

Tablice

Tablice jednowymiarowe

Tablice wielowymiarowe

Tablice jednowymiarowe

- Najprostsza struktura danych w Javie
- Pozwala przechowywać wiele wartości tego samego typu
- Może przechowywać wartości dowolnych typów prostych lub obiektowych
- Pozwala ograniczyć ilość powtarzalnych zmiennych
- Po utworzeniu jest wypełniona wartościami domyślnymi (false, 0, null)

`int[] tablica = new int[5];`

Typ Nazwa Utworzenie tablicy



Tablice jednowymiarowe

- Do elementów tablicy odwołujemy się po indeksach
- Komórki tablicy indeksowane są **od 0**
- Do długości tablicy możemy odwołać się poprzez właściwość **length** – zwraca ona rozmiar tablicy a nie ilość faktycznie przechowywanych elementów

```
int[] tablica = new int[5];  
tablica[0] = 5;  
tablica[1] = 10;  
tablica[2] = 15;  
System.out.println(tablica.length); //5
```

tablica



index	0	1	2	3	4
wartość	5	10	15	0	0

Tablice jednowymiarowe – przykład 1 (demo)

```
import java.util.Arrays;

public class NamesArray {
    public static void main(String[] args) {
        String[] names = new String[3];
        names[0] = "Jan";
        names[1] = "Karol";
        names[2] = "Ania";
        System.out.println(names); //źle [Ljava.lang.String;@15db9742
        System.out.println(names[0]); //Jan
        System.out.println(names[1]); //Karol
        System.out.println(names[2]); //Ania
        //lub
        String arrayString = Arrays.toString(names);
        System.out.println(arrayString);
    }
}
```

Tablice jednowymiarowe – przykład 2 (demo)

```
public class Book {  
    private String title;  
    private String author;  
  
    public Book(String t, String a) {  
        this.title = t;  
        this.author = a;  
    }  
  
    public String getTitle() {  
        return title;  
    }  
  
    public void setTitle(String title) {  
        this.title = title;  
    }  
  
    public String getAuthor() {  
        return author;  
    }  
  
    public void setAuthor(String author) {  
        this.author = author;  
    }  
}
```

```
public class Library {  
    public static void main(String[] args) {  
        Book[] books = new Book[3];  
        books[0] = new Book("Adam Mickiewicz", "Dziady");  
        books[1] = new Book("J.K. Rowling", "Harry Potter");  
        System.out.println(books[0].getAuthor() + " "  
            + books[1].getTitle());  
        System.out.println(books[1].getAuthor() + " "  
            + books[2].getTitle());  
    }  
}
```

Tablice jednowymiarowe – przykład 2 (demo)

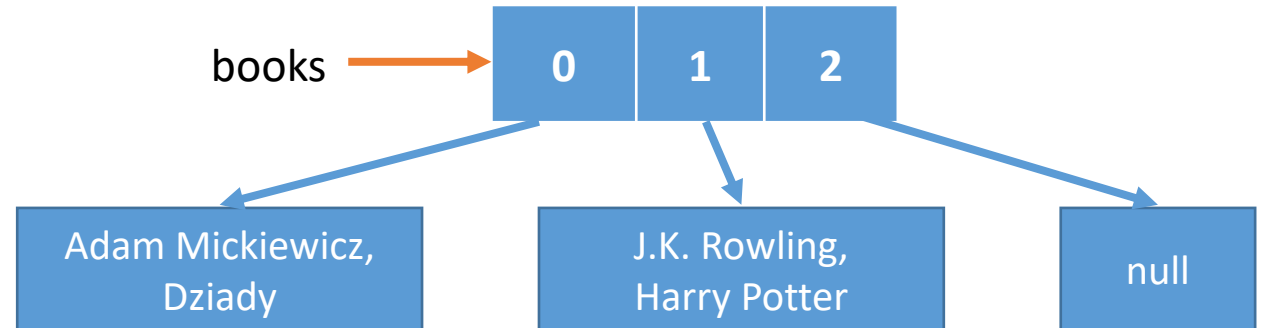
```
public class Book {  
    private String title;  
    private String author;  
  
    public Book(String t, String a) {  
        this.title = t;  
        this.author = a;  
    }  
  
    public String getTitle() {  
        return title;  
    }  
  
    public void setTitle(String title) {  
        this.title = title;  
    }  
  
    public String getAuthor() {  
        return author;  
    }  
  
    public void setAuthor(String author) {  
        this.author = author;  
    }  
}
```

```
public class Library {  
    public static void main(String[] args) {  
        Book[] books = new Book[3];  
        books[0] = new Book("Adam Mickiewicz", "Dziady");  
        books[1] = new Book("J.K. Rowling", "Harry Potter");  
        System.out.println(books[0].getAuthor() + " "  
            + books[1].getTitle());  
        System.out.println(books[1].getAuthor() + " "  
            + books[2].getTitle());  
    }  
}
```

Tablice jednowymiarowe – przykład 2 (demo)

```
public class Book {  
    private String title;  
    private String author;  
  
    public Book(String t, String a) {  
        this.title = t;  
        this.author = a;  
    }  
  
    public String getTitle() {  
        return title;  
    }  
  
    public void setTitle(String title) {  
        this.title = title;  
    }  
  
    public String getAuthor() {  
        return author;  
    }  
  
    public void setAuthor(String author) {  
        this.author = author;  
    }  
}
```

```
public class Library {  
    public static void main(String[] args) {  
        Book[] books = new Book[3];  
        books[0] = new Book("Adam Mickiewicz", "Dziady");  
        books[1] = new Book("J.K. Rowling", "Harry Potter");  
        System.out.println(books[0].getAuthor() + " "  
                             + books[0].getTitle());  
        System.out.println(books[1].getAuthor() + " "  
                             + books[1].getTitle());  
    }  
}
```



Ćwiczenie

Napisz program, w którym wczytasz od użytkownika 3 liczby zmiennoprzecinkowe, a następnie wyświetlisz na ekranie wynik sumowania w postaci takiej jak przedstawiono poniżej. Do przechowywania liczb wykorzystaj tablicę.

```
Podaj pierwszą liczbę
5.5
Podaj drugą liczbę
2.8
Podaj trzecią liczbę
7
5.5 + 2.8 + 7.0 = 15.30
```


Ćwiczenie

Zdefiniuj klasę Employee (pracownik) z kilkoma polami jak imię, nazwisko, wypłata.

W osobnej klasie stwórz tablicę pracowników (minimum 2), wczytaj informacje o pracownikach od użytkownika (minimum 2) i wyświetl ile w sumie pieniędzy firma musi wydać na wypłaty.

Ćwiczenie*

Przerób poprzednie zadanie w taki sposób, aby klasa Company była "kontenerem" na pracowników. Tzn. powinna posiadać prywatne pole będące tablicą obiektów typu Employee. Zdefiniuj metody

- add(Employee emp), która dodaje pracownika na pierwsze wolne miejsce w tablicy,
- get(int index), która zwraca obiekt tablicy zapisany pod indeksem index.

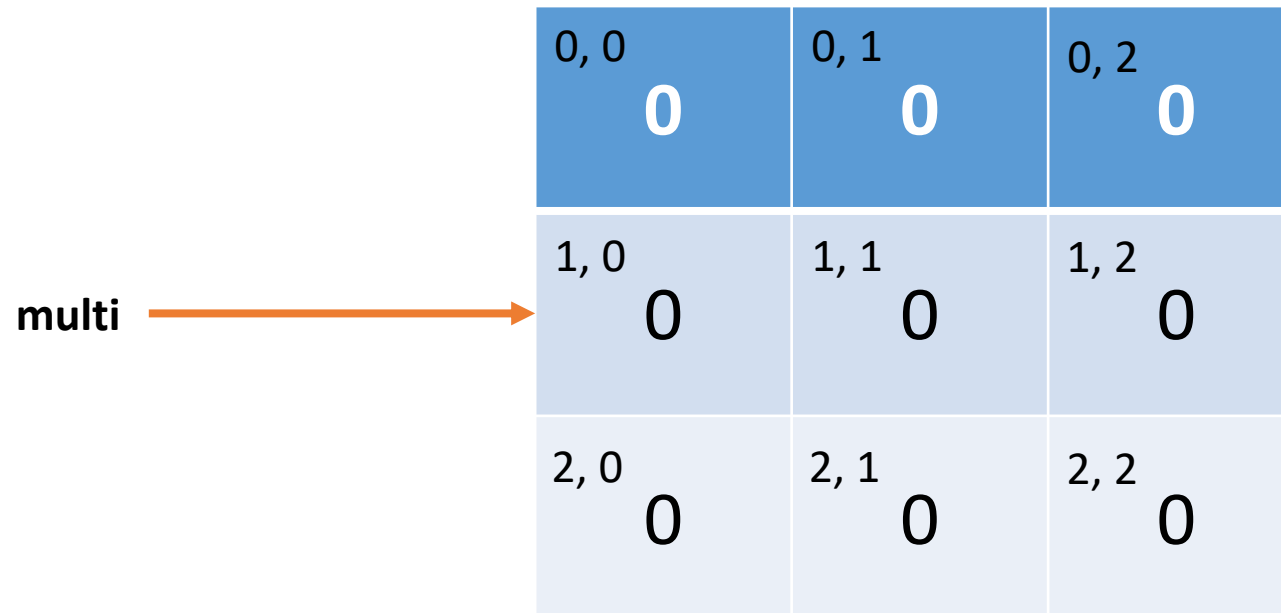
Przetestuj działanie nowej klasy Company w osobnej klasie.

Tablice wielowymiarowe

- Tablice wielowymiarowe to w Javie tablice złożone z innych tablic
- Można je sobie wyobrazić jako wiersze i kolumny

```
int[][] multi = new int[3][3];
```

```
int[][][] multi3 = new int[3][3][3];
```



Tablice wielowymiarowe – przykład

```
public class MultiArray {  
    public static void main(String[] args) {  
        int[][] multi = new int[3][3];  
        multi[0][0] = 1;  
        multi[1][1] = 10;  
        multi[2][2] = 100;  
    }  
}
```

multi



0, 0 1	0, 1 0	0, 2 0
1, 0 0	1, 1 10	1, 2 0
2, 0 0	2, 1 0	2, 2 100

Tablice asymetryczne

```
String[][] asymetric = new String[3][];  
asymetric[0] = new String[3];  
asymetric[1] = new String[2];  
asymetric[2] = new String[1];
```

asymetric



0, 0 null	0, 1 null	0, 2 null
1, 0 null	1, 1 null	
2, 0 null		

Ćwiczenie

Napisz program, w którym utworzysz dwuwymiarową tablicę liczb o wymiarach zgodnych z poniższym obrazem. Uzupełnij wszystkie komórki liczbami różnymi od 0 i wyświetl na ekranie sumę pierwszych z liczb w każdym wierszu.

1	2	3	
4	5		
6	7	8	9
10			