

PROGRAMOWANIE OBIEKTOWE JAVA – LABORATORIUM

TABLICE

TABLICE JEDNOWYMIAROWE

```
public class ArrayExample {  
  
    public static void main(String[] args) {  
        // deklaracja tablicy dataType[] arrayName;  
        // dataType - int, char, double, byte, etc. or Java objects  
        // arrayName - nazwa tablicy  
  
        // I sposob  
        double[] data;  
        //alokacja pamieci  
        data = new double[10];  
        // II sposob  
        double[] data1 = new double[10];  
  
        //inicjalizacja tablicy  
        int[] wiek = {12,4,45,32,24};  
  
        int[] wiek1 = new int[5];  
        wiek1[0]=12;  
        wiek1[1]=21;  
        wiek1[2]=5;  
        wiek1[3]=6;  
        wiek1[4]=46;  
  
    }  
}
```

Dostęp do elementów tablicy

```
public class ArrayExample {  
  
    public static void main(String[] args) {  
  
        //dostęp do elementow tablicy  
        int[] wiek = {12,4,45,32,24};  
        // access each array elements  
        System.out.println("Accessing Elements of Array:");  
        System.out.println("First Element: " + wiek[0]);  
        System.out.println("Second Element: " + wiek[1]);  
        System.out.println("Third Element: " + wiek[2]);  
        System.out.println("Fourth Element: " + wiek[3]);  
        System.out.println("Fifth Element: " + wiek[4]);  
  
    }  
}
```

Użycie pętli

```
public class ArrayExample {  
  
    public static void main(String[] args) {  
  
        int[] wiek = {12, 4, 5};  
  
        System.out.println("Using for Loop:");  
  
    }  
}
```

```

        for(int i = 0; i < wiek.length; i++) {
            System.out.println(wiek[i]);
        }
    }
}

```

Użycie pętli for-each

```

public class ArrayExample {

    public static void main(String[] args) {

        int[] wiek = {12, 4, 5};

        System.out.println("Using for Loop:");
        for(int i : wiek) {
            System.out.println(i);
        }
    }
}

```

TABLICE WIELOWYMIAROWE

```

public class ArrayExample {

    public static void main(String[] args) {

        //deklaracja
        int[][] matrixA = new int[3][4];
        String[][][] data = new String[3][4][2];

        //inicjalizacja tablicy wielowymiarowej
        int[][] a = {
            {1, 2, 3},
            {4, 5, 6, 9},
            {7},
        };

        double[][] matrix = {
            {1.2, 4.3, 4.0},
            {4.1, -1.1}
        };

        //wyświetlenie elementów tablicy
        System.out.println("Elementy tablicy:");
        for (int i = 0; i < a.length; ++i) {
            for(int j = 0; j < a[i].length; ++j) {
                System.out.print(a[i][j] + "\t");
            }
            System.out.println();
        }
    }
}

```

Jak widać, każdy element tablicy wielowymiarowej sam w sobie jest tablicą. Ponadto, w przeciwieństwie do C/C++, każdy wiersz tablicy wielowymiarowej w Javie może mieć różną długość.

```

public class ArrayExample {

    public static void main(String[] args) {

        // create a 2d array

```

```

int[][] a = {
    {1, -2, 3},
    {-4, -5, 6, 9},
    {7},
};

// first for...each loop access the individual array
// inside the 2d array
for (int[] innerArray: a) {
    // second for...each loop access each element inside the row
    for (int data : innerArray) {
        System.out.print(data + "\t");
    }
    System.out.println();
}
}

```

TABLICE TRZYWYMIAROWE

```

public class ArrayExample {

    public static void main(String[] args) {

        // test is a 3d array
        int[][][] test = {
            {
                {1, -2, 3},
                {2, 3, 4}
            },
            {
                {-4, -5, 6, 9},
                {1},
                {2, 3}
            }
        };

        // for..each loop to iterate through elements of 3d array
        for (int[][] array2D: test) {
            for (int[] array1D: array2D) {
                for(int item: array1D) {
                    System.out.print(item + "\t");
                }
                System.out.println();
            }
        }
    }
}

```

KOPIOWANIE TABLICY

```

public class ArrayExample {

    public static void main(String[] args) {

        int [] numbers = {1, 2, 3, 4, 5, 6};
        int [] positiveNumbers = numbers;    // copying arrays

        System.out.println("Tablica: ");
        for (int number: positiveNumbers) {
            System.out.print(number + ", ");
        }
    }
}

```

```

        System.out.println("\nKopia tablicy: ");
        for (int number: positiveNumbers) {
            System.out.print(number + ", ");
        }
    }
}

```

UŻYCIE PĘTLI PRZY KOPIOWANIU TABLICY

```

public class ArrayExample {

    public static void main(String[] args) {

        int [] source = {1, 2, 3, 4, 5, 6};
        int [] destination = new int[6];

        // iterate and copy elements from source to destination
        for (int i = 0; i < source.length; ++i) {
            destination[i] = source[i];
        }

        // converting array to string
        System.out.println(Arrays.toString(destination));
    }
}

```

W powyższym przykładzie użyliśmy pętli for do iteracji przez każdy element tablicy źródłowej. W każdej iteracji kopiujemy elementy z tablicy źródłowej do tablicy docelowej. W tym przypadku tablica źródłowa i docelowa odnoszą się do różnych obiektów (głęboka kopia). W związku z tym, jeśli elementy jednej tablicy zostaną zmienione, odpowiednie elementy innej tablicy pozostaną niezmienione. Metoda toString() służy do konwersji tablicy na ciąg.

KOPIOWANIE TABLICY Z WYKORZYSTANIEM METODY ARRAYCOPY()

```

arraycopy(Object src, int srcPos, Object dest, int destPos, int length)
//src - tablica źródłowa, którą chcemy skopiować
//srcPoc - pozycja początkowa (indeks) w tablicy źródłowej
//dest - tablica docelowa, do której elementy zostaną skopiowane ze źródła
//destPos - pozycja początkowa (indeks) w tablicy docelowej
//length - liczba elementów do skopiowania

```

```

public class ArrayExample {

    public static void main(String[] args) {

        int[] tab1 = {2, 3, 12, 4, 12, -2};
        int[] tab3 = new int[5];

        // tworzenie tablicy tab2 o długości tab1
        int[] tab2 = new int[tab1.length];

        // kopiowanie tablic1 do tab2
        System.arraycopy(tab1, 0, tab2, 0, tab1.length);
        System.out.println("tab2 = " + Arrays.toString(tab2));

        // kopiowanie elementów indeksu 2 do tab1
        // kopiowanie elementów indeksu 1 do tab3
        // 2 elementy zostaną skopiowane
        System.arraycopy(tab1, 2, tab3, 1, 2);
        System.out.println("tab3 = " + Arrays.toString(tab3));
    }
}

```

```

    }
}

METODA COPYOFRANGE()

public class ArrayExample {

    public static void main(String[] args) {

        int[] source = {2, 3, 12, 4, 12, -2};

        // copying entire source array to destination
        int[] destination1 = Arrays.copyOfRange(source, 0, source.length);
        System.out.println("destination1 = " +
Arrays.toString(destination1));

        // copying from index 2 to 5 (5 is not included)
        int[] destination2 = Arrays.copyOfRange(source, 2, 5);
        System.out.println("destination2 = " +
Arrays.toString(destination2));
    }
}

```

Zadania do samodzielnego rozwiązania:

1. Napisz program obliczający sumę i średnią elementów tablicy z użyciem pętli for oraz for each.
2. Napisz program, w którym zostanie utworzona 20-elementowa tablica typu boolean. Komórkom o indeksie parzystym przypisz wartości true, a o indeksie nieparzystym – false (zero możesz uznać za wartość parzystą). Zawartość tablicy wyświetl na ekranie.
3. Napisz program obliczający wariancję dla zestawu liczb, które mogą być np. pomiarami. Stwórz tablicę 1-wymiarową, wypełnioną liczbami typu double. Oblicz i wydrukuj wariancję dla tego zestawu liczb. Wzór na wariancję wygląda tak:

$$s^2 = \frac{\sum (x_i - x_s)^2}{N - 1}$$

s^2 - wariancja, x_i - kolejny pomiar, x_s - średnia z pomiarów, N - liczba pomiarów.

Wskazówka:

Kolejne pomiary, to kolejne liczby w naszej tablicy. Potrzebna będzie najpierw średnia z pomiarów, a do jej policzenia musimy mieć sumę wszystkich elementów. Można to zrobić przy użyciu pętli iterującej po tablicy. Sumę dzielimy przez liczbę elementów w tablicy, otrzymując średnią. Kolejnym krokiem jest obliczenie sumy kwadratów różnicy $x_i - x_s$. Ponieważ w kolejnym etapie należy odejmować średnią od kolejnych pomiarów, ponownie należy zastosować pętlę.