

PROGRAMOWANIE OBIEKTOWE JAVA – LABORATORIUM

TABLICE

Tablice to ciągi obiektów tego samego typu. Tablice mają określony podczas ich tworzenia rozmiar, który nie może ulec zmianie. Tablice to pierwszy rodzaj *kolekcji*. Kolekcje to zbiór powiązanych ze sobą elementów. Kolekcje są bardzo często wykorzystywane w programowaniu, ponieważ często mamy potrzebę przechowywać w programach tablice (bądź listy, zbiory itp.) wielu elementów, a nie pojedyncze wartości.

Wybrane metody używane w tablicach

length – rozmiar tablicy

charAt – pobranie znaku w tablicy

TABLICE JEDNOWYMIAROWE

```
public class Main {
    public static void main(String[] args) {
        /*definicja tablicy
        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];

        //III sposób
        String[] slowa;
        slowa = new String[] {"Ala","ma","kota"};

        //inicjalizacja tablicy
        int[] LiczbyCalkowite = {12,4,45,32,24};

        int[] Liczby = new int[5];
        Liczby[0]=12;
        Liczby[1]=21;
        Liczby[2]=5;
        Liczby[3]=6;
        Liczby[4]=46;
    }
}
```

Dostęp do elementów tablicy

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

```

String[] slowa;
slowa = new String[] {"Ala","ma","kota"};

System.out.println(slowa[0]+ " "+slowa[1]+" "+slowa[2]);
}

```

Metoda length

```

public static void Dlugosc(){
    int[] calkowite = new int[5];
    double[] rzeczywiste = { 3.14, 5, -20.5 };
    System.out.println(
        "Liczba elementow w tablicy calkowite: " +
calkowite.length);
    System.out.println(
        "Liczba elementow w tablicy rzeczywiste: " +
rzeczywiste.length);
}

public static void RozmiarArrayString(){
    String tekst = "Witajcie!";
    String[] slowa = { "Ania", "ma", "kota" };
    System.out.println("Liczba slow w zmiennej tekst: " + tekst.length());
    System.out.println("Liczba elementow w tablicy: " + slowa.length);
    System.out.println("Liczba znakow w pierwszym slowie z tablicy: " +
slowa[0].length());
}

```

Użycie pętli

```

public static void PetlaFor(){
    int[] Liczby = {12, 4, 5};

    System.out.println("Using for Loop:");
    for(int i = 0; i < Liczby.length; i++) {
        System.out.println(Liczby[i]);
    }
}
//wypisywanie elementów tablicy od ostatniego indeksu
public static void PetlaFor1(){
    int[] Liczby = {12, 4, 5};
    for (int i = Liczby.length - 1; i >= 0 ; i--) {
        System.out.print(Liczby[i] + " ");
    }
}

```

Użycie pętli for-each

```

public static void PetlaForEach(){
    int[] Liczby = {12, 4, 5};

    System.out.println("Using for Loop:");
    for(int i : Liczby) {
        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 elementow 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
//srcPos - 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 skopiowanie
```

```
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 tablicy tab1 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));
    }
}
```

Zmiana rozmiaru tablicy

```
//metoda do zmiany rozmiaru tablicy
public static void ChangeLength() {
    int[] liczby = {10, 100, -5};
    System.out.println("Elementy tablicy liczby:");
}
```

```

for (int x : liczby) {
    System.out.print(x + ", ");
}
System.out.println();
// tworzymy druga tablice o wiekszym rozmiarze
int[] tymczasowaTabela = new int[5];
// przepisujemy elementy z pierwszej tablicy
for (int i = 0; i < liczby.length; i++) {
    tymczasowaTabela[i] = liczby[i];
}
// ustawiamy dodatkowe elementy
tymczasowaTabela[3] = 20;
tymczasowaTabela[4] = 1;
// przypisujemy druga tablice do pierwszej
liczby = tymczasowaTabela;
System.out.println("Elementy tablicy liczby:");
for (int x : liczby) {
    System.out.print(x + ", ");
}
}

```

Pytania

1. Czy poniższy kod jest poprawny?

```

int[] tablica = { 1, 2, 3 };
System.out.println(tablica[3]);

```

2. Jaki będzie wynik działania poniższego programu, gdy wartość zmiennej szukanaLiczba będzie równa: 0 czy 500 ?

```

public static void main(String[] args) {
    boolean znaleziona = false;
    int[] tablica = { -20, 105, 0, 26, -99, 7, 1026 };
    int szukanaLiczba = ?; // pewna wartosc
    for (int i = 0; i <= tablica.length; i++) {
        if (tablica[i] == szukanaLiczba) {
            znaleziona = true;
            break; // znalezlismy liczbe - mozemy wiec przerwac petle
        }
    }
    if (znaleziona) {
        System.out.println("Liczba " + szukanaLiczba + " zostala
znaleziona!");
    } else {
        System.out.println("Liczba " + szukanaLiczba + " nie zostala
znaleziona.");
    }
}

```

3. Która z poniższych tablic jest zdefiniowana/utworzona niepoprawnie i dlaczego?

```

public static void main(String[] args) {
    int liczby = { 1, 2, 3 };
    String[] litery = { 'a', 'b', 'c' };
    String[] slowa = new String[];
    slowa = { "Ala", "ma", "kota" };
    double[] rzeczywiste = new double[] { 3.14, 2.44, 20 };
    double[] innaTablica = new int[3];
    int[] tablica = new int[5] { 1, 10, 100 };
    double[] kolejnaTablica = new double[3];
    kolejnaTablica = { 5, 10, 15 };
    String[] tab = { "Ala ma kota" };
}

```

Zadania do samodzielnego rozwiązania:

1. Napisz program obliczający sumę i średnią elementów tablicy z użyciem pętli for oraz for each. Elementy do tablicy mają zostać wylosowane z dowolnego przedziału. Należy zaproponować oddzielną metodę do losowania liczb.
2. Napisz program, który wypisze co drugi element zdefiniowanych przez Ciebie tablic. Pierwsza tablica powinna mieć parzystą liczbę elementów, a druga nieparzystą.
3. Napisz program, w którym zdefiniujesz tablicę wartości typu String i zainicjalizujesz ją przykładowymi wartościami. Skorzystaj z pętli for-each, aby wypisać wszystkie wartości tablicy ze wszystkimi literami zamienionymi na wielkie. Skorzystaj z funkcjonalności toUpperCase wartości typu String..
4. Napisz program, który pobierze od użytkownika pięć słów i zapisze je w tablicy. Następnie, program powinien wypisać wszystkie słowa, od ostatniego do pierwszego, z literami zapisanymi od końca do początku. Dla przykładu, dla podanych słów "Ala", "ma", "kota", "i", "psa" program powinien wypisać: "asp", "i", "atok", "am", "aLA".
5. Napisz program, który pobierze od użytkownika osiem liczb, zapisze je w tablicy, a następnie posortuje tą tablicę rosnąco i wypisze wynik sortowania na ekran. Dla przykładu, dla liczb 10, -2, 1, 100, 20, -15, 0, 10, program wypisze -15, -2, 0, 1, 10, 10, 20, 100. Zastanów się, jak można posortować ciąg liczb i spróbuj zaimplementować swoje rozwiązanie. Przetestuj je na różnych zestawach danych. Możesz też skorzystać z jednego z popularnych algorytmów sortowania, np. sortowania przez wstawianie. Działanie algorytmu omówiono szczegółowo na zajęciach algorytmy i struktury danych.
6. Napisz program, który pobierze od użytkownika pięć liczb, zapisze je w tablicy, a następnie policzy i wypisze silnię każdej z pobranych liczb.
7. Napisz program, w którym zdefiniujesz dwie tablice przechowujące wartości typu String. Zainicjalizuj obie tablice takimi samymi wartościami, w takiej samej kolejności. Napisz kod, który porówna obie tablice i odpowie na pytanie, czy są one takie same.