch znaków; liczba zapisana w systemie dwójkowym

Wynik:

- `liczbaHex` – łańcuch znaków; liczba zapisana w systemie szesnastkowym

Twoje zadania

1. W funkcji głównej użyj funkcji `parseInt()` do przeprowadzenia konwersji liczby binarnej na dziesiętną, jeżeli długość liczby nie przekracza 31 znaków. W przeciwnym wypadku program powinien wyświetlić komunikat: `Podaj liczbę binarną, która zawiera mniej bitów`.
2. W funkcji głównej wypisz skonwertowaną wartość zwróconą przez funkcję `dec2Hex()`.

```java
public class Main {

    public static String dec2Hex(int liczbaDec) {
        if (liczbaDec == 0) {
            return "0";
```

```
6           }
7           String liczbaHex = "";
8           int reszta = 0;
9
10          while (liczbaDec > 0) {
11              if (liczbaDec % 16 <= 9) {
12                  reszta = liczbaDec % 16;
13                  liczbaHex = reszta + liczbaHex;
14              } else {
```

```
1
```

Ćwiczenie 2

Dany jest program przeprowadzający konwersję części ułamkowej liczby zapisanej w systemie dwójkowym na system szesnastkowy. Blok polecenia `switch` jest jednak niepełny. Dopisz brakujący kod, aby program działał poprawnie, czyli konwertował podaną część ułamkową liczby zapisanej w systemie binarnym na system szesnastkowy.

Działanie programu przetestuj dla części ułamkowej liczby binarnej: $10111_{(2)}$.

Specyfikacja problemu:

Dane:

- `liczbaBin` – łańcuch znaków; część ułamkowa liczby zapisanej w systemie dwójkowym

Wynik:

- `liczbaHex` – łańcuch znaków; część ułamkowa liczby zapisana w systemie szesnastkowym wypisana z poprzedzającymi znakami „0,” części całkowitej

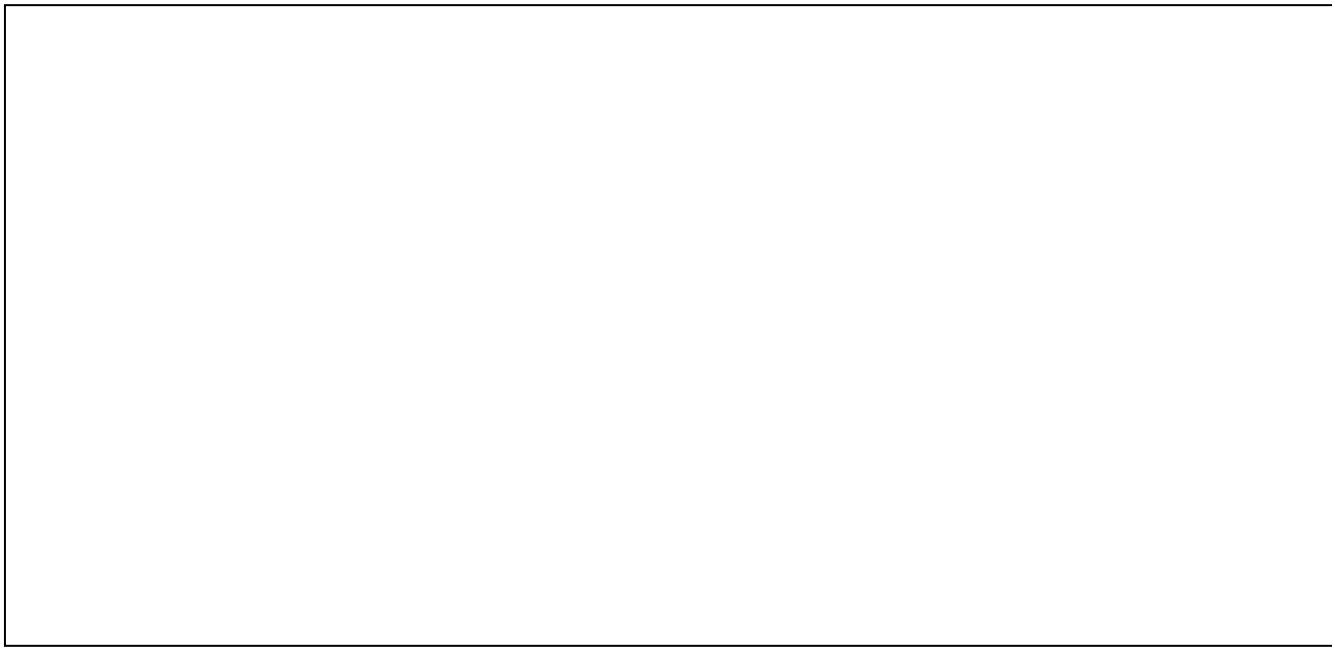
Twoje zadania

1. W funkcji `konwertujUlamek()` uzupełnij kod instrukcji `switch` tak, aby program poprawnie konwertował część ułamkową liczby binarnej na liczbę w systemie szesnastkowym.

```
1  public class Main {
2      public static String konwertujUlamek(String liczbaBin) {
3          String liczbaHex = "";
4          int dlugosc = liczbaBin.length();
5
6          if (dlugosc % 4 != 0) {
7              int ileBrakuje = 4 - dlugosc % 4;
8
9              for (int j = 0; j < ileBrakuje; j++) {
10                 liczbaBin += "0";
11             }
12         }
```

```
13
```

```
1
```



Ćwiczenie 3

Napisz program przeprowadzający konwersję liczby naturalnej z systemu binarnego na szesnastkowy, wykorzystując bazy skojarzone.

Działanie programu przetestuj dla liczby $1111111111100_{(2)}$.

Specyfikacja problemu:

Dane:

- `liczbaBin` – łańcuch znaków; liczba naturalna zapisana w systemie dwójkowym

Wynik:

- `liczbaHex` – łańcuch znaków; liczba zapisana w systemie szesnastkowym

Twoje zadania

1. Uzupełnij kod funkcji `konwertujLiczbe()` tak, aby funkcja konwertowała liczbę binarną na liczbę szesnastkową z wykorzystaniem baz skojarzonych.
2. Funkcja powinna zwracać reprezentację szesnastkową konwertowanej liczby binarnej.

```
public class Bin2Hex {
    public static String konwertujLiczbe(String liczbaBin) {
        // Tu uzupełnij kod
    }

    public static void main(String[] args) {
        System.out.println(konwertujLiczbe("1111111111100"));
    }
}
```

```
1
```

Ćwiczenie 4

Wykorzystując podane fragmenty kodu, napisz program, który zrealizuje konwersję liczby stałoprzecinkowej zapisanej w systemie dwójkowym na system szesnastkowy.

Działanie programu przetestuj dla liczby $10110{,}101101_{(2)}$.

Specyfikacja problemu:

Dane:

- `liczbaBin` – łańcuch znaków; liczba zapisana w systemie dwójkowym

Wynik:

- `liczbaHex` – łańcuch znaków; liczba zapisana w systemie szesnastkowym

Twoje zadania

1. Uzupełnij kod funkcji `bin2Hex()` tak, aby funkcja odczytała część całkowitą i ułamkową liczby binarnej, przeprowadziła ich konwersję oraz zwróciła ciąg znaków reprezentujących konwertowaną liczbę w systemie szesnastkowym.
2. Uzupełnij kod funkcji `bin2Dec()` tak, aby funkcja zwracała całkowitą liczbę binarną zamienioną na liczbę dziesiętną przy użyciu schematu Hornera.
3. Uzupełnij kod funkcji `konwertujCzescCalkowita()` tak, aby funkcja konwertowała część całkowitą liczby binarnej na szesnastkową.
4. Uzupełnij kod funkcji `konwertujCzescUlamkowa()` tak, aby funkcja konwertowała część ułamkową liczby binarnej na szesnastkową.

```
1 public class Main {
2
3     public static int bin2Dec(String liczbaBin) {
4         // Tu uzupełnij kod
5         return liczbaDec;
6     }
7
8     public static String konwertujCzescCalkowita(String
  czescCalkowita) {
```

```
        int czescCalkowitaDec = bin2Dec(czescCalkowita);
        String czescCalkowitaHex = "";
        int reszta = 0;

        // Tu uzupełnij kod
```

```
1
```

Dla nauczyciela

Autor: Maurycy Gast
Przedmiot: Informatyka

Temat: Konwersja liczb z systemu dwójkowego na szesnastkowy w języku Java

Grupa docelowa:

Szkoła ponadpodstawowa, 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.

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,

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

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

2) stosuje zasady programowania strukturalnego i obiektowego w rozwiązywaniu problemów;

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:

2) wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów:

b) wykonywania działań na liczbach w systemach innych niż dziesiętny,

Kształtowane kompetencje kluczowe:

- kompetencje obywatelskie;
- 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):

- Przeanalizujesz algorytm zamiany liczby zapisanej w systemie binarnym na liczbę w systemie szesnastkowym.
- Napiszesz program, który automatycznie dokona konwersji liczby z systemu binarnego na szesnastkowy.
- Rozwiążesz kilka zadań związanych z tematem tego e-materiału.

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 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 szesnastkowy w języku Java". Nauczyciel prosi uczniów o zapoznanie się z multimedium w sekcji „Prezentacja multimedialna".

Faza wstępna:

1. Nauczyciel wyświetla temat i cele zajęć zawarte w sekcji „Wprowadzenie". Następnie wspólnie z uczniami ustala kryteria sukcesu.
2. Prowadzący prosi uczniów, aby zgłaszali swoje propozycje pytań do tematu. Jedna osoba może zapisywać je na tablicy. Gdy uczniowie wyczerpią swoje pomysły, a pozostały jakieś ważne kwestie do poruszenia, nauczyciel je dopowiada.

Faza realizacyjna:

1. **Praca z tekstem.** Nauczyciel ocenia, na podstawie informacji na platformie, stan przygotowania uczniów do zajęć. Jeżeli jest ono niewystarczające, prosi wybraną osobę o przedstawienie najważniejszych informacji z sekcji „Przeczytaj". W kolejnym kroku uczniowie pracując w parach, testują na swoich komputerach program omówiony w tej sekcji.
2. **Praca z multimedium.** Nauczyciel wyświetla zawartość sekcji „Prezentacja multimedialna". Uczniowie indywidualnie analizują przedstawiony w prezentacji program, a następnie powtarzają go i testują na swoich komputerach. Na forum klasy dzielą się swoimi spostrzeżeniami. Nauczyciel wyjaśnia niezrozumiałe kwestie.
3. **Ćwiczenie umiejętności.** Uczniowie, pracując w parach, wykonują ćwiczenie nr 1 z sekcji „Sprawdź się". Nauczyciel sprawdza poprawność pisanych kodów, porównuje je i omawia wraz z uczniami. Wskazuje najbardziej efektywne rozwiązanie.
4. Ćwiczenia nr 2 i 3 z sekcji „Sprawdź się" uczniowie wykonują w grupach czteroosobowych, a następnie porównują swoje odpowiedzi z inną grupą.

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, zwracając uwagę na nabyte umiejętności, omawia ewentualne problemy podczas rozwiązania ćwiczeń z programowania w języku Java.

Praca domowa:

1. Uczniowie wykonują ćwiczenie nr 4 z sekcji „Sprawdź się”.
2. Uczniowie wykonują polecenie nr 3 z sekcji „Prezentacja multimedialna”.

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 multimedium w sekcji „Prezentacja multimedialna” do przygotowania się do lekcji powtórkowej.