

Napredno programiranje i programski jezici

09 Java

Fakultet tehničkih nauka, Novi Sad

23-24/Z

Dunja Vrbaški

```
public static void main(String[] args) {  
    int x = 5;  
    int y = 1;  
    int z = x /y;  
  
    System.out.println("x/y = " + z);  
}
```

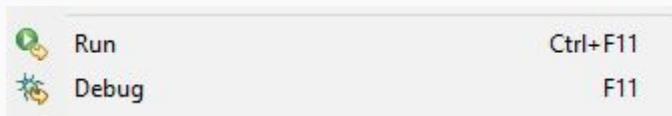
x/y = 5

```
public static void main(String[] args) {  
    int x = 5;  
    int y = 2;  
    int z = x /y;  
  
    System.out.println("x/y = " + z);  
}
```

x/y = 2

```
public static void main(String[] args) {  
    int x = 5;  
    int y = 0;  
    int z = x /y;  
  
    System.out.println("x/y = " + z);  
}
```

```
public static void main(String[] args) {  
    int x = 5;  
    int y = 0;  
    int z = x /y;  
  
    System.out.println("x/y = " + z);  
}
```



```
public static void main(String[] args) {  
    int x = 5;  
    int y = 0;  
    int z = x /y;  
  
    System.out.println("x/y = " + z);  
}
```

ctrl + F11

```
Exception in thread "main" java.lang.ArithmetiсException: / by zero  
at nppj.ModifiersTest.main(ModifiersTest.java:8)
```

```
public static void main(String[] args) {  
    int x = 5;  
    int y = 0;  
  
    try {  
        z = x / y;  
        System.out.println("x/y = " + z);  
    } catch (Exception e) {  
        System.out.println("Deljenje nulom");  
    }  
}
```

```
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
```

```
public static void main(String[] args) {  
    int x = 5;  
    int y = 0;  
  
    try {  
        z = x / y;  
        System.out.println("x/y = " + z);  
    } catch (Exception e) {  
    } catch (ArithmetricException e) {  
        System.out.println("Deljenje nulom");  
    }  
}
```

```
public static void main(String[] args) {
    int x = 5;
    int y = 0;

    try {
        z = x / y;
        System.out.println("x/y = " + z);
    } catch (Exception e) {
        System.out.println("Deljenje nulom");
        System.out.println(e);
    }
}
```

```
public static void main(String[] args) {
    int[] x = new int[]{0, 1, 2, 3, 4};

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

```
public static void main(String[] args) {
    int[] x = new int[]{0, 1, 2, 3, 4};

    for (int i = 0; i < x.length; i++) {
        int z = x[i - 1];
        System.out.println(z);
    }
}
```

```
public static void main(String[] args) {
    int[] x = new int[]{0, 1, 2, 3, 4};

    try {
        for (int i = 0; i < x.length; i++) {
            int z = x[i-1];
            System.out.println(z);
        }
    } catch (Exception e) {
        System.out.println(e);
    }
}
```

```
java.lang.ArrayIndexOutOfBoundsException: Index -1 out of bounds for length 5
```

Realni scenariji - ako nam neko prosleđuje indeks ili granicu za petlju kao parametar metode, ako baratamo indeksima,...

```
public static void main(String[] args) {
    int[] x = new int[]{0, 1, 2, 3, 4};

    try {
        for (int i = 0; i < x.length; i++) {
            int z = x[i + 1] / x[i];
            System.out.println(z);
        }
    } catch (Exception e) {
        System.out.println(e);
    }
}
```

```
java.lang.ArithmeticException: / by zero
```

```
public static void main(String[] args) {
    int[] x = new int[]{0, 1, 2, 3, 4};

    try {
        for (int i = 0; i < x.length; i++) {
            int z = x[i + 1] / x[i];
            System.out.println(z);
        }
    } catch (ArrayIndexOutOfBoundsException e) {
        System.out.println(e);
    } catch (ArithmetricException e) {
        System.out.println(e);
    }
}
```

?

```
public static void main(String[] args) {
    int[] x = new int[]{0, 1, 2, 3, 4};

    try {
        for (int i = 0; i < x.length; i++) {
            int z = x[i + 1] / x[i];
            System.out.println(z);
        }
    } catch (ArrayIndexOutOfBoundsException e) {
        System.out.println(e);
    } catch (ArithmaticException e) {
        System.out.println(e);
    }
}
```

java.lang.ArithmaticException: / by zero

```
public static void main(String[] args) {
    int[] x = new int[]{0, 1, 2, 3, 4};

    try {
        for (int i = 0; i < x.length; i++) {
            int z = x[i + 1] / x[i];
            System.out.println(z);
        }
    } catch (ArrayIndexOutOfBoundsException e) {
        System.out.println(e);
    } catch (ArithmException e) {
        System.out.println(e);
    } catch (Exception e) {
        System.out.println(e);
    }
}
```

```
java.lang.ArithmException: / by zero
```

```
public static void main(String[] args) {
    int[] x = new int[]{0, 1, 2, 3, 4};

    try {
        for (int i = 0; i < x.length; i++) {
            int z = x[i + 1] / x[i];
            System.out.println(z);
        }
    } catch (Exception e) {
        System.out.println(e);
    } catch (ArrayIndexOutOfBoundsException e) {
        System.out.println(e);
    } catch (ArithmaticException e) {
        System.out.println(e);
    } catch (Exception e) {
        System.out.println(e);
    }
}
```

Unreachable catch block for ArithmaticException. It is already handled by the catch block for Exception

```
public static void main(String[] args) {
    int[] x = new int[]{0, 1, 2, 3, 4};

    try {
        for (int i = 0; i < x.length; i++) {
            int z = x[i + 1] / x[i];
            System.out.println(z);
        }
    } catch (ArrayIndexOutOfBoundsException e) {
        System.out.println(e);
    } catch (ArithmetricException e) {
        System.out.println(e);
    }
}
```

try → for

```
public static void main(String[] args) {
    int[] x = new int[]{0, 1, 2, 3, 4};

    for (int i = 0; i < x.length; i++) {
        try {
            int z = x[i + 1] / x[i];
            System.out.println(z);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println(e);
        } catch (ArithmetricException e) {
            System.out.println(e);
        }
    }
}
```

?

```
public static void main(String[] args) {
    int[] x = new int[]{0, 1, 2, 3, 4};

    for (int i = 0; i < x.length; i++) {
        try {
            int z = x[i + 1] / x[i];
            System.out.println(z);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println(e);
        } catch (ArithmeticException e) {
            System.out.println(e);
        }
    }
}
```

?

```
java.lang.ArithmeticException: / by zero
2
1
1
java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 5
```

```
public static void main(String[] args) {  
    int x = 5;  
    int y = 0;  
    int z = x /y;  
  
    System.out.println("x/y = " + z);  
}
```

F11

The screenshot shows a Java development environment with the following details:

- Code Editor:** The file `TestApp.java` is open. The code defines a class `TestApp` with a `main` method. Line 8 contains the assignment `int z = x / y;`, which is highlighted with a green bar.
- Debugger Variables View:** The `Variables` tab is selected in the debugger. A red circle highlights the entry for variable `z`, which has a value of 0. Other variables listed are `args` (String[0]), `x` (5), and `y` (0).

```
1 package nppj;
2
3 public class TestApp {
4
5     public static void main(String[] args) {
6         int x = 5;
7         int y = 0;
8         int z = x / y; // Line 8 is highlighted with a green bar
9
10        System.out.println("x/y = " + z);
11    }
12
13
14
15
16
17
18
19
20
21
22
23
24
25
```

```
public static void main(String[] args) {
    int x = 3;
    int y = 5;

    int z = Saberi(x, y);
    System.out.println(z);

    int[] niz = new int[] {0, 1, 2, 3, 4};

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

private static int saberi(int x, int y) {
    return x + y;
}
```

A screenshot of a Java IDE interface. The title bar says "TestApp.java X". Below it is a code editor window containing the following Java code:

```
1 package nppj;
2
3 public class TestApp {
4
5     public static void main(String[] args) {
6         System.out.println("Hello World!");
7     }
8 }
```

A context menu is open over the line "public static void main(String[] args)". The menu has a blue background and contains the following items:

- Toggle Breakpoint (Ctrl+Shift+B)
- Disable Breakpoint (Shift+Double Click)
- Toggle Tracepoint
- Run to Line (Ctrl+Alt+Click)
- Go to Annotation (Ctrl+1)
- Validate
- Add Bookmark...
- Add Task...

ili dupli klik

The screenshot shows an IDE interface with the following details:

- Editor Tab:** TestApp.java X
- Code Content:**

```
1 package nppj;
2
3 public class TestApp {
4
5     public static void main(String[] args) {
6         int x = 3;
7         int y = 5;
8
9         int z = Saberi(x, y);
10        System.out.println(z);
11
12        int[] niz = new int[] {0, 1, 2, 3, 4};
13
14        int sum = 0;
15        for (int i = 0; i < niz.length; i++) {
16            sum += niz[i];
17        }
18        System.out.println(sum);
19    }
20}
```
- Variables Window:** Shows the current state of variables.

Name	Value
↳ no method return value	
args	String[0] (id=20)

- Bottom Status Bar:** Shows the message "<Choose a previously entered expression>"

F11 / F6 / F5

The screenshot shows an IDE interface with two main panes. On the left is the code editor for `TestApp.java`, displaying the following Java code:

```
1 package nppj;
2
3 public class TestApp {
4
5     public static void main(String[] args) {
6         int x = 3;
7         int y = 5;
8
9         int z = Saberi(x, y);
10        System.out.println(z);
11
12        int[] niz = new int[] {0, 1, 2, 3, 4};
13
14        int sum = 0;
15        for (int i = 0; i < niz.length; i++) {
16            sum += niz[i];
17        }
18        System.out.println(sum);
19    }
20}
```

The line `15 | for (int i = 0; i < niz.length; i++) {` is highlighted with a green background. On the right is the debugger's variables view, titled '(x) Variables'. It lists the following variables and their values:

Name	Value
args	String[0] (id=20)
x	3
y	5
z	8
niz	(id=28)
sum	6
i	3

The variable `sum` is highlighted with a yellow background.

The screenshot shows an IDE interface with two main panes. The left pane displays the source code of a Java application named TestApp.java. The right pane is a debugger's variables window.

TestApp.java:

```
1 package nppj;
2
3 public class TestApp {
4
5     public static void main(String[] args) {
6         int x = 3;
7         int y = 5;
8
9         int z = Saberi(x, y);
10        System.out.println(z);
11
12        int[] niz = new int[] {0, 1, 2, 3, 4};
13
14        int sum = 0;
15        for (int i = 0; i < niz.length; i++) {
16            sum += niz[i];
17        }
18        System.out.println(sum);
19    }
20}
```

Variables Window:

Name	Value
sum+1	7
x/y	0
niz.length	5

No details to display for the current selection.

ZADATAK

(neobavezno)

Isprobati sve iz prezentacije.

Pogledati dokumentaciju za String.

Proveriti šta se dešava u C++ ako probamo deljenje sa 0.

Pokušati debug.

Pogledati šta je Singleton design pattern i zašto bismo poželeli da ga koristimo.