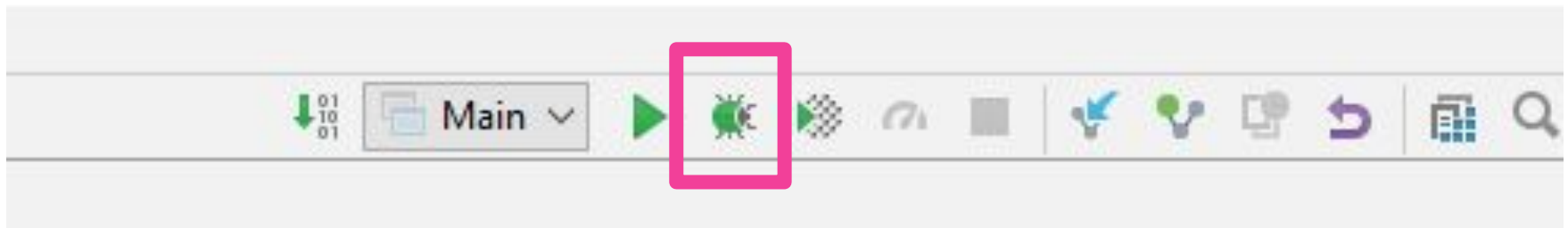




VILNIUSCODINGSCHOOL

# JAVA 4


# DEBUGGING - KLAIDŲ IEŠKOJIMAS



```
2
3 ► public class Main {
4 ►   public static void main(String[] args) {
5     System.out.println("Hello World!");
6     int a = 2;
7 ●   int b = 3 ;
8     int sum = sum(a,b);
9     System.out.println(sum);
10  }
11
12 @   private static int sum(int a, int b) {
13 ●   int result = a + b;
14     return result;
15  }
```

# BREAK POINT

- Programos sustojimo taškas DEBUG režime.



```
int b = 3 ;  
int sum = sum(a,b) ;  
System.out.println(sum) ;
```

```

3  public class Main {
4      public static void main(String[] args) {  args: {}
5          System.out.println("Hello World!");
6          int a = 2;  a: 2
7          int b = 3 ;  b: 3
8          int sum = sum(a,b);  sum: 5  a: 2  b: 3
9          System.out.println(sum);  sum: 5
10     }
11
12     @ private static int sum(int a, int b) {
13         int result = a + b;
14         return result;
15     }

```

Main > main()

Debug: Main x

Debugger Console →

Frames Variables

"main"@1 in group "main": RUNNING

main:9, Main (lesson4)

args = {String[0]@666}

a = 2

b = 3

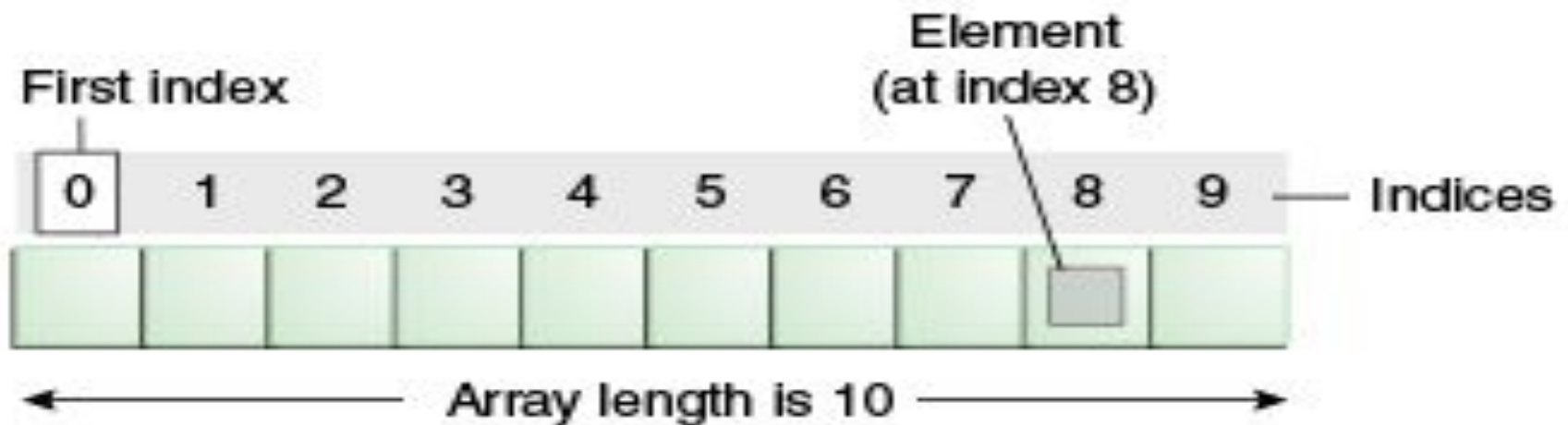
sum = 5

# DEBUGGING - KLAIDŲ IEŠKOJIMAS

- Klaidų ieškojimui
- Stebėti kaip veikia kodas
  - Kintamųjų kitimas
  - Metodų rezultatai
  - Klaidų šaltiniai
- F7 vykdyti eilutę
- F8 eiti į metodą
- [Papildomas šaltinis](#)(Youtube)

# MASYVAI

- Kolekcija/ dėžė
- Fiksuotas dydis
- Bet koks duomenų tipas
- Indeksuojami nuo 0
- Individuali reikšmė pasiekama per indeksą



# MASYVAI

```
//default values  
int arr[] = new int[10];  
int[] arr2 = new int[5];  
//defined values  
int[] arr3 = {1,2};  
int arr4[] = new int[2];  
arr4[0] = 1;  
arr4[1] = 2;
```

# CIKLAI IR MASYVAI

```
int[] arr = new int[3];  
for (int i = 0; i < arr.length; i++) {  
    arr[i] = i;  
}
```



# DVIMAČIAI MASYVAI

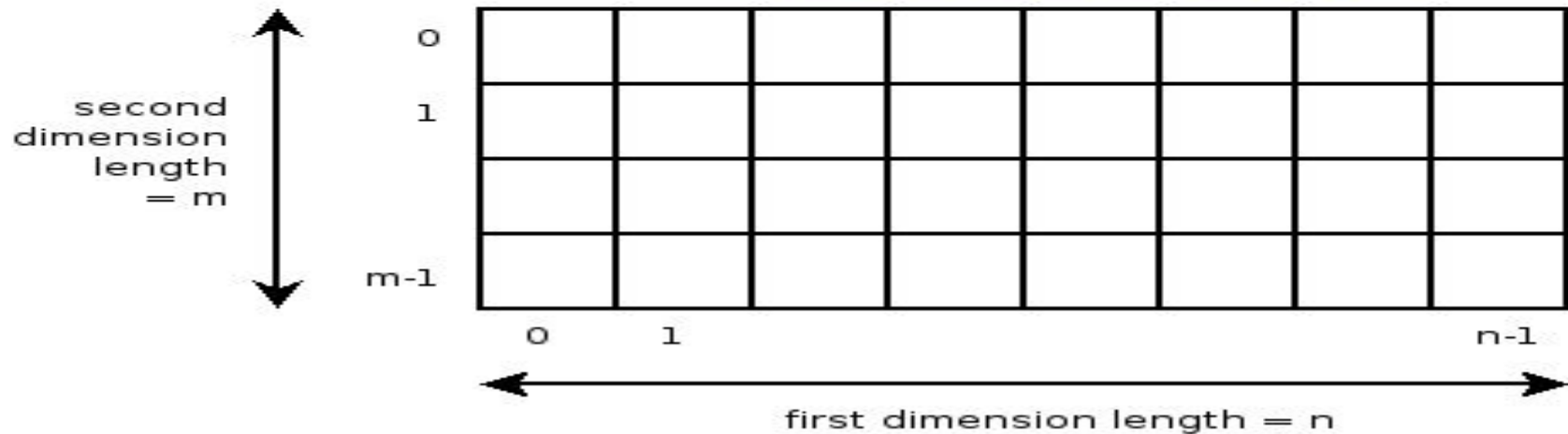
- Kolekcija kolekcijų
- Vienmatis masyvas eilutė - dvimatis lentelė
- Visos savybės kaip įprastų masyvų

```
int[][] m1 = new int[2][2];  
int[][] m2 = { { 4, 5, 6 }, { 1, 2, 3 } };
```

### One-dimensional array



### Two-dimensional array



```
int[][] table = {{1, 3, 4, 5, 6},  
                 {3, 46, 76, 7},  
                 {5, 3}};
```

```
int[] firstRow = table[0]; // result is {1,3,4,5,6}
```

```
int firstRowFirstColumn = table[0][0]; //result is 1
```

## DVIGUBI CIKLAI(ciklas cikle)

```
for (int i = 0; i < 3; i++) {  
    for (int k = 0; k < 4; k++) {  
        System.out.println(i + "-" + k);  
    }  
}
```

# DVIMAČIO MASYVO ATSPAUSDINIMAS

```
int[][] table = {{1, 3, 4, 5, 6},  
                 {3, 46, 76, 7},  
                 {5, 3}};
```

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

Rezultatas:

[I@71e7a66b

[I@2ac1fdc4

[I@5f150435

```
int[][] table = {{1, 3, 4, 5, 6},  
                 {3, 46, 76, 7},  
                 {5, 3}};
```

```
for (int i = 0; i < table.length; i++) {  
    for (int j = 0; j < table[i].length; j++) {  
        System.out.print(table[i][j]);  
    }  
    System.out.println();  
}
```

Rezultatas:

13456

346767

53