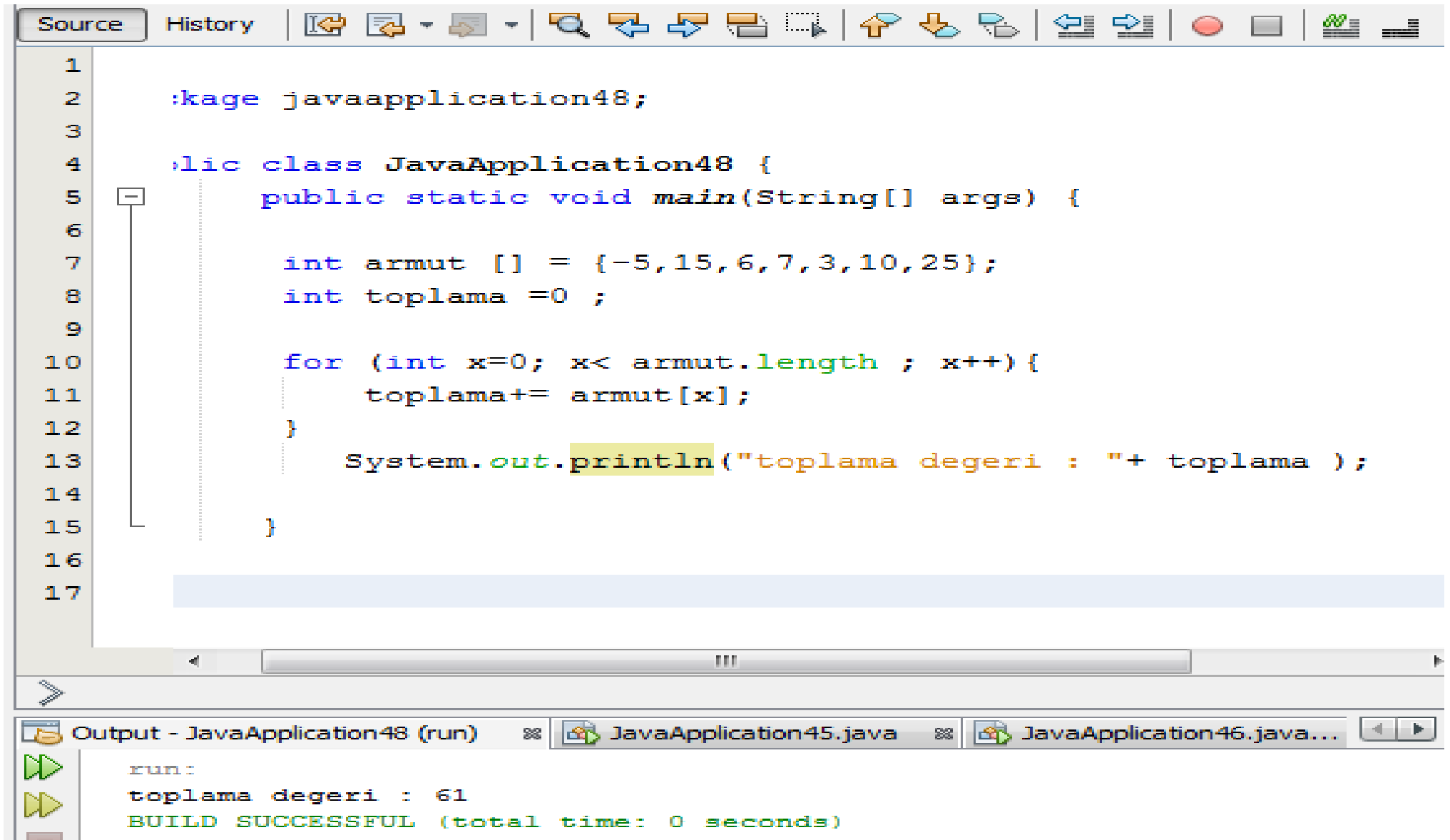


Array 4. yol:



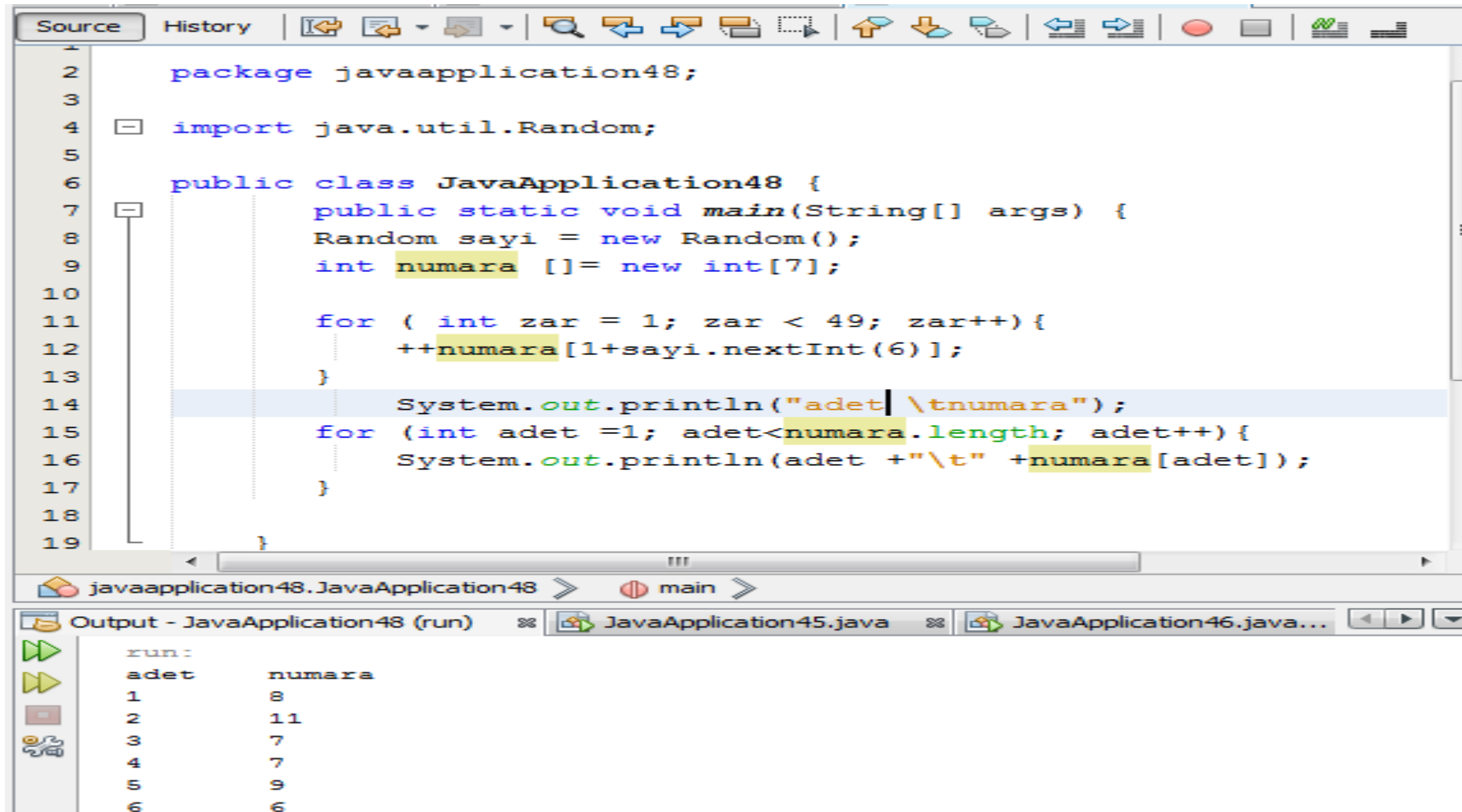
The image shows a screenshot of an IDE with a Java source file and its output. The source code defines a class `JavaApplication48` with a `main` method that calculates the sum of an array `armut` containing the values `{-5, 15, 6, 7, 3, 10, 25}`. The output window shows the result of the program execution.

```
1
2 :kage javaapplication48;
3
4 public class JavaApplication48 {
5     public static void main(String[] args) {
6
7         int armut [] = {-5,15,6,7,3,10,25};
8         int toplama =0 ;
9
10        for (int x=0; x< armut.length ; x++){
11            toplama+= armut[x];
12        }
13        System.out.println("toplama degeri : "+ toplama );
14
15    }
16
17
```

Output - JavaApplication48 (run)

```
run:
toplama degeri : 61
BUILD SUCCESSFUL (total time: 0 seconds)
```

Array 5. Örnek:



The screenshot shows an IDE window with a Java source file. The code defines a class `JavaApplication48` with a `main` method. Inside `main`, a `Random` object is created, and an array `numara` of size 7 is initialized. A loop from `zar = 1` to `48` fills the array with random integers between 1 and 6. Finally, the array is printed with column headers `adet` and `numara`.

```
1 package javaapplication48;
2
3
4 import java.util.Random;
5
6 public class JavaApplication48 {
7     public static void main(String[] args) {
8         Random sayi = new Random();
9         int numara []= new int[7];
10
11         for ( int zar = 1; zar < 49; zar++){
12             ++numara[1+sayi.nextInt(6)];
13         }
14         System.out.println("adet \tnumara");
15         for (int adet =1; adet<numara.length; adet++){
16             System.out.println(adet +"\t" +numara[adet]);
17         }
18     }
19 }
```

The output window shows the following results:

run:	adet	numara
	1	8
	2	11
	3	7
	4	7
	5	9
	6	6

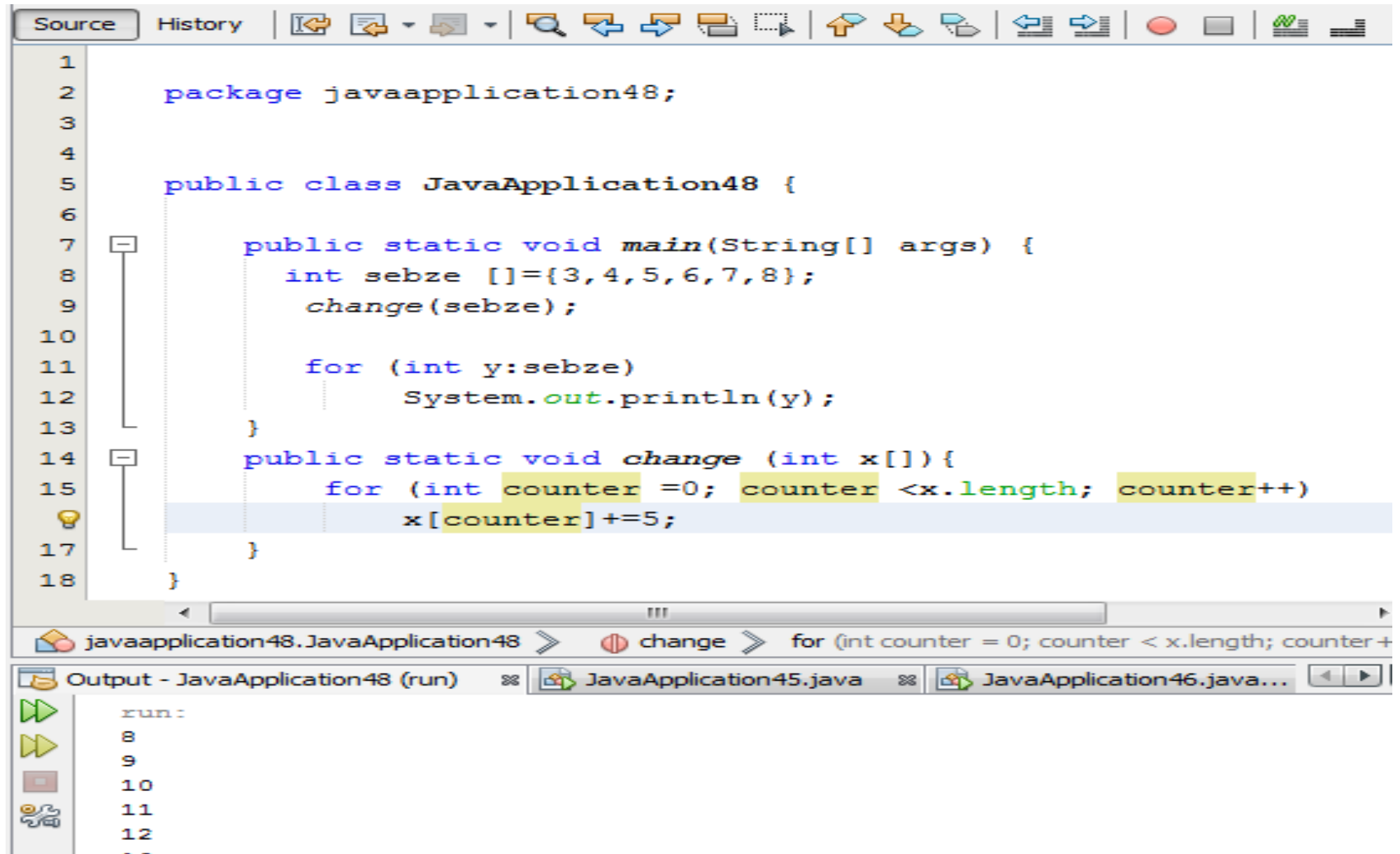
Array 6:

```
1
2 package javaapplication48;
3
4
5 public class JavaApplication48 {
6
7     public static void main(String[] args) {
8
9         int meyve []= {3,4,5,6,7};
10        int toplam = 0;
11
12        for (int x: meyve){
13            toplam += x;
14        }
15        System.out.println(toplam);
16    }
17 }
18
```

Output - JavaApplication48 (run) JavaApplication45.java JavaApplication46.java...

```
run:
25
BUILD SUCCESSFUL (total time: 0 seconds)
```

Array 7: (metot)



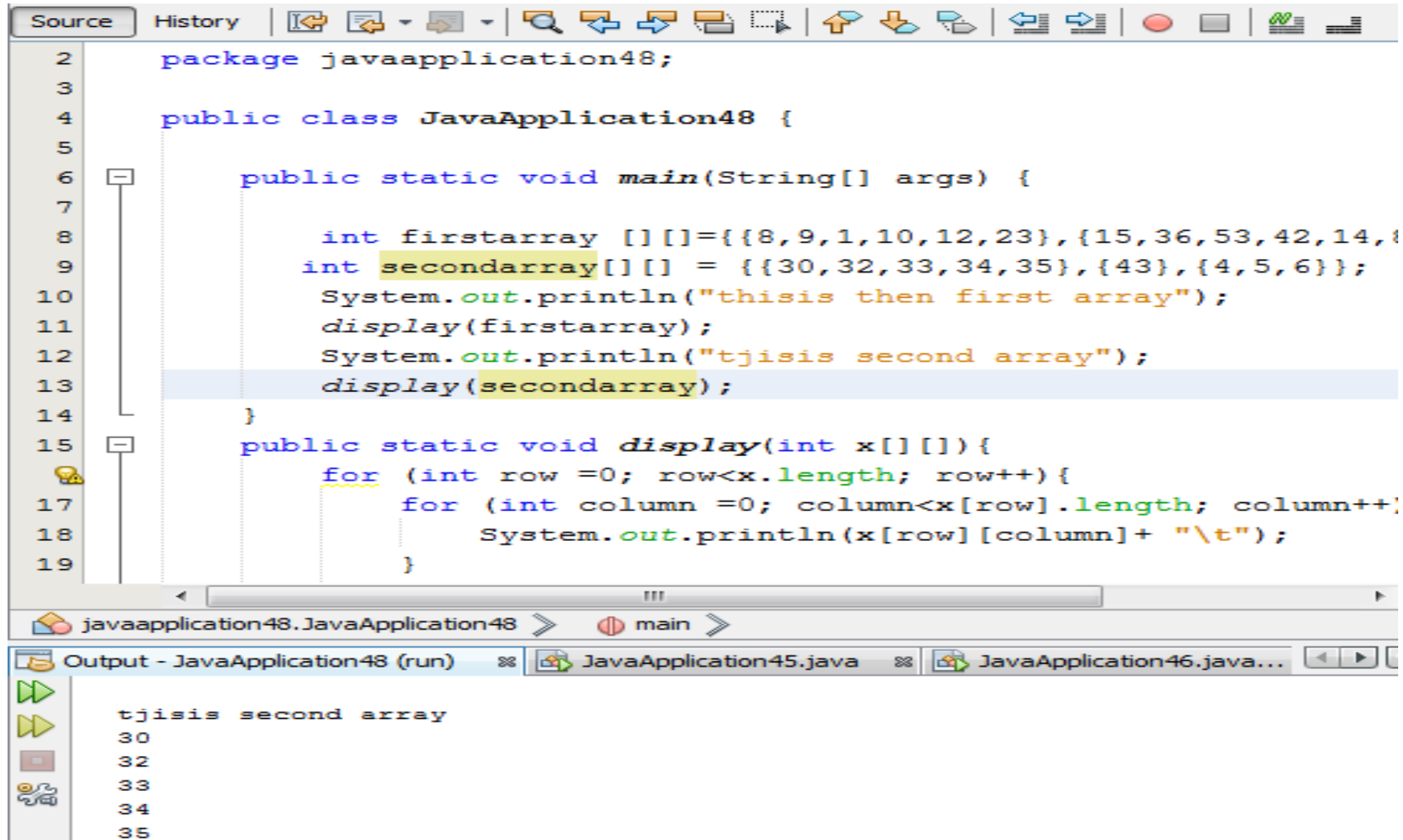
The screenshot shows an IDE window with the following Java code:

```
1  
2 package javaapplication48;  
3  
4  
5 public class JavaApplication48 {  
6  
7     public static void main(String[] args) {  
8         int sebze []={3,4,5,6,7,8};  
9         change(sebze);  
10  
11         for (int y:sebze)  
12             System.out.println(y);  
13     }  
14     public static void change (int x[]){  
15         for (int counter =0; counter <x.length; counter++)  
16             x[counter] +=5;  
17     }  
18 }
```

The IDE interface includes a toolbar at the top with icons for source, history, and various editing actions. The bottom of the window shows a breadcrumb trail: `javaapplication48.JavaApplication48 > change > for (int counter = 0; counter < x.length; counter +`. Below the code editor, there are tabs for `Output - JavaApplication48 (run)`, `JavaApplication45.java`, and `JavaApplication46.java...`. The output window displays the following text:

```
run:  
8  
9  
10  
11  
12  
..
```

Array 8:



```
2 package javaapplication48;
3
4 public class JavaApplication48 {
5
6     public static void main(String[] args) {
7
8         int firstarray [][]={{8,9,1,10,12,23},{15,36,53,42,14,8}};
9         int secondarray [][] = {{30,32,33,34,35},{43},{4,5,6}};
10        System.out.println("thisis then first array");
11        display(firstarray);
12        System.out.println("tjisis second array");
13        display(secondarray);
14    }
15    public static void display(int x[][]){
16        for (int row =0; row<x.length; row++){
17            for (int column =0; column<x[row].length; column++){
18                System.out.println(x[row][column]+ "\t");
19            }
20        }
21    }
22 }
```

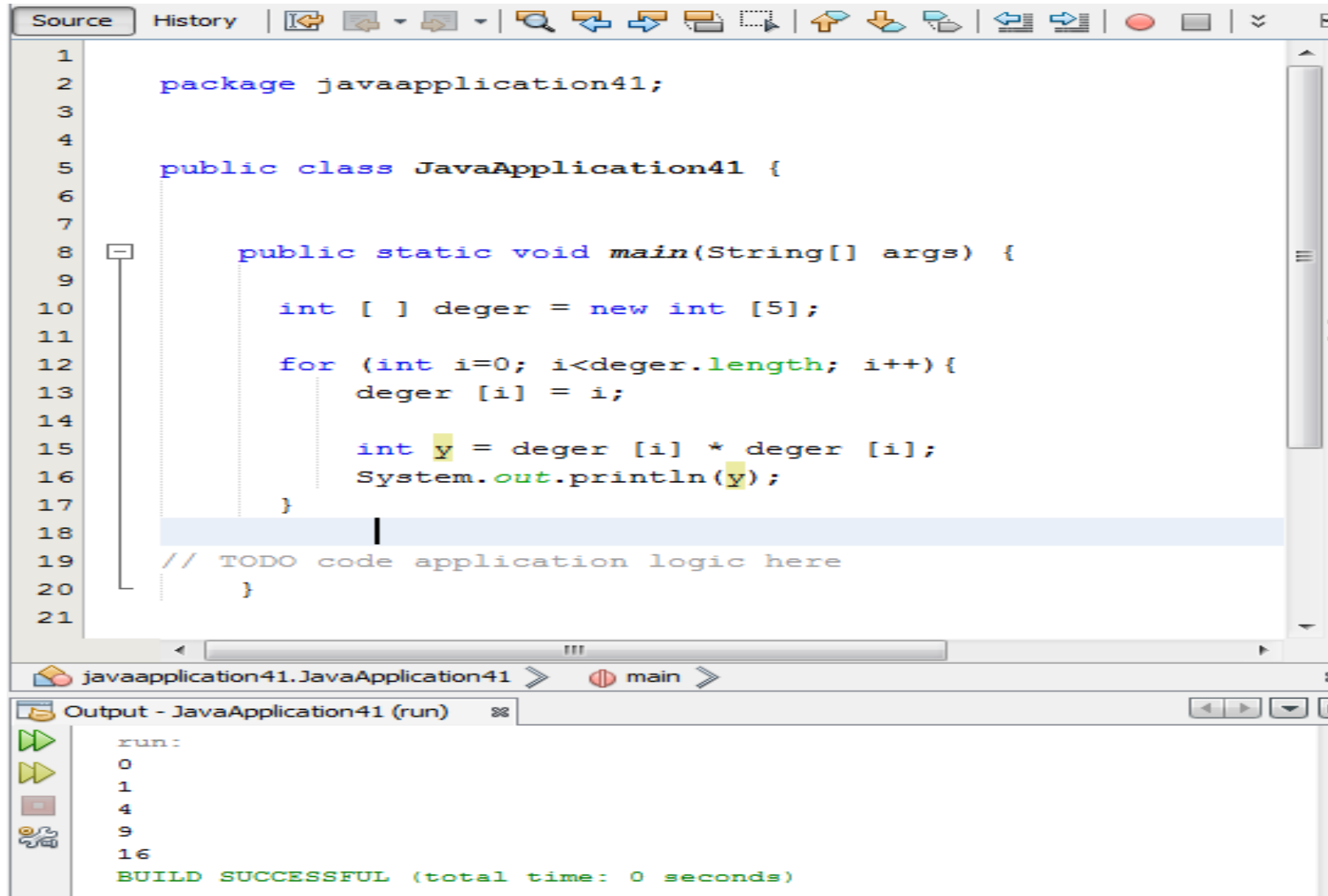
Output - JavaApplication48 (run)

```
tjisis second array
30
32
33
34
35
```

String arrays:

```
Public static void main (String [ ] arguments) {  
    System.out.println (arguments.length);  
    System.out.println (arguments[0]);  
    System.out.println (arguments [1]);  
}
```

FOR VE ARRAY ÖRNEĞİ:

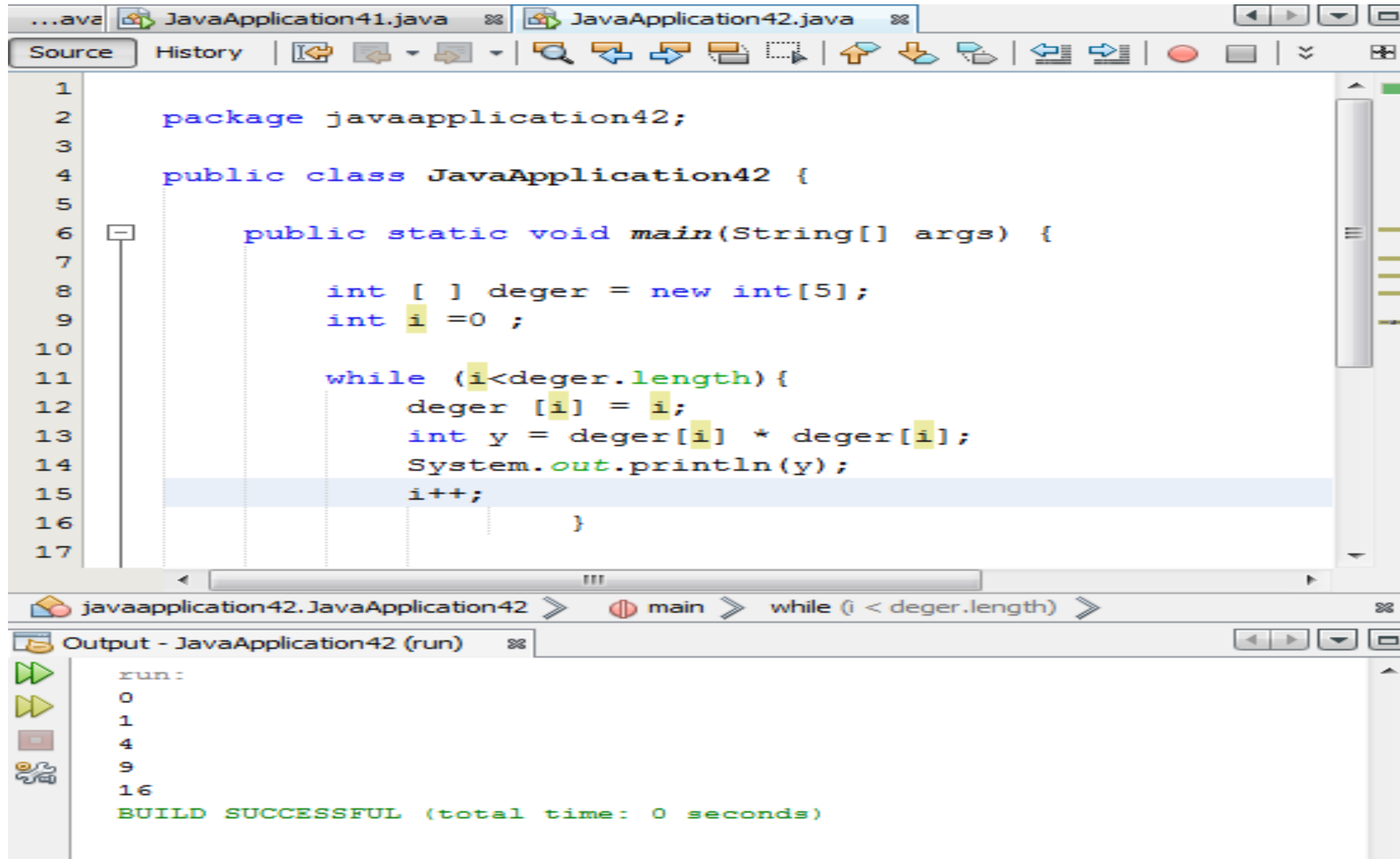


```
1
2 package javaapplication41;
3
4
5 public class JavaApplication41 {
6
7
8     public static void main(String[] args) {
9
10        int [ ] deger = new int [5];
11
12        for (int i=0; i<deger.length; i++){
13            deger [i] = i;
14
15            int y = deger [i] * deger [i];
16            System.out.println(y);
17        }
18
19        // TODO code application logic here
20    }
21
```

Output - JavaApplication41 (run)

```
run:
0
1
4
9
16
BUILD SUCCESSFUL (total time: 0 seconds)
```

While – array örnek:



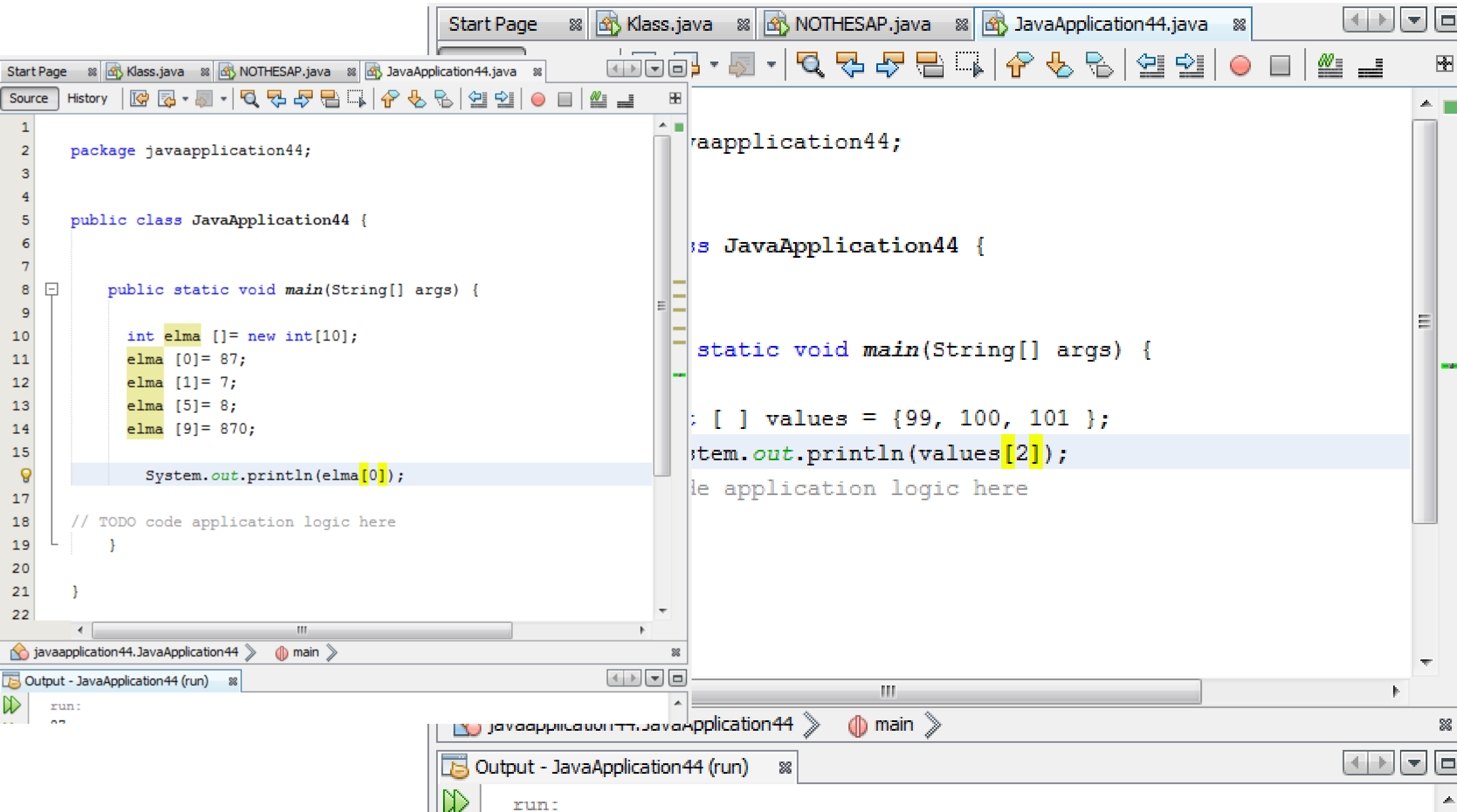
The screenshot shows an IDE window with two tabs: 'JavaApplication41.java' and 'JavaApplication42.java'. The 'JavaApplication42.java' tab is active, displaying the following code:

```
1 package javaapplication42;
2
3
4 public class JavaApplication42 {
5
6     public static void main(String[] args) {
7
8         int [ ] deger = new int[5];
9         int i = 0 ;
10
11         while (i < deger.length) {
12             deger [i] = i;
13             int y = deger[i] * deger[i];
14             System.out.println(y);
15             i++;
16         }
17
```

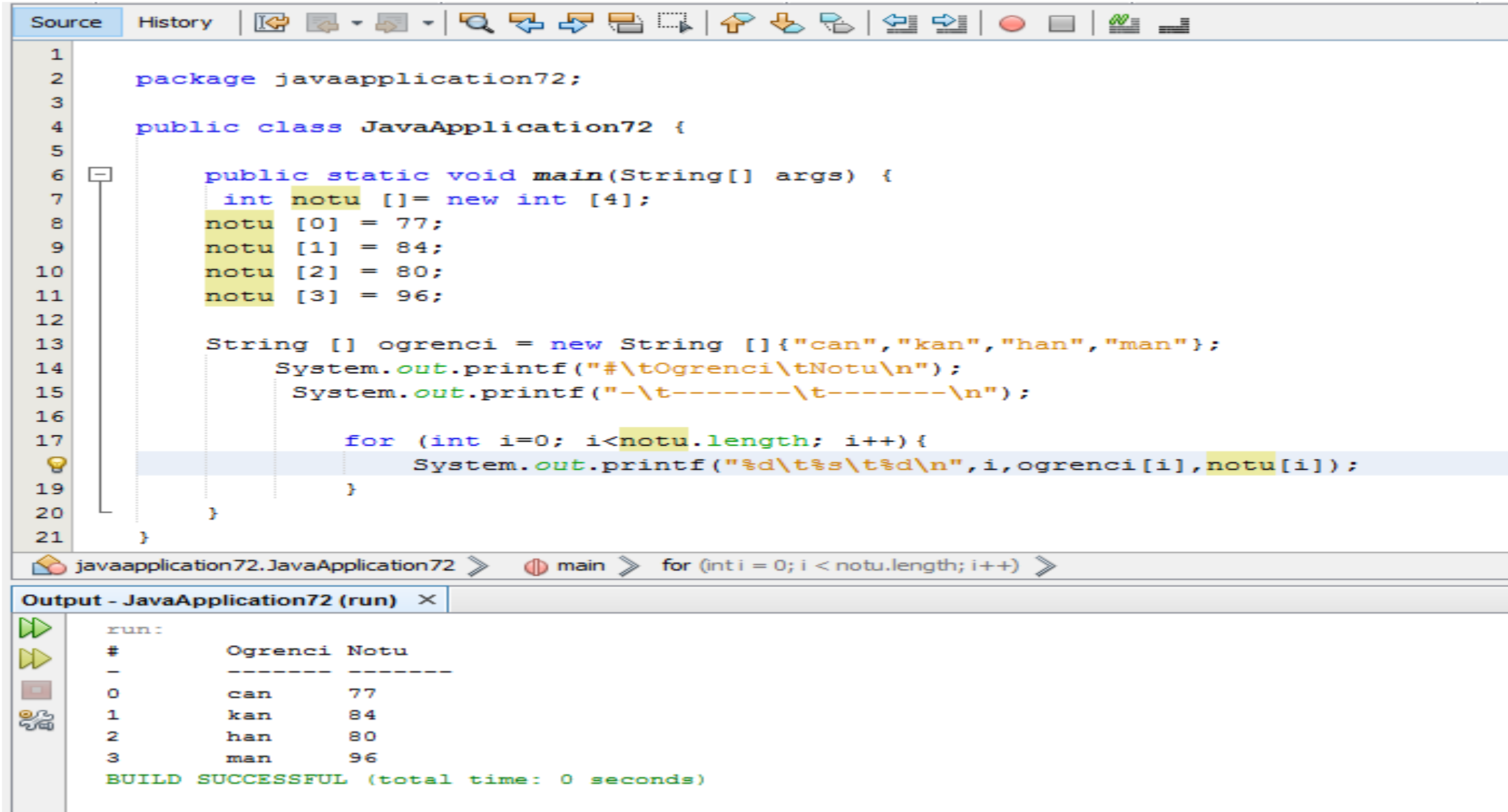
The code is executed, and the output window shows the following results:

```
run:
0
1
4
9
16
BUILD SUCCESSFUL (total time: 0 seconds)
```

The IDE interface includes a toolbar with various icons for navigation and editing, and a breadcrumb trail at the bottom of the code editor showing the current location: 'javaapplication42.JavaApplication42 > main > while (i < deger.length)'.



Array ve String ve For :



The image shows a screenshot of an IDE with a Java source file and its execution output. The source code defines a class `JavaApplication72` with a `main` method. It creates an integer array `notu` with values [77, 84, 80, 96] and a string array `ogrenci` with values ["can", "kan", "han", "man"]. A `for` loop iterates over the arrays, printing the index, student name, and score. The output window shows the results of this loop and a successful build message.

```
1 package javaapplication72;
2
3
4 public class JavaApplication72 {
5
6     public static void main(String[] args) {
7         int notu []= new int [4];
8         notu [0] = 77;
9         notu [1] = 84;
10        notu [2] = 80;
11        notu [3] = 96;
12
13        String [] ogrenci = new String []{"can","kan","han","man"};
14        System.out.printf("#\tOgrenci\tNotu\n");
15        System.out.printf("-\t-----\t-----\n");
16
17        for (int i=0; i<notu.length; i++){
18            System.out.printf("%d\t%s\t%d\n",i,ogrenci[i],notu[i]);
19        }
20    }
21 }
```

Output - JavaApplication72 (run) x


```
run:
#      Ogrenci Notu
-      -
0      can      77
1      kan      84
2      han      80
3      man      96
BUILD SUCCESSFUL (total time: 0 seconds)
```




```
...ava JavaApplication55.java JavaApplication57.java
Source History
1
2 package javaapplication57;
3
4
5 public class JavaApplication57 {
6
7
8     public static void main(String[] args) {
9         int []deger = new int[5];
10
11         for (int i=0; i<deger.length; i++){
12             deger [i]= i;
13             int y = deger [i]* deger [i];
14             System.out.println(y);
15         }
16
17         // TODO code application logic here
18     }
19
20 }
21
```






Output - JavaApplication57 (run)

```
run:
0
1
4
9
16
```

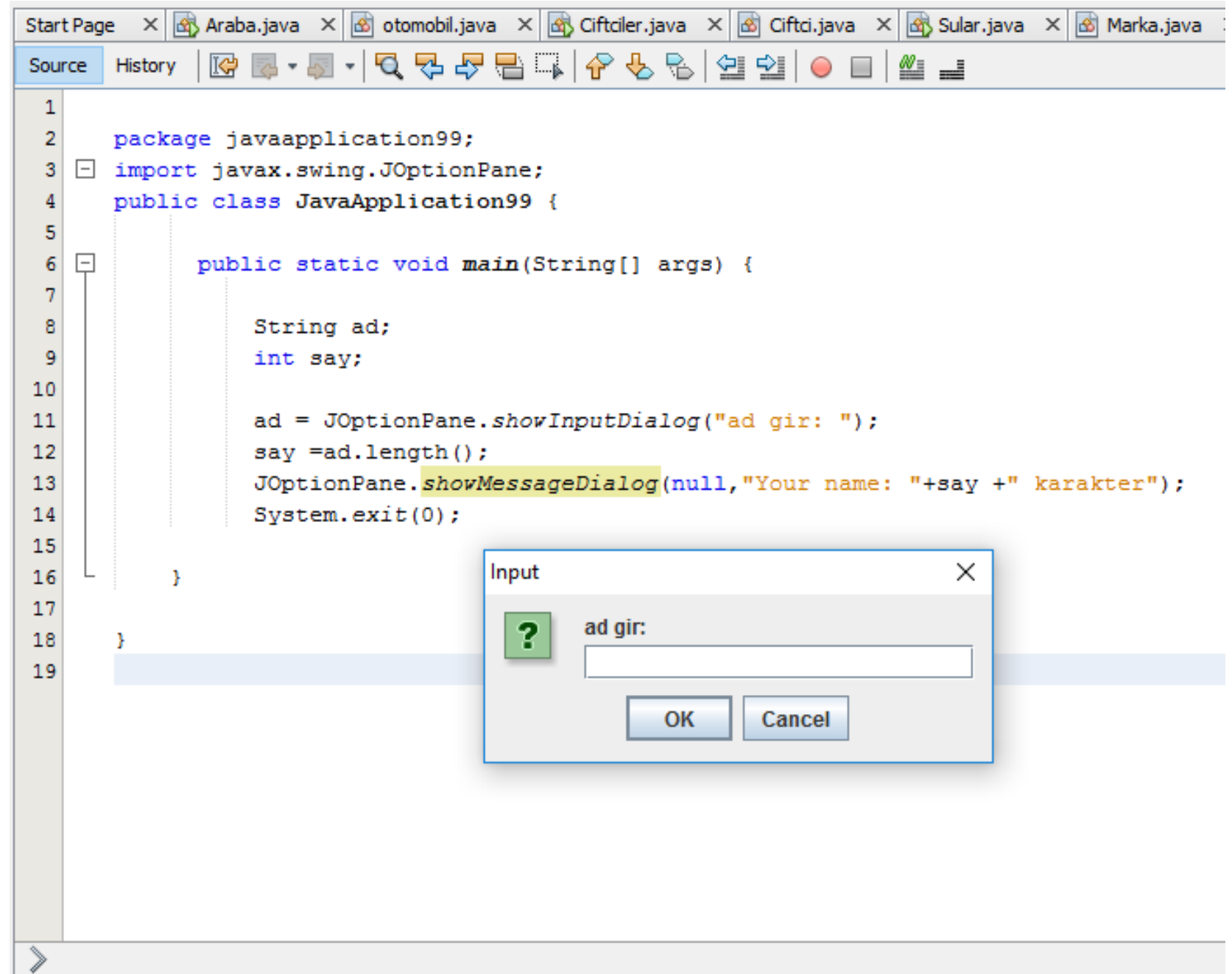
Source History 

```
1
2 package javaapplication57;
3
4
5 public class JavaApplication57 {
6
7
8     public static void main(String[] args) {
9         int [] sayi = new int [5];
10        int i=0;
11        while (i<sayi.length){
12            sayi [i]=i;
13            int y = sayi [i] * sayi [i];
14            System.out.println(y);
15            i++;
16        }
17        // TODO code application logic here
18    }
19
20 }
21
```

Output - JavaApplication57 (run) 

 run:
 0
 1
 4
 9
16

MESAJ KUTUSU UYGULAMASI



The image shows a screenshot of an IDE with several Java files open in the background: Start Page, Araba.java, otomobil.java, Ciftcler.java, Ciftci.java, Sular.java, and Marka.java. The main editor window displays the source code for a Java application named JavaApplication99. The code is as follows:

```
1 package javaapplication99;
2
3 import javax.swing.JOptionPane;
4 public class JavaApplication99 {
5
6     public static void main(String[] args) {
7
8         String ad;
9         int say;
10
11         ad = JOptionPane.showInputDialog("ad gir: ");
12         say =ad.length();
13         JOptionPane.showMessageDialog(null,"Your name: "+say +" karakter");
14         System.exit(0);
15     }
16 }
17
18
19 }
```

An "Input" dialog box is displayed in the foreground, titled "Input" with a close button (X). It contains a question mark icon, the text "ad gir:", an empty text input field, and "OK" and "Cancel" buttons.