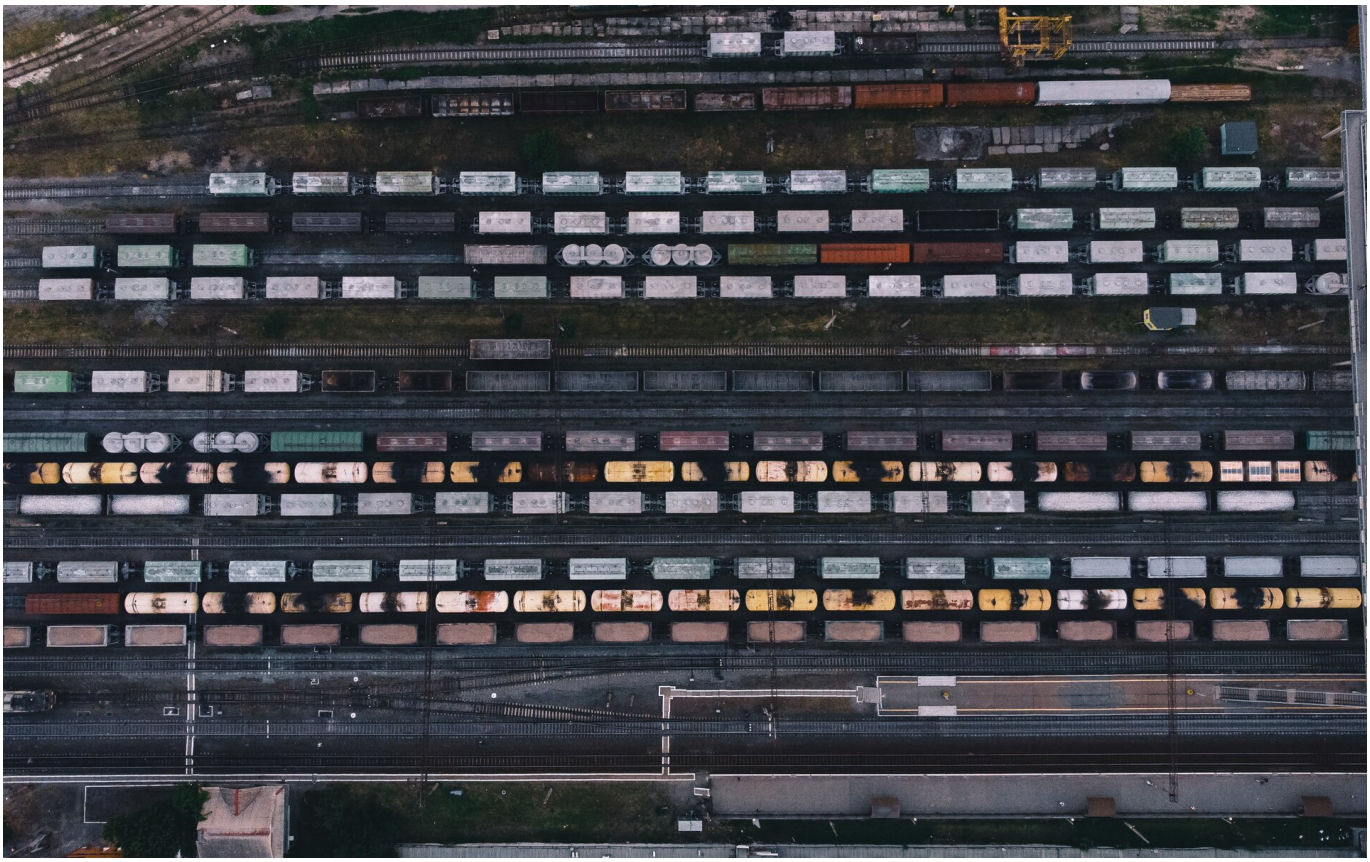
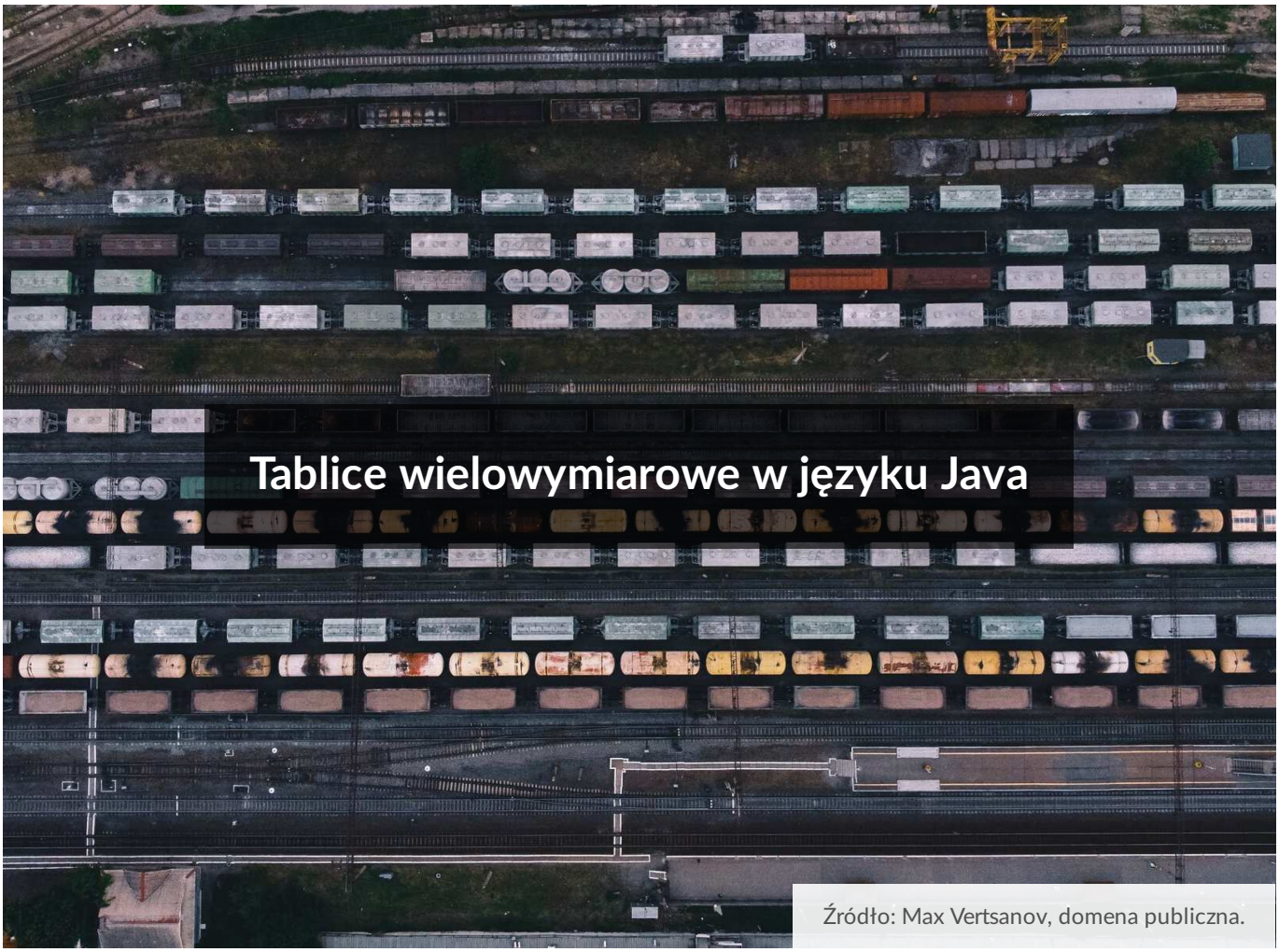


Tablice wielowymiarowe w języku Java



Źródło: Max Vertsanov, domena publiczna.

W e-materiale Tablice wielowymiarowe poznaliśmy definicję tego typu tablic. Czy potrafisz podać ich przykłady z życia codziennego?

Jeśli nie, przypomnij sobie popularną grę w statki. Plansza jest niczym innym jak przykładem tablicy wielowymiarowej, a dokładnie – tablicy dwuwymiarowej. Składa się ze 100 pól, każde jest oznaczone przez jedną literę oraz jedną liczbę i przechowuje pewną informację, w tym przypadku o obecności lub nieobecności statku.

W tym e-materiale zapoznamy się z implementacją tablic wielowymiarowych w języku Java.

Implementację tablic wielowymiarowych w wybranych językach programowania znajdziesz w e-materiałach:

- Tablice wielowymiarowe w języku C++,
- Tablice wielowymiarowe w języku Python.

Więcej zadań? Przejdź do e-materiału Tablice wielowymiarowe – zadania maturalne.

Informacje na temat tablic jednowymiarowych znajdziesz w e-materiałach:

- Tablice jednowymiarowe,
- Tablice jednowymiarowe w języku Java.

Twoje cele

- Prześledzisz i uporządkujesz informacje o tablicach.
- Przeanalizujesz, jak tworzyć oraz przetwarzać tablice dwuwymiarowe w języku Java.
- Stworzysz własne programy korzystające z tablic dwuwymiarowych.

Przeczytaj

Powtórzenie informacji o tablicach

Tablica jednowymiarowa

Tablica jednowymiarowa jest uporządkowanym zbiorem elementów tego samego typu. Elementy tablicy są numerowane (indeksowane). Pierwszy element ma numer 0.

Przykład tablicy jednowymiarowej, przechowującej 10 liczb całkowitych z przedziału ⟨`1, 10`⟩:

```
[1,2,3,4,5,6,7,8,9,10]
```

Więcej na temat tablic jednowymiarowych znajdziesz w e-materiale: Tablice jednowymiarowe.

Tablica wielowymiarowa

Tablica wielowymiarowa to tablica tablic, czyli tablica, której elementami są inne tablice. To, w jakim stopniu te tablice są „zagnieżdżone", ma wpływ na wymiar danej tablicy. I tak np. tablica dwuwymiarowa to tablica, której elementami są tablice jednowymiarowe, tablica trójwymiarowa to tablica, której elementami są tablice dwuwymiarowe itd.

Oto ułożenie elementów w tablicy dwuwymiarowej `dane` o rozmiarze 4 na 10:

```
    0  1  2  3  4  5  6  7  8  9
0 [[1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
1  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
2  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
3  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]]
```

W każdej komórce głównej tablicy jednowymiarowej (4-elementowej) znajduje się kolejna tablica jednowymiarowa (10-elementowa).

Nawigacja po takiej tablicy wygląda następująco:

1. Podajemy numer wiersza, czyli decydujemy, którą zagnieżdżoną tablicę chcemy wybrać.

2. Następnie podajemy numer kolumny, czyli wybieramy konkretną komórkę (przechowywaną wartość) wybranej tablicy zagnieżdżonej.

Podając `dane[3][2]`, wybieramy elementy znajdujące się w wierszu o indeksie trzy oraz kolumnie o indeksie dwa. W przypadku przedstawionej tablicy będzie to wartość 3.

Tworzenie tablic w języku Java

Tablica jednowymiarowa

Krótkie przypomnienie, jak tworzymy tablice w języku Java:

```
public class Main
{
    public static void main (String [] args)
    {
        /* Oto schemat tworzenia tablicy jednowymiarowej
        przechowującej 10 liczb całkowitych */

        int [] tablicaJednowymiarowa = new int [10];
    }
}
```

Tablica dwuwymiarowa

Jak już wspomnieliśmy, tablica dwuwymiarowa to technicznie tablica jednowymiarowa przechowująca inne tablice jednowymiarowe. Stwórzmy zatem tablicę dwuwymiarową, w której znajduje się sześć tablic przechowujących po pięć elementów.

Sposób I:

```
public class Main
{
    public static void main (String [] args)
    {
        int [][] tablicaDwuwymiarowa = new int [6][5];
    }
}
```

Sposób II:

```java
public class Main
{
    public static void main (String [] args)
    {
        int [][] tablicaDwuwymiarowa = new int [6][]; // tutaj de

        for (int i = 0; i <= tablicaDwuwymiarowa.length - 1; i++)
        {
            tablicaDwuwymiarowa[i] = new int [5]; // tutaj deklar
        }
    }
}
```

> Oba zaprezentowane sposoby są poprawne. W dalszej części materiału będziemy korzystać ze sposobu I.

W takim przypadku tablica prezentuje się następująco (w języku Java jej wartości są domyślnie ustawiane na 0):

```
    0  1  2  3  4
0 [[0, 0, 0, 0, 0],
1  [0, 0, 0, 0, 0],
2  [0, 0, 0, 0, 0],
3  [0, 0, 0, 0, 0],
4  [0, 0, 0, 0, 0],
5  [0, 0, 0, 0, 0]]
```

Przykładowe zadanie

Stwórzmy tablicę dwuwymiarową o wymiarach `m × n` i wypełnijmy ją kolejnymi liczbami naturalnymi. Wartości `m` oraz `n` podaje użytkownik.

Na początek utworzymy odpowiedni szkielet tablicy.

```java
import java.util.Scanner;

public class Main
{
    public static void main (String [] args)
    {
        Scanner scanner = new Scanner(System.in);
```

```
        int m = Integer.parseInt(scanner.next());
        int n = Integer.parseInt(scanner.next());

        int [][] tablicaLiczb = new int [m][n];
    }
}
```

Tablica jest już utworzona, więc możemy teraz przypisać do niej wartości liczbowe (używamy do tego zagnieżdżonej pętli `for`).

```
import java.util.Scanner;

public class Main
{
    public static void main (String [] args)
    {
        Scanner scanner = new Scanner(System.in);
        int m = Integer.parseInt(scanner.next());
        int n = Integer.parseInt(scanner.next());

        int [][] tablicaLiczb = new int [m][n];
        int liczba = 0;

        for (int i = 0; i <= tablicaLiczb.length - 1; i++)
        {
            for (int j = 0; j <= tablicaLiczb[i].length - 1; j++)
            {
                tablicaLiczb[i][j] = liczba;
                liczba++;
            }
        }
    }
}
```

> **Ważne!**
>
> Zapis `liczba++` oznacza skrócony zapis inkrementacji.

Stworzona tablica dla `m = 5` oraz `n = 5` prezentuje się następująco:

```
    0   1   2   3   4
0 [[0,  1,  2,  3,  4 ],
1  [5,  6,  7,  8,  9 ],
2  [10, 11, 12, 13, 14],
3  [15, 16, 17, 18, 19],
4  [20, 21, 22, 23, 24]]
```

Problem 1

Napisz program, który doda do siebie dwie tablice o rozmiarze `m` × `n` wypełnione liczbami całkowitymi. Przetestuj jego działanie dla tablicy o 4 kolumnach i 3 wierszach.

Specyfikacja problemu:

Dane:

- `n` – liczba kolumn; liczba naturalna
- `m` – liczba wierszy; liczba naturalna
- `a` – tablica liczb całkowitych o `n` kolumnach i `m` wierszach
- `b` – tablica liczb całkowitych o `n` kolumnach i `m` wierszach

Wynik:

Na standardowym wyjściu wyświetlona jest tablica będąca sumą tablic `a` i `b`.

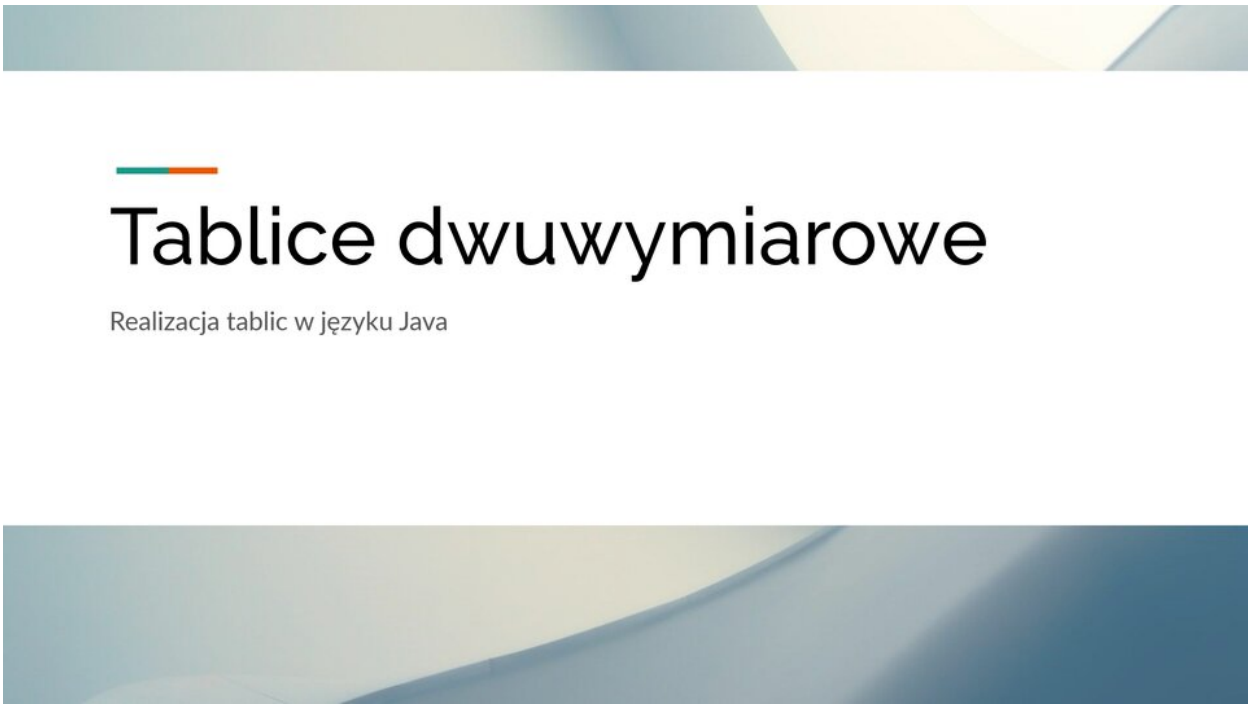
> **Ważne!**
>
> W filmie z rozwiązaniem zadania pada pojęcie *macierz*. W matematyce macierzą nazywamy układ liczb zapisany w postaci prostokątnej tablicy. Dodawanie macierzy wykonuje się dla macierzy o tych samych wymiarach. Polega ono na dodaniu do siebie elementów dwóch macierzy o tych samych indeksach.

```
1
```

```
1
```

Polecenie 1

Porównaj swoje rozwiązanie z przedstawionym w filmie.



Film dostępny pod adresem /preview/resource/RdW4FUz4mwmLR

Film nawiązujący do treści materiału: Realizacja tablic dwuwymiarowych w języku Java.

Plik o rozmiarze 734.00 B w języku polskim

Plik o rozmiarze 1.26 KB w języku polskim

Słownik

> **inkrementacja**
>
> operacja powodująca zwiększenie danej wartości o 1; przeciwieństwem inkrementacji jest dekrementacja, czyli zmniejszenie wartości o 1

Animacja

Polecenie 1

Przeanalizuj prezentację. Przedstawiony w niej program ma za zadanie stworzyć tablicę dwuwymiarową o wymiarach `m × n` i wypełnić ją losowymi liczbami całkowitymi z przedziału ⟨`a, b`⟩. Działanie programu przetestuj dla tablicy o wymiarach `5 × 5` i przedziału ⟨`a, b`⟩ równego ⟨`1, 10`⟩.

Specyfikacja problemu:

Dane:

- `tablicaLiczb[m][n]` – tablica dwuwymiarowa o wymiarach `m × n`
- `a`, `b` – liczby całkowite, granice przedziału
- `m` – liczba naturalna; liczba wierszy
- `n` – liczba naturalna, liczba kolumn

Wynik:

Na standardowym wyjściu wyświetlona jest tablica `tablicaLiczb` wypełniona losowymi liczbami całkowitymi.

1

Na początek tworzymy w kodzie szkielet tablicy dwuwymiarowej.

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

```
        int m = 5;
        int n = 5;
        int a = 1;
        int b = 10;
        int [][]
tablicaLiczb = new int[m]
[n];
    }
}
```

2

Teraz tworzymy obiekt poznanej na poprzednich lekcjach klasy `Random`, który będzie generować losowe liczby całkowite.

```
import java.util.Random;

public class Main
{
    public static void
main(String[] args)
    {
        int m = 5;
        int n = 5;
        int a = 1;
        int b = 10;
        int [][]
tablicaLiczb = new int[m]
[n];

        Random generator =
new Random();
    }
}
```

Następnym krokiem jest zdefiniowanie pierwszej pętli `for`, która umożliwi

3

przetwarzanie kolejnych elementów tablicy (są nimi tablice jednowymiarowe).

```
import java.util.Random;

public class Main
{
    public static void main (String [] args)
    {
        int m = 5;
        int n = 5;
        int a = 1;
        int b = 10;
        int [][] tablicaLiczb = new int[m][n];

        Random generator = new Random();

        for(int i = 0; i < m; i++)
        {

        }
    }
}
```

4

Kolejnym dodanym elementem będzie zagnieżdżona pętla `for`, umożliwiająca operacje na każdej komórce tablicy.

```
import java.util.Random;

public class Main
{
    public static void main (String[] args)
    {
        int m = 5;
```

```
        int n = 5;
        int a = 1;
        int b = 10;
        int [][] tablicaLiczb = new int[m][n];

        Random generator = new Random();

        for(int i = 0; i < m; i++)
        {
            for(int j = 0; j < n; j++)
            {

            }
        }
    }
}
```

Teraz napiszemy linię kodu, w której zapiszemy wygenerowane wartości do komórki.

5

```
import java.util.Random;

public class Main
{
    public static void main(String[] args)
    {
        int m = 5;
        int n = 5;
        int a = 1;
        int b = 10;
        int [][] tablicaLiczb = new int[m][n];

```

```
13          Random generator =
new Random();
14
15          for(int i = 0; i <
m; i++)
16          {
17              for(int j = 0;
j < n; j++)
18              {
19
 tablicaLiczb[i][j] =
generator.nextInt(b - a +
1) + a;
20              }
21          }
22      }
23  }
```

6

Na zakończenie dodamy fragment, który spowoduje wypisanie zawartości tablicy.

```
1  import java.util.Random;
2
3  public class Main
4  {
5      public static void
main(String[] args)
6      {
7          int m = 5;
8          int n = 5;
9          int a = 1;
10         int b = 10;
11         int [][]
tablicaLiczb = new int[m]
[n];
12
13         Random generator =
new Random();
14
15         for (int i = 0; i
< m; i++)
```

```
        {
            for (int j = 0; j < n; j++)
            {
 tablicaLiczb[i][j] = generator.nextInt(b - a + 1) + a;
 System.out.print(tablicaLiczb[i][j] + " ");
            }
 System.out.println();
        }
    }
}
```

Poniżej zobaczysz wyniki działania programu:

```
5 4 5 1 4
7 2 9 4 1
4 6 3 7 8
3 6 10 1 10
4 9 1 4 4
```

Są to oczywiście wartości losowe, więc jeżeli sprawdzisz działanie programu na swoim komputerze, bardzo prawdopodobnym jest, że zawartość tablicy będzie inna.

Polecenie 2

Przeanalizuj program, który ma za zadanie obliczyć sumę wartości na przekątnych kwadratowej tablicy dwuwymiarowej o wymiarach `n × n`.

Specyfikacja problemu:

Dane:

- `n` – liczba naturalna
- `tablicaLiczb[n][n]` – tablica dwuwymiarowa o wymiarach `n × n`

Wynik:

- `przekatna1` – suma liczb na przekątnej przechodzącej z lewego górnego do prawego dolnego rogu tablicy
- `przekatna2` – suma liczb na przekątnej przechodzącej z lewego dolnego do prawego górnego rogu tablicy

1

Na początek tworzymy w kodzie szkielet kwadratowej tablicy dwuwymiarowej. Rozumowanie przeprowadzimy na tablicy dla `n = 6`.

```
1 public class Main
2 {
3     public static void
  main(String[] args)
4     {
5         int n = 6;
6         int [][]
  tablicaLiczb = new int[n]
  [n];
7
```

```
        for(int i = 0; i < n; i++)
        {

 tablicaLiczb[i] = new int[6];
        }
    }
}
```

2

Tworzymy dwie zmienne: `przekatna1` oraz `przekatna2`, które będą przechowywać sumy wartości przekątnych.

```
public class Main
{
    public static void main(String[] args)
    {
        int n = 6;
        int [][] tablicaLiczb = new int[n][n];

        int przekatna1 = 0;
        int przekatna2 = 0;
    }
}
```

3

Teraz dodajemy potrzebne pętle `for`, które posłużą do przetwarzania tablicy. Ustalamy, że każda komórka ma przechowywać sumę jej indeksów (numer wiersza + numer kolumny).

```
public class Main
{
    public static void main(String[] args)
    {
        int n = 6;
        int [][] tablicaLiczb = new int[n][n];

        int przekatna1 = 0;
        int przekatna2 = 0;

        for(int i = 0; i < n; i++)
        {
            for(int j = 0; j < n; j++)
            {

 tablicaLiczb[i][j] = i + j;
            }
        }
    }
}
```

4

Zastanówmy się teraz, jak sprawdzić, czy podana komórka należy do danej przekątnej. Zmienna `przekatna1` ma przechowywać sumę wartości przekątnej zaczynającej się od komórki `[0][0]`, a kończącej się na komórce `[5][5]`.

Cechą tej przekątnej jest to, że oba indeksy są sobie równe, więc mamy już warunek dla pierwszej przekątnej.

```
public class Main
```

```
2  {
3      public static void
   main(String[] args)
4      {
5          int n = 6;
6          int [][]
   tablicaLiczb = new int[n]
   [n];
7
8          int przekatna1 =
   0;
9          int przekatna2 =
   0;
10
11         for(int i = 0; i <
   n; i++)
12         {
13             for(int j = 0;
   j < n; j++)
14             {
15
    tablicaLiczb[i][j] = i +
   j;
16
17                 if(i == j)
18                 {
19
   przekatna1 +=
   tablicaLiczb[i][j];
20                 }
21             }
22         }
23     }
24 }
```

Jeżeli chodzi o drugą przekątną, to zaczyna się ona na komórce `[5][0]`, a kończy na komórce `[0][5]`. Suma indeksów tych komórek jest równa 5 (wynika to z tego, że tablica ma

wymiary 6 x 6), a zatem mamy też warunek dla komórek drugiej przekątnej.

```
1  public class Main
2  {
3      public static void
   main(String[] args)
4      {
5          int n = 6;
6          int [][]
   tablicaLiczb = new int[n]
   [n];
7
8          int przekatna1 =
   0;
9          int przekatna2 =
   0;
10
11         for(int i = 0; i <
   n; i++)
12         {
13             for(int j = 0;
   j < n; j++)
14             {
15
    tablicaLiczb[i][j] = i +
   j;
16
17                 if(i == j)
18                 {
19
    przekatna1 +=
   tablicaLiczb[i][j];
20                 }
21
22                 if(i + j
   == n - 1)
23                 {
24
   przekatna2 +=
   tablicaLiczb[i][j];
25                 }
26             }
27         }
28     }
```

```
29 }
```

6

Czas na zadanie dla ciebie. Spróbuj w odpowiednim miejscu dodać instrukcję wypisania zawartości zmiennych `przekatna1` i `przekatna2`. W następnym bloku znajdziesz rozwiązanie.

7

Oto finalna wersja programu:

```
1  public class Main
2  {
3      public static void
   main(String[] args)
4      {
5          int n = 6;
6          int [][]
   tablicaLiczb = new int[n]
   [n];
7
8          int przekatna1 =
   0;
9          int przekatna2 =
   0;
10
11         for (int i = 0; i
   < n; i++)
12         {
13             for (int j =
   0; j < n; j++)
14             {
15
    tablicaLiczb[i][j] = i +
   j;
16
17                 if (i ==
   j)
```

```
18                  {
19
                     przekatna1 +=
                tablicaLiczb[i][j];
20                  }
21
22                  if (i + j
                == n - 1)
23                  {
24
                przekatna2 +=
                tablicaLiczb[i][j];
25                  }
26              }
27          }
28
                     System.out.println(przeka
                tna1 + " " + przekatna2);
29      }
30  }
```

Polecenie 3

Napisz program, który utworzy tablicę kwadratową o rozmiarze `m × m`, wypełni ją wielokrotnościami danej liczby naturalnej, a następnie wypisze liczby podzielne przez 3.

Specyfikacja problemu:

Dane:

- `wielokrotnosci` – liczba naturalna
- `m` – liczba naturalna
- `tab` – tablica kwadratowa liczb naturalnych o wymiarach `m × m`

Wynik:

Na standardowym wyjściu wyświetlone są liczby z tablicy `tab`, które są podzielne przez 3.

```
1
```

```
1
```

Polecenie 4

Zapoznaj się z animacją. Porównaj z nią swoje rozwiązanie. Zmodyfikuj swój program tak, żeby użytkownik mógł sam podać liczbę, której wielokrotności mają zostać zapisane.

Film dostępny pod adresem /preview/resource/RO1igfw5F1DMr

Film nawiązujący do treści materiału: Tablice wielowymiarowe w języku Java.

Kod programu zaprezentowanego w filmie:

Plik o rozmiarze 737.00 B w języku polskim

Sprawdź się

Pokaż ćwiczenia: 

Ćwiczenie 1

Napisz program, który wypisze zawartość jedynie tych wierszy tablicy `tablicaLiczb` o wymiarach `n × m` wypełnionej liczbami całkowitymi, w których wszystkie elementy są większe od liczby 2. Swój program przetestuj dla tablicy o wymiarach `10 × 3`, której każda komórka przyjmuje wartość sumy obu indeksów.

```
tablicaLiczb[i][j] = i + j
```

Specyfikacja:

Dane:

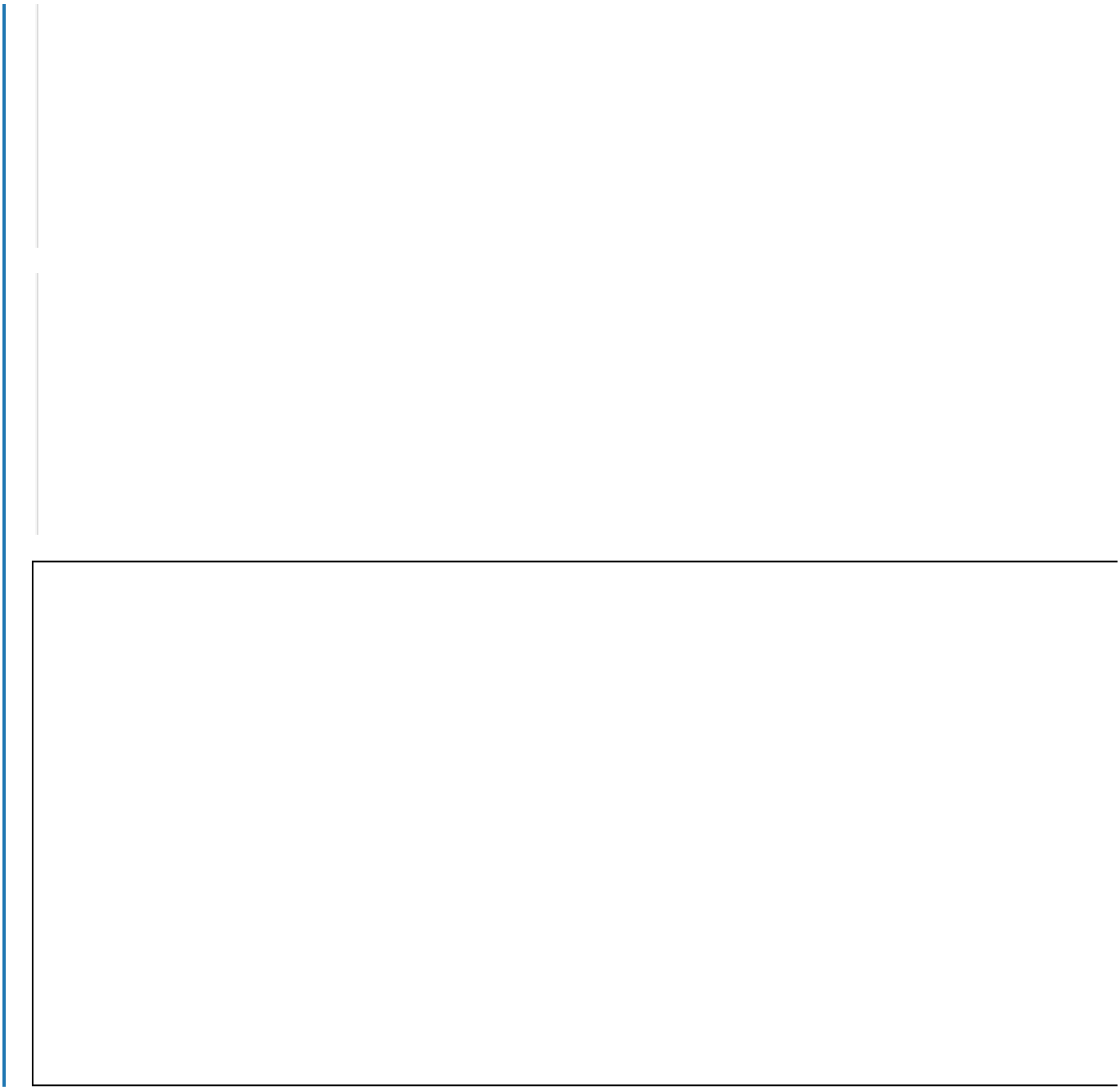
- `tablicaLiczb` – tablica o wymiarach `n × m`
- `n` – liczba naturalna
- `m` – liczba naturalna

Wynik:

Na standardowym wyjściu wyświetlane są te wiersze tablicy `tablicaLiczb`, których każdy element jest większy niż 2.

Twoje zadania

1. Program wyświetla na ekranie zawartość jedynie tych wierszy tablicy, których wszystkie elementy są większe od 2. Kolejne wiersze należy wypisać w nowej linii, zaś elementy w wierszu rozdzielić za pomocą spacji.



Ćwiczenie 2

Napisz program wypisujący kwadrat sumy wartości znajdujących się na obu przekątnych tablicy `tablicaLiczb` o wymiarach `n × n` wypełnionej liczbami całkowitymi (wartości wspólne mogą się powtarzać). Swój program przetestuj dla tablicy o wymiarach `5 × 5`, której każda komórka przyjmuje wartość sumy obu indeksów, np.:

```
tablicaLiczb[i][j] = i + j
```

Specyfikacja:

Dane:

- `tablicaLiczb` – tablica o wymiarach `n × n`
- `n` – liczba naturalna

Wynik:

Na standardowym wyjściu wyświetlana jest liczba całkowita – kwadrat sumy wartości znajdujących się na przekątnych.

Twoje zadania

1. Program wyświetla kwadrat sumy wartości leżących na przekątnych tablicy dwuwymiarowej. Sprawdź rozwiązanie dla `n = 5`.

Ćwiczenie 3

Napisz program, który wyszuka w tablicy `tablicaLiczb` o wymiarach n × n, której elementy są liczbami całkowitymi, największą wartość i wyświetli ją na ekranie. Swój program przetestuj dla tablicy o wymiarach 5 × 5, która jest wypełniona wartościami w taki sposób, że każda komórka `[i][j]` tabeli przyjmuje wartość $-(3 - i) \cdot (2 - j)$.

```
tablicaLiczb[i][j] = -(3-i)*(2-j)
```

Specyfikacja:

Dane:

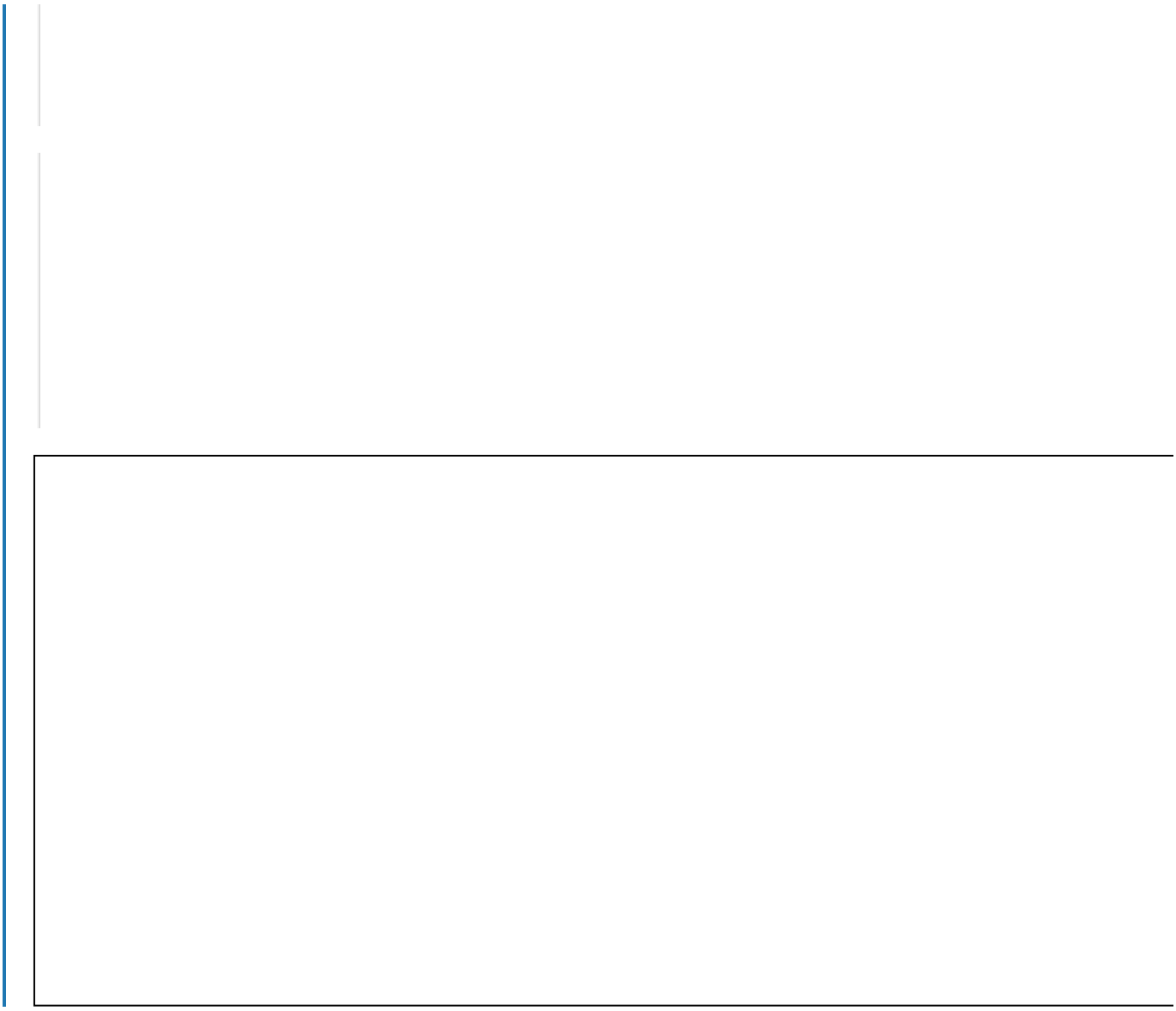
- `tablicaLiczb` – tablica o wymiarach n × n
- n – liczba naturalna

Wynik:

Na standardowym wyjściu wyświetlana jest liczba całkowita – największa wartość z tablicy `tablicaLiczb`

Twoje zadania

1. Program szuka największej wartości w tablicy i wyświetla ją na ekranie. Sprawdź rozwiązanie dla n = 5.



Dla nauczyciela

Autor: Maurycy Gast
Przedmiot: Informatyka

Temat: Tablice wielowymiarowe 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:

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 podstawowy. Uczeń:

1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2);

2) do realizacji rozwiązań problemów prawidłowo dobiera środowiska informatyczne, aplikacje oraz zasoby, wykorzystuje również elementy robotyki;

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

1) projektuje i tworzy rozbudowane programy w procesie rozwiązywania problemów, wykorzystuje w programach dobrane do algorytmów struktury danych, w tym struktury dynamiczne i korzysta z dostępnych bibliotek dla tych struktur;

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

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

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):

- Prześledzisz i uporządkujesz informacje o tablicach.
- Przeanalizujesz, jak tworzyć oraz przetwarzać tablice dwuwymiarowe w języku Java.
- Stworzysz własne programy korzystające z tablic dwuwymiarowych.

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: „Tablice wielowymiarowe w języku Java". Uczniowie zapoznają się z treściami w sekcji „Przeczytaj" (bez Problemu 1 oraz Polecenia 1).
2. Nauczyciel prosi uczniów o przypomnienie sobie najważniejszych informacji dotyczących tablic wielowymiarowych.

Faza wstępna:

1. Chętna lub wybrana osoba przypomina najważniejsze informacje dotyczące tablic wielowymiarowych.
2. Nauczyciel wyświetla uczniom temat zajęć oraz cele. Prosi, by na ich podstawie uczniowie sformułowali kryteria sukcesu.
3. **Rozpoznanie wiedzy uczniów.** Nauczyciel prosi wybranego ucznia lub uczniów o przedstawienie sytuacji problemowej związanej z tematem lekcji.

Faza realizacyjna:

1. **Praca z tekstem.** Nauczyciel sprawdza przygotowanie uczniów do lekcji. Jeśli jest ono niewystarczające prosi wybrane osoby o przedstawienie najważniejszych informacji z sekcji „Przeczytaj".
2. Uczniowie w parach rozwiązują Problem 1 z sekcji „Przeczytaj". Zapisują propozycję rozwiązania, następnie chętna lub wybrana osoba prezentuje swój program. Uczniowie zapoznają się z rozwiązaniem zaprezentowanym w filmie.
3. **Praca z multimedium.** Nauczyciel wyświetla zawartość sekcji „Animacja". Uczniowie w parach zapoznają się z Poleceniem 1. W parach przygotowują propozycję rozwiązania. Chętne lub wybrane pary przedstawiają swoją propozycję rozwiązania. Uczniowie zapoznają się z pierwszą prezentacją.
4. **Praca z multimedium.** Uczniowie indywidualnie zapoznają się z Poleceniem 2. Przygotowują propozycję rozwiązania. Chętne lub wybrane osoby przedstawiają swoją propozycję rozwiązania. Uczniowie zapoznają się z drugą prezentacją.
5. **Ćwiczenie umiejętności.** Uczniowie wykonują Ćwiczenia 1–3 z sekcji „Sprawdź się". Nauczyciel sprawdza poprawność wykonanych zadań, omawiając je wraz z uczniami.

Faza podsumowująca:

1. Nauczyciel ponownie wyświetla na tablicy temat i cele lekcji zawarte w sekcji „Wprowadzenie". W kontekście ich realizacji następuje omówienie ewentualnych problemów z rozwiązaniem ćwiczeń z sekcji „Sprawdź się".
2. Wybrany uczeń podsumowuje zajęcia z programowania w Javie, zwracając uwagę na nabyte umiejętności.

Praca domowa:

1. Uczniowie wykonują polecenia 3 i 4 z sekcji „Animacja".

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", „Animacja", „Sprawdź się" jako materiał do lekcji powtórkowej.