

STRING

CLASS

IN THE NAME OF ALLAH

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

THE GRACIOUS, THE MERCIFUL.



GETS - GETTING EDUCATION WITH TECHNOLOGICAL SYSTEM

Instructor: Sir A.Rehman Ali Brohi

LECTURE: 10

String Class



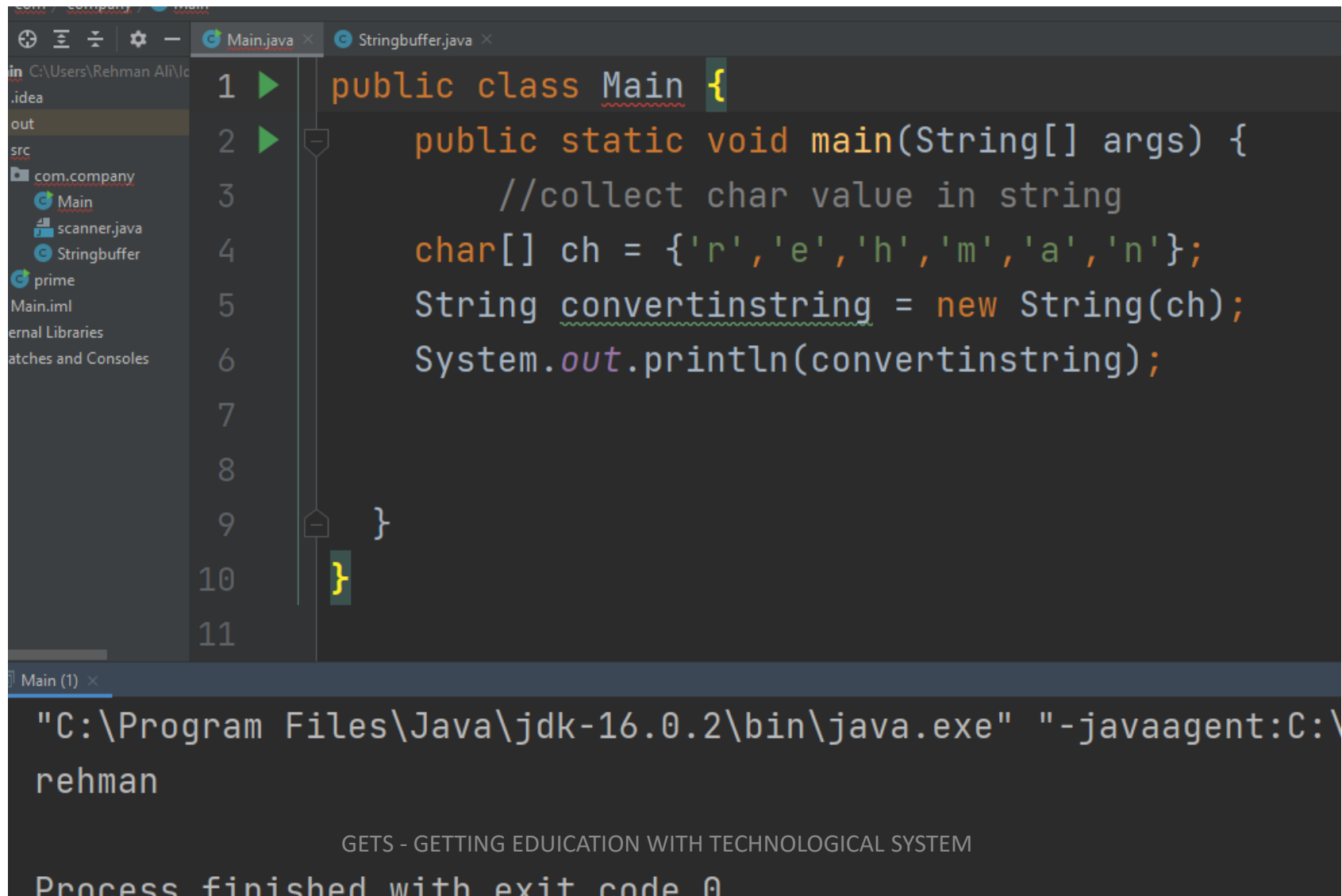
JAVA STRING

- What is String
- Immutable String
- String Comparison
- String Concatenation
- Substring
- Methods of String class
- StringBuffer class
- StringBuilder class
- String vs StringBuffer
- StringBuffer vs Builder
- toString method

WHAT IS STRING?

- In Java, String is an object that represents sequence of char values.
- An array of characters works same as Java String

If want to convert char into String



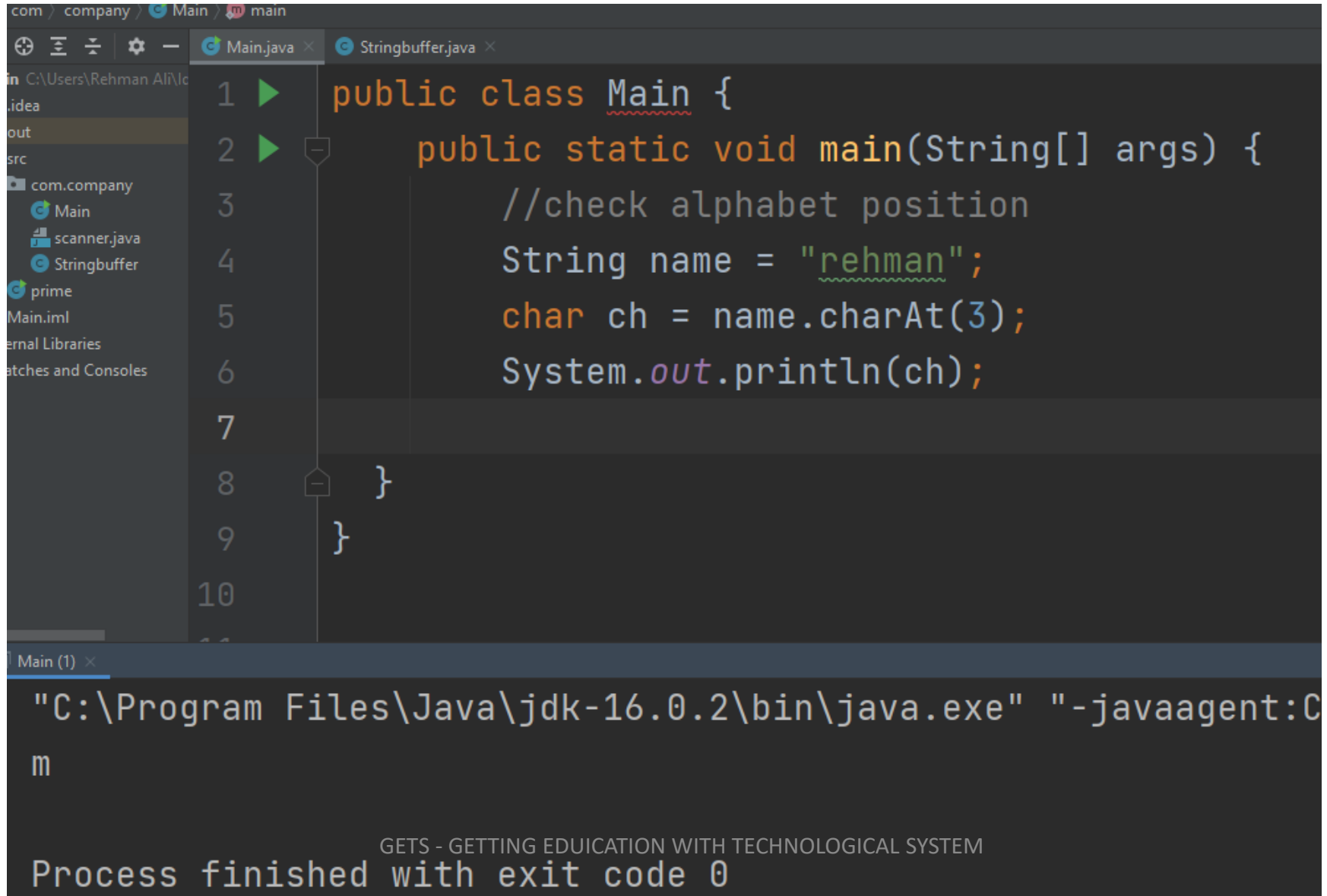
The screenshot shows an IDE with two tabs: `Main.java` and `Stringbuffer.java`. The `Main.java` tab is active, displaying the following code:

```
1 public class Main {  
2     public static void main(String[] args) {  
3         //collect char value in string  
4         char[] ch = {'r','e','h','m','a','n'};  
5         String convertinstring = new String(ch);  
6         System.out.println(convertinstring);  
7  
8  
9     }  
10 }  
11
```

The console output at the bottom shows the command executed and the result:

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\  
rehman  
GETS - GETTING EDUCATION WITH TECHNOLOGICAL SYSTEM  
Process finished with exit code 0
```

If want to check the position of character



The screenshot shows an IDE with two tabs: `Main.java` and `Stringbuffer.java`. The `Main.java` tab is active, displaying the following code:

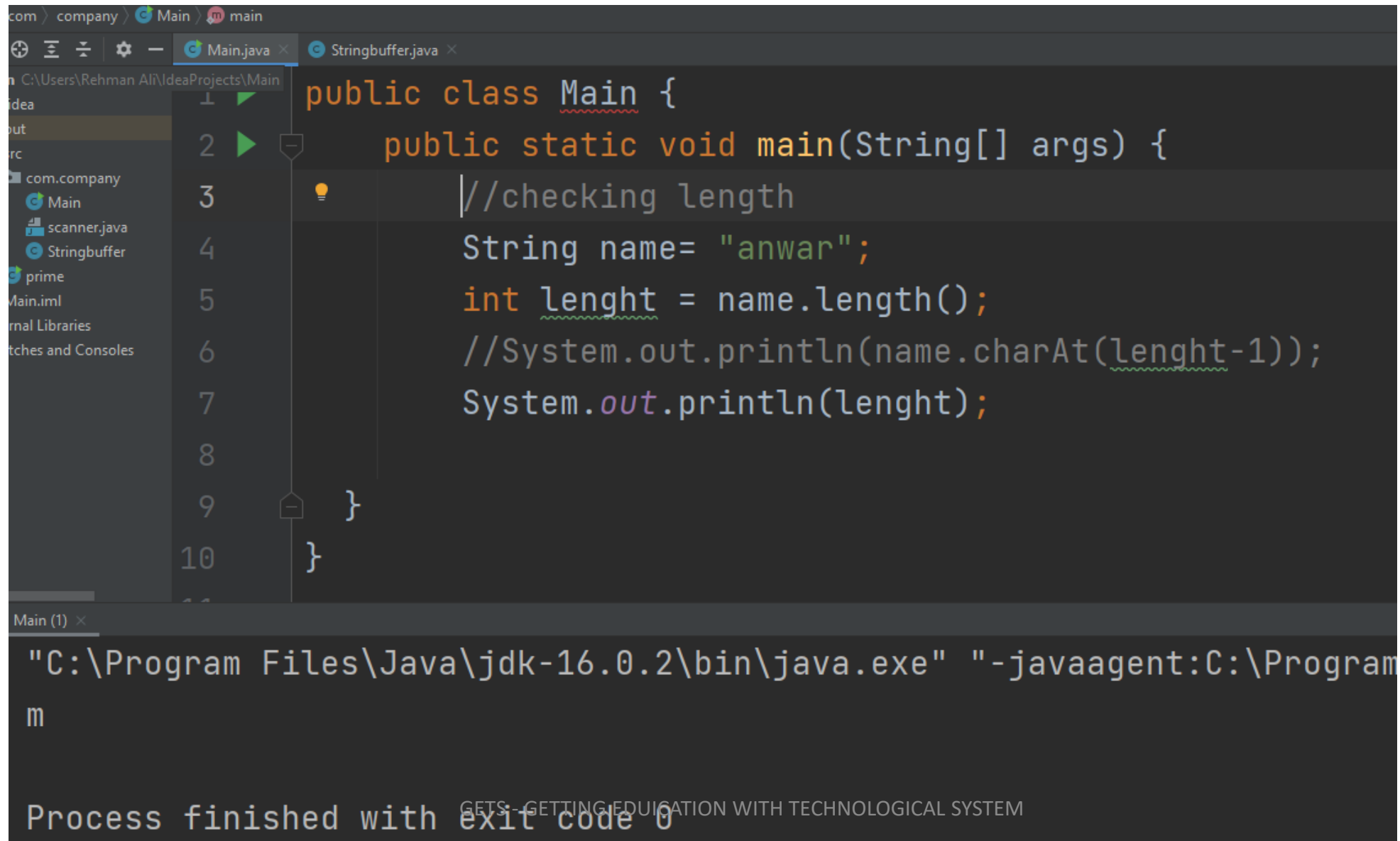
```
1 public class Main {  
2     public static void main(String[] args) {  
3         //check alphabet position  
4         String name = "rehman";  
5         char ch = name.charAt(3);  
6         System.out.println(ch);  
7     }  
8 }  
9  
10
```

The output console at the bottom shows the command used to run the program:

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\Java\jdk-16.0.2\bin\javaagent.jar" -m
```

Below the command, the text "GETS - GETTING EDUCATION WITH TECHNOLOGICAL SYSTEM" is displayed, followed by the message "Process finished with exit code 0".

If want to check length of character or print the last character



The screenshot shows an IDE with a project named 'Main'. The 'Main.java' file is open, displaying the following code:

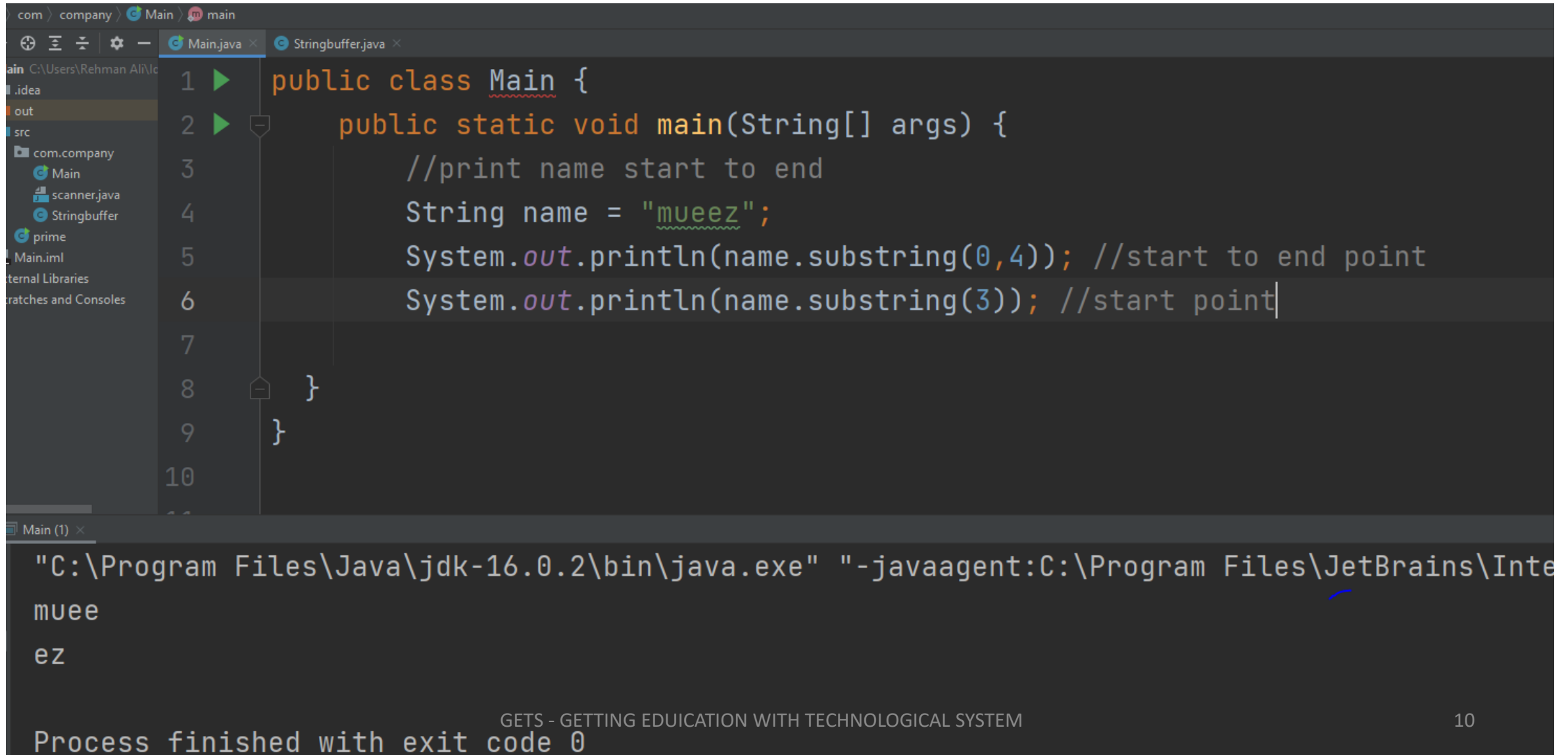
```
1 public class Main {  
2     public static void main(String[] args) {  
3         //checking length  
4         String name= "anwar";  
5         int lenght = name.length();  
6         //System.out.println(name.charAt(lenght-1));  
7         System.out.println(lenght);  
8     }  
9 }  
10
```

The console output at the bottom shows the command used to run the program:

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program  
m  
Process finished with exit code 0
```

GETS - GETTING EDUCATION WITH TECHNOLOGICAL SYSTEM

If want to display the First to end character



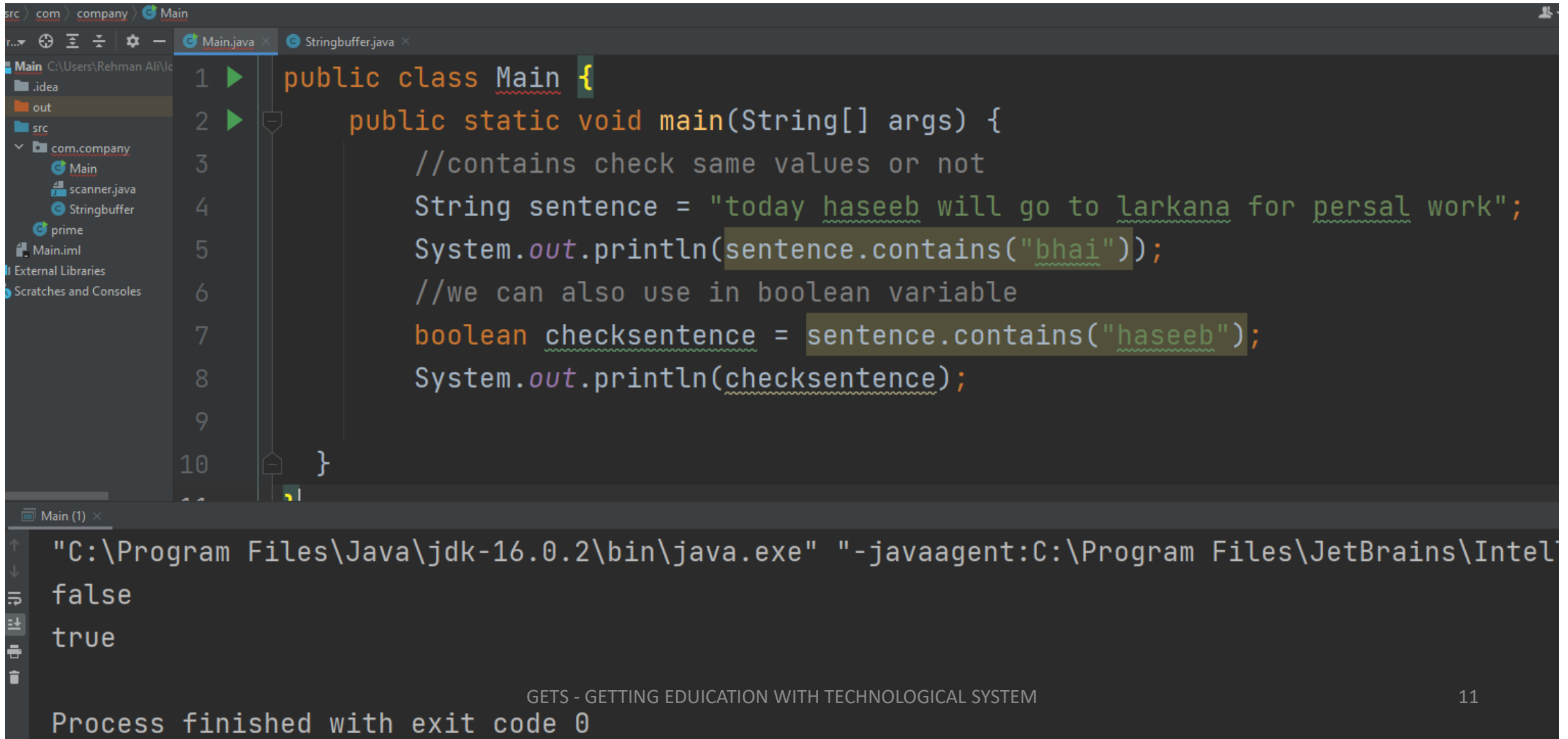
The screenshot shows an IDE with a project named 'com' containing a package 'company' and a class 'Main'. The code in 'Main.java' defines a 'Main' class with a 'main' method. The method prints the substring of 'mueez' from index 0 to 4, and then the substring starting from index 3. The console output shows the command to run the program, followed by the output 'muee' and 'ez' on separate lines, and finally 'Process finished with exit code 0'.

```
com > company > Main > main
Main.java x Stringbuffer.java x
1 public class Main {
2     public static void main(String[] args) {
3         //print name start to end
4         String name = "mueez";
5         System.out.println(name.substring(0,4)); //start to end point
6         System.out.println(name.substring(3)); //start point
7
8     }
9 }
10

"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea-agent-1.0.0.jar" -Dfile.encoding=UTF-8
muee
ez

Process finished with exit code 0
```

If want to check the contains are same or not



The screenshot shows an IDE with two tabs: `Main.java` and `Stringbuffer.java`. The `Main.java` tab is active, displaying the following code:

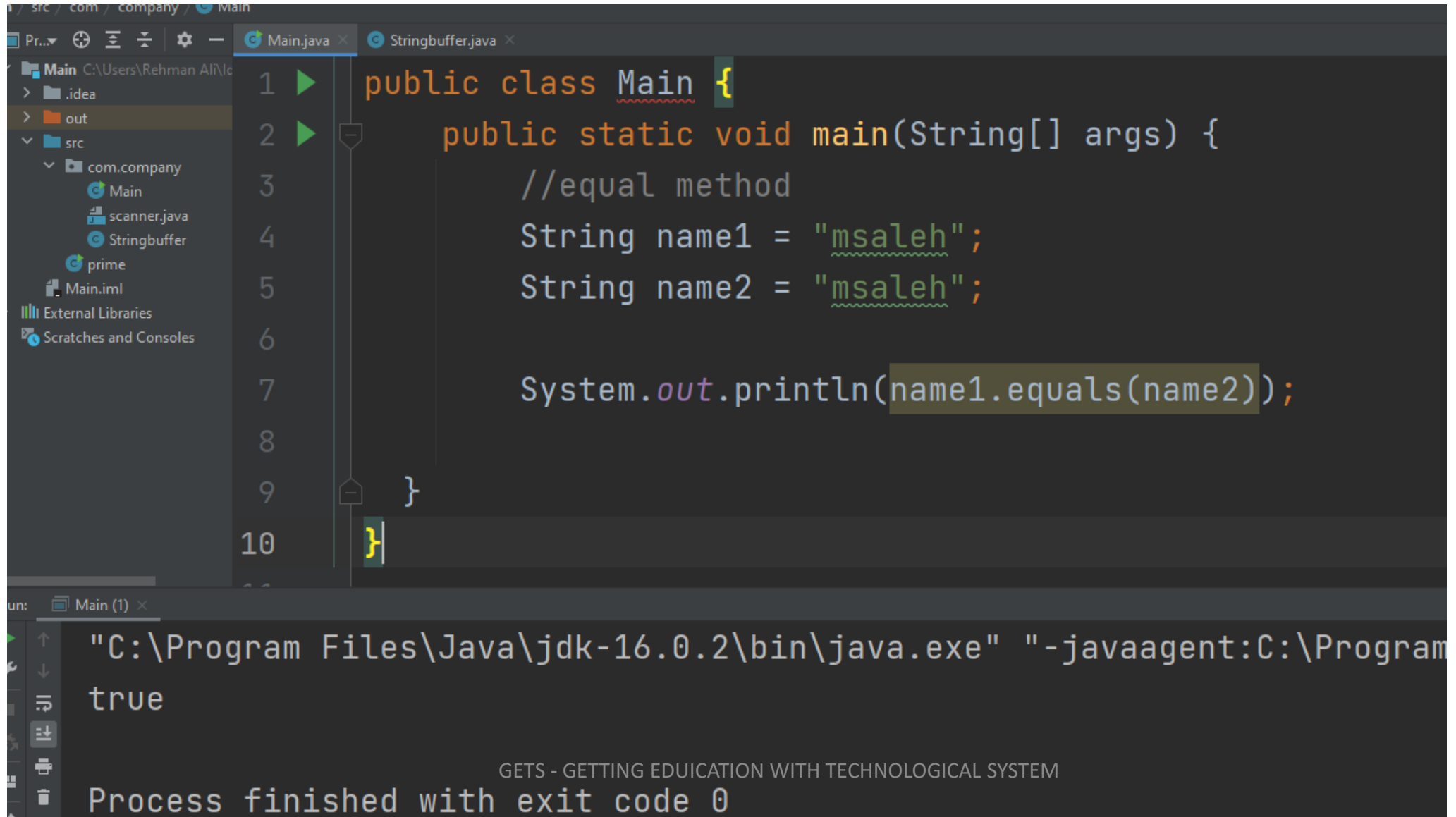
```
1 public class Main {  
2     public static void main(String[] args) {  
3         //contains check same values or not  
4         String sentence = "today haseeb will go to larkana for persal work";  
5         System.out.println(sentence.contains("bhai"));  
6         //we can also use in boolean variable  
7         boolean checksentence = sentence.contains("haseeb");  
8         System.out.println(checksentence);  
9     }  
10 }
```

The output console at the bottom shows the execution of the program:

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\Intel  
false  
true  
Process finished with exit code 0
```

The output shows that the first check, `contains("bhai")`, returns `false`, and the second check, `contains("haseeb")`, returns `true`.

If want to check the string are same using equal method



The screenshot shows an IDE with two tabs: `Main.java` and `Stringbuffer.java`. The `Main.java` tab is active, displaying the following code:

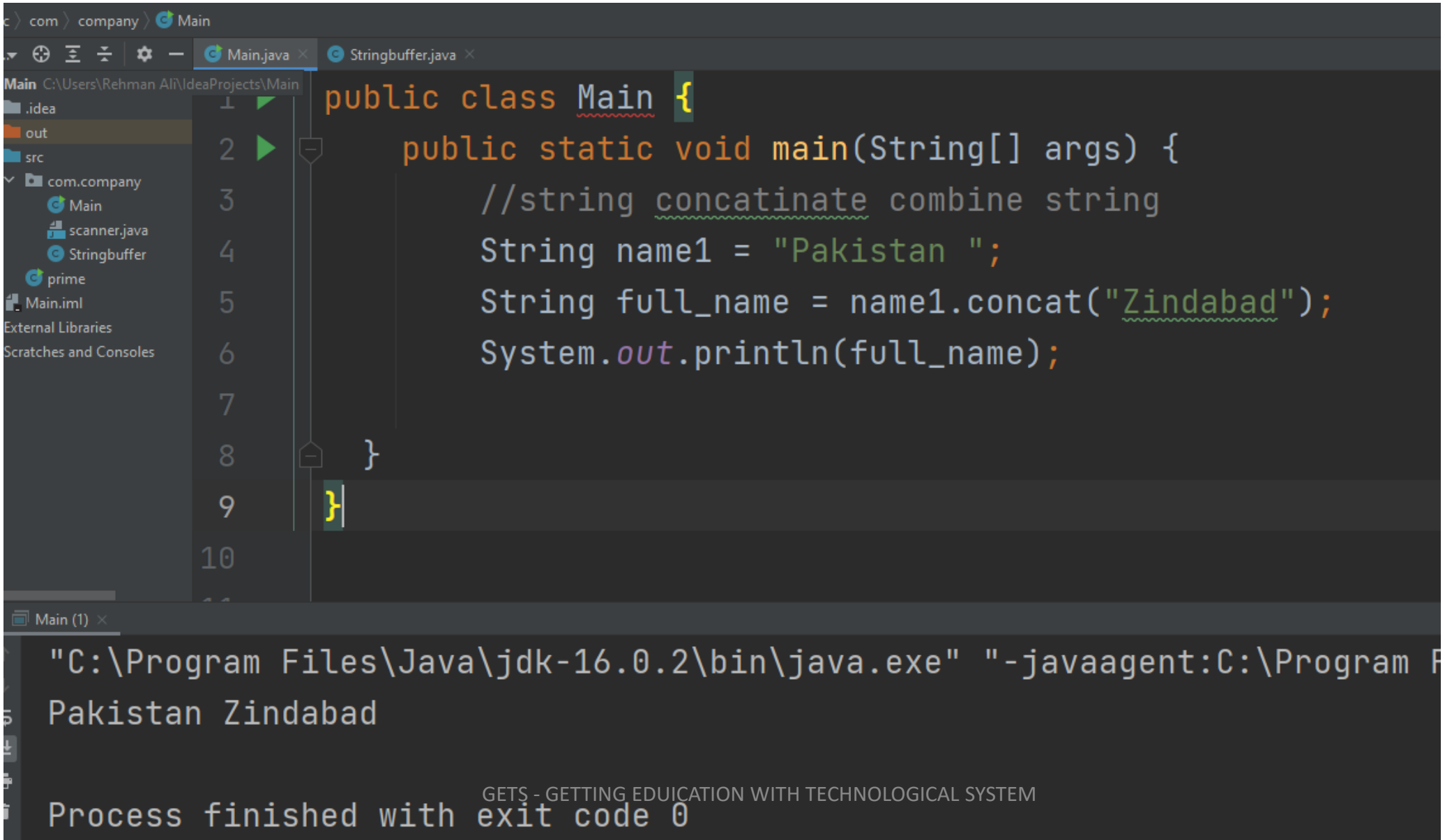
```
1 public class Main {  
2     public static void main(String[] args) {  
3         //equal method  
4         String name1 = "msaleh";  
5         String name2 = "msaleh";  
6  
7         System.out.println(name1.equals(name2));  
8  
9     }  
10 }
```

The IDE's left sidebar shows a project structure with a package `com.company` containing `Main`, `scanner.java`, and `Stringbuffer`. The bottom console shows the execution command and output:

```
run: "C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program  
true  
Process finished with exit code 0
```

At the bottom of the console, the text "GETS - GETTING EDUCATION WITH TECHNOLOGICAL SYSTEM" is visible.

If want to concatenate two String



The screenshot shows an IDE with a project named 'Main'. The 'Main.java' file is open, displaying the following code:

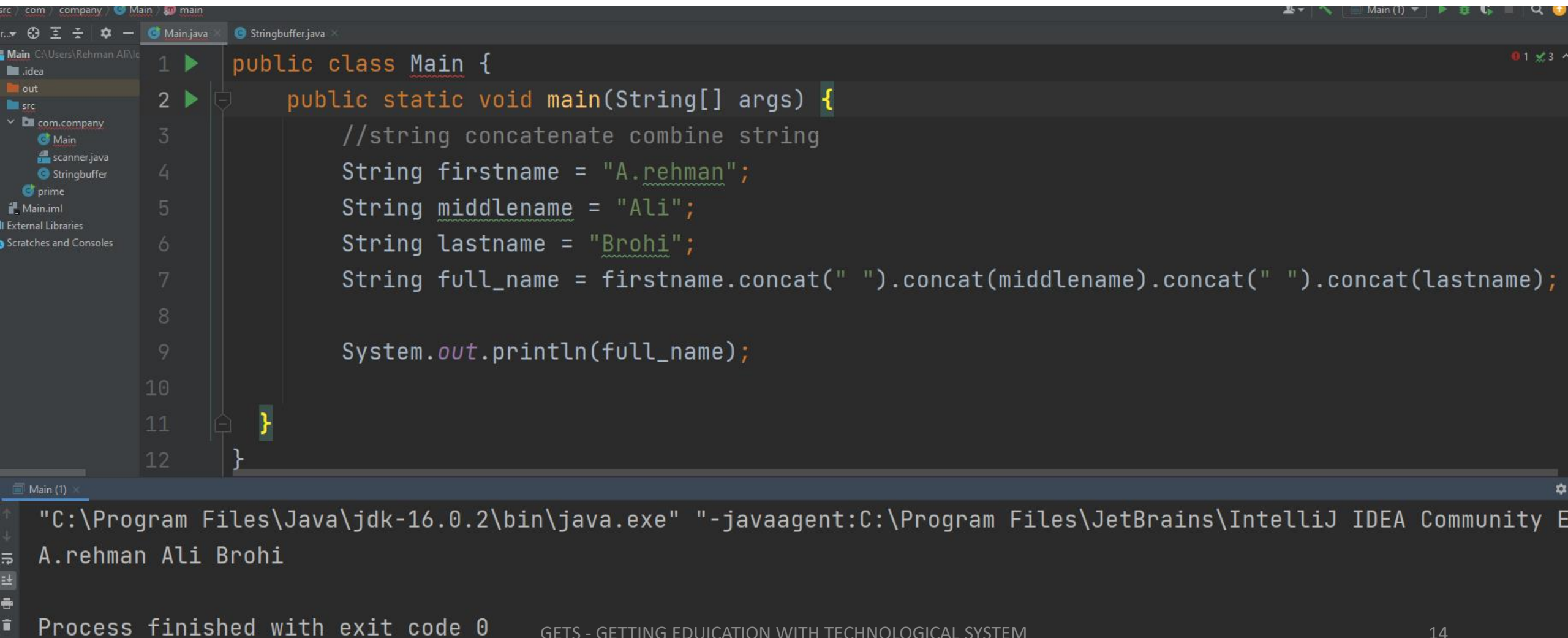
```
public class Main {  
    public static void main(String[] args) {  
        //string concatenate combine string  
        String name1 = "Pakistan ";  
        String full_name = name1.concat("Zindabad");  
        System.out.println(full_name);  
    }  
}
```

The output console at the bottom shows the command used to run the program and the resulting output:

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program F  
Pakistan Zindabad  
Process finished with exit code 0
```

GETS - GETTING EDUCATION WITH TECHNOLOGICAL SYSTEM

If want to concatenate multiple Strings

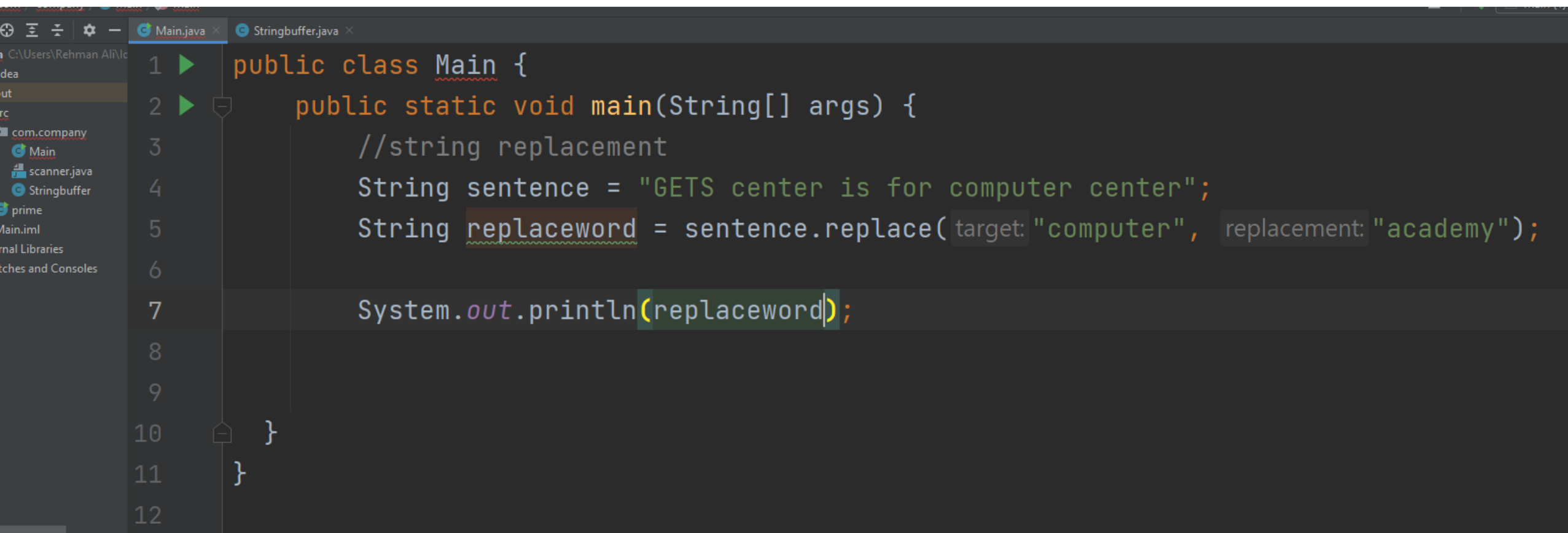


The screenshot displays an IDE window with a Java file named `Main.java`. The code defines a `Main` class with a `main` method that concatenates three strings: `"A.rehman"`, `"Ali"`, and `"Brohi"`. The concatenated result is printed to the console. The output window at the bottom shows the command used to run the program and the resulting output `A.rehman Ali Brohi`.

```
1 public class Main {  
2     public static void main(String[] args) {  
3         //string concatenate combine string  
4         String firstname = "A.rehman";  
5         String middlename = "Ali";  
6         String lastname = "Brohi";  
7         String full_name = firstname.concat(" ").concat(middlename).concat(" ").concat(lastname);  
8  
9         System.out.println(full_name);  
10  
11     }  
12 }
```

Process finished with exit code 0

If want to replace word



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

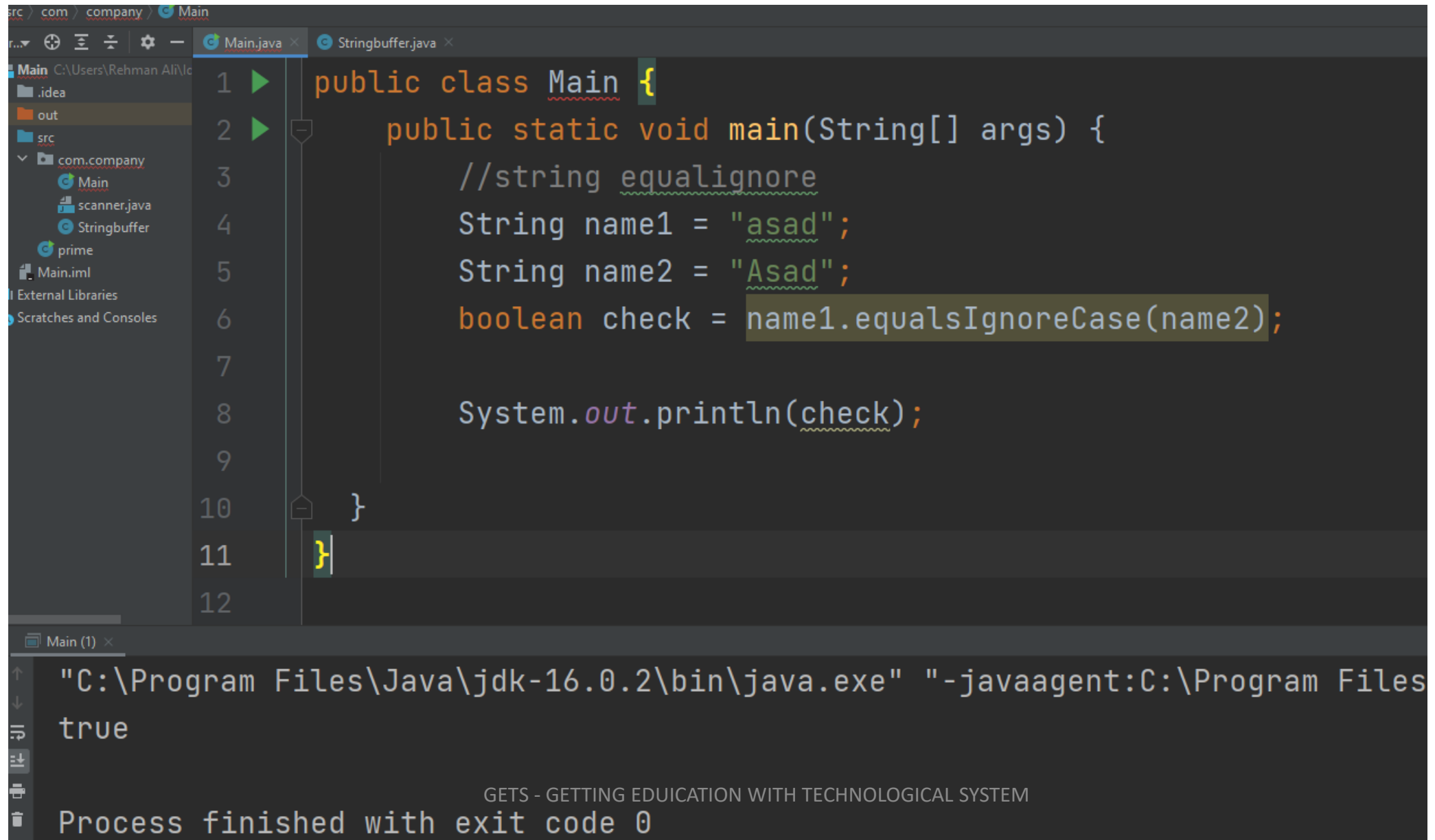
```
1 public class Main {  
2     public static void main(String[] args) {  
3         //string replacement  
4         String sentence = "GETS center is for computer center";  
5         String replaceword = sentence.replace(target: "computer", replacement: "academy");  
6  
7         System.out.println(replaceword);  
8  
9  
10    }  
11 }  
12
```

The left sidebar shows a project structure with a package 'com.company' containing files 'Main', 'scanner.java', and 'Stringbuffer'. The bottom status bar shows 'Main (1)'.

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA  
GETS center is for academy center  
  
Process finished with exit code 0
```

GETS - GETTING EDUCATION WITH TECHNOLOGICAL SYSTEM

If want to print words but ignore the uppercase or lower



The screenshot shows an IDE with a project named 'Main' in the package 'com.company'. The file 'Main.java' is open, displaying the following code:

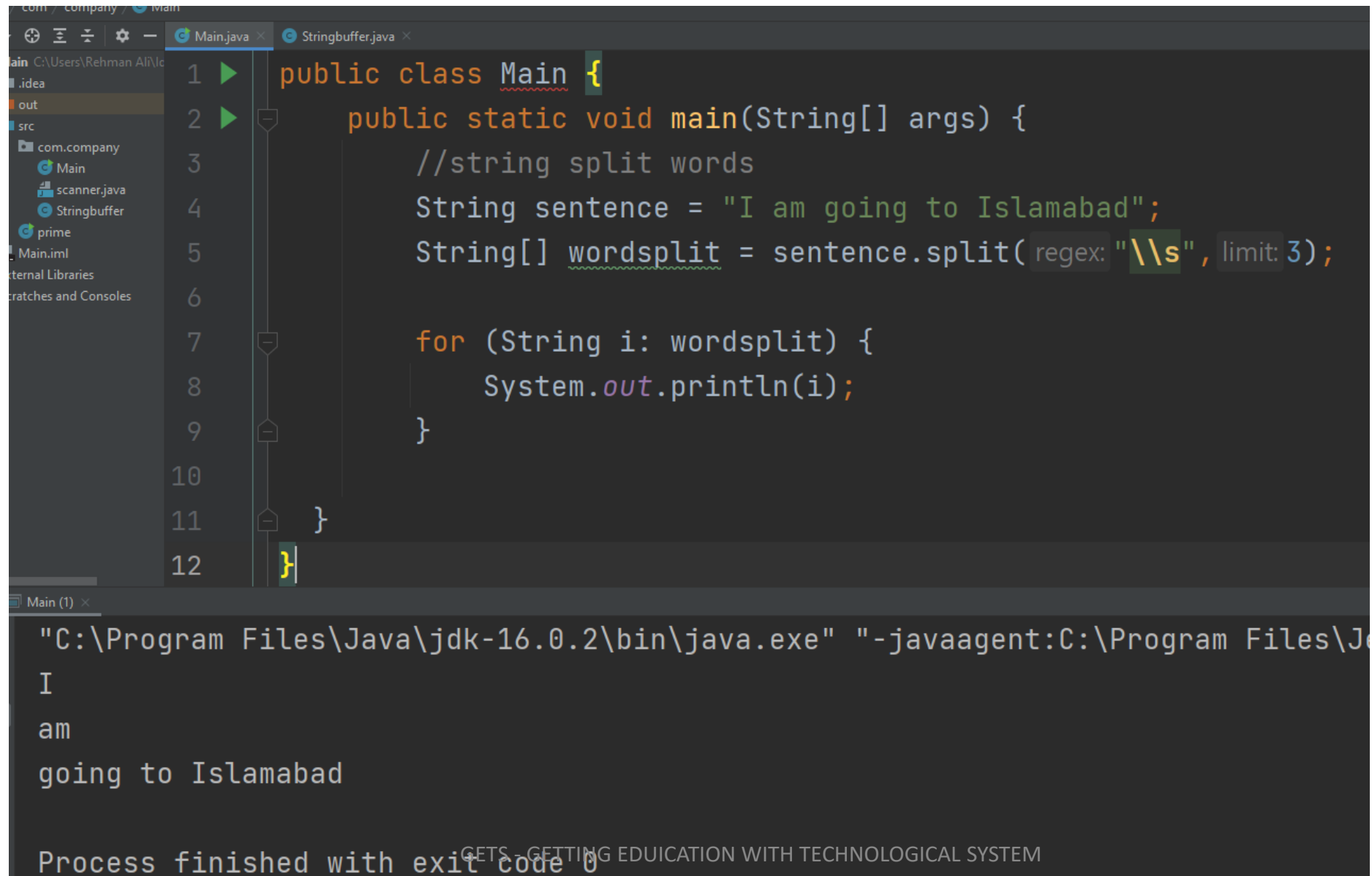
```
1 public class Main {  
2     public static void main(String[] args) {  
3         //string equalignore  
4         String name1 = "asad";  
5         String name2 = "Asad";  
6         boolean check = name1.equalsIgnoreCase(name2);  
7  
8         System.out.println(check);  
9     }  
10 }  
11  
12
```

The code is executed, and the output is shown in the console:

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files  
true  
Process finished with exit code 0
```

The output 'true' indicates that the strings 'asad' and 'Asad' are equal when case is ignored.

If want to split the words



The screenshot shows an IDE with two tabs: `Main.java` and `Stringbuffer.java`. The `Main.java` tab is active, displaying the following Java code:

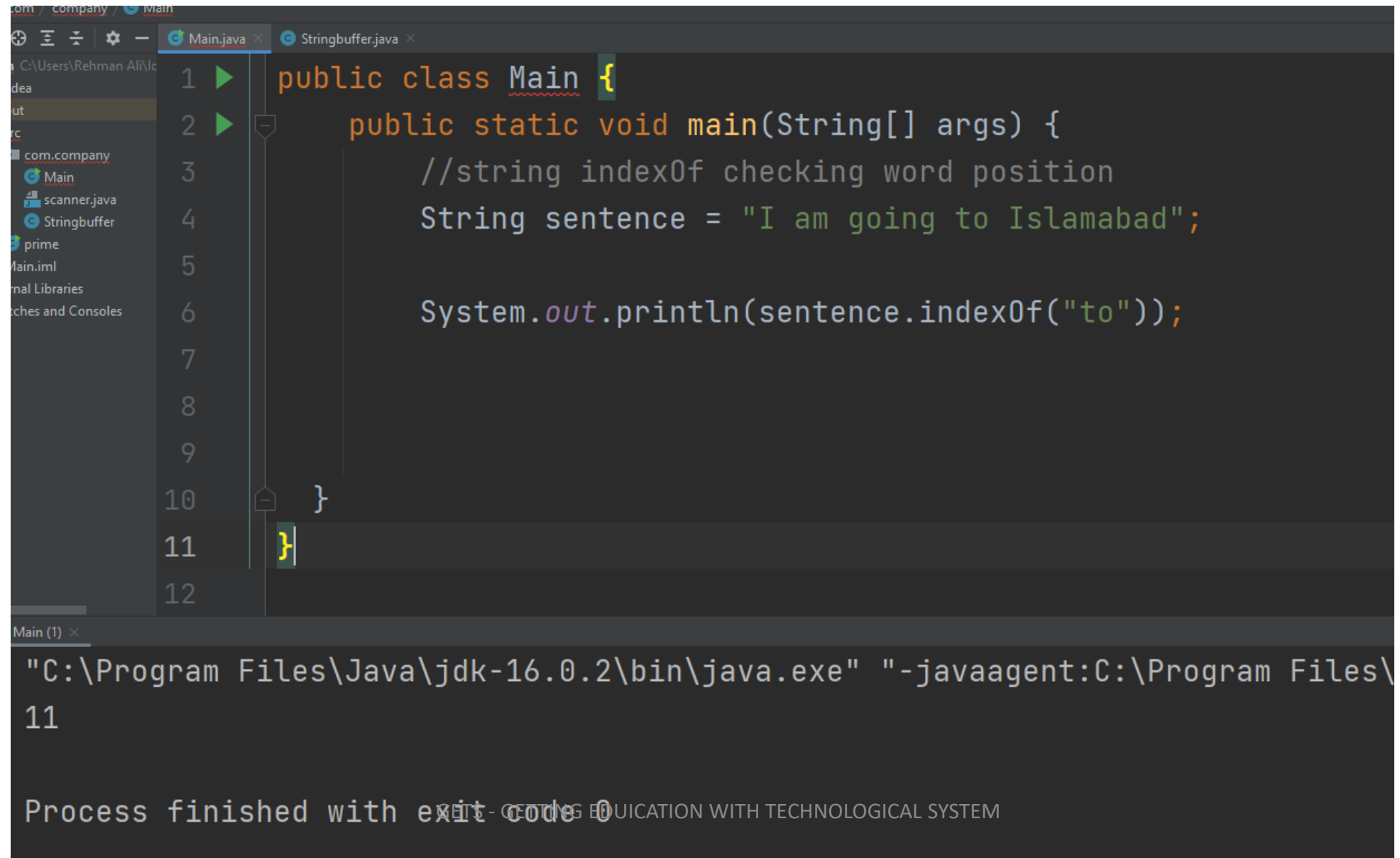
```
1 public class Main {  
2     public static void main(String[] args) {  
3         //string split words  
4         String sentence = "I am going to Islamabad";  
5         String[] wordsplit = sentence.split(regex: "\\s", limit: 3);  
6  
7         for (String i: wordsplit) {  
8             System.out.println(i);  
9         }  
10  
11     }  
12 }
```

The console output at the bottom shows the execution of the program:

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\J  
I  
am  
going to Islamabad  
  
Process finished with exit code 0
```

GETS - GETTING EDUCATION WITH TECHNOLOGICAL SYSTEM

If want to check the position of word using indexOf() method

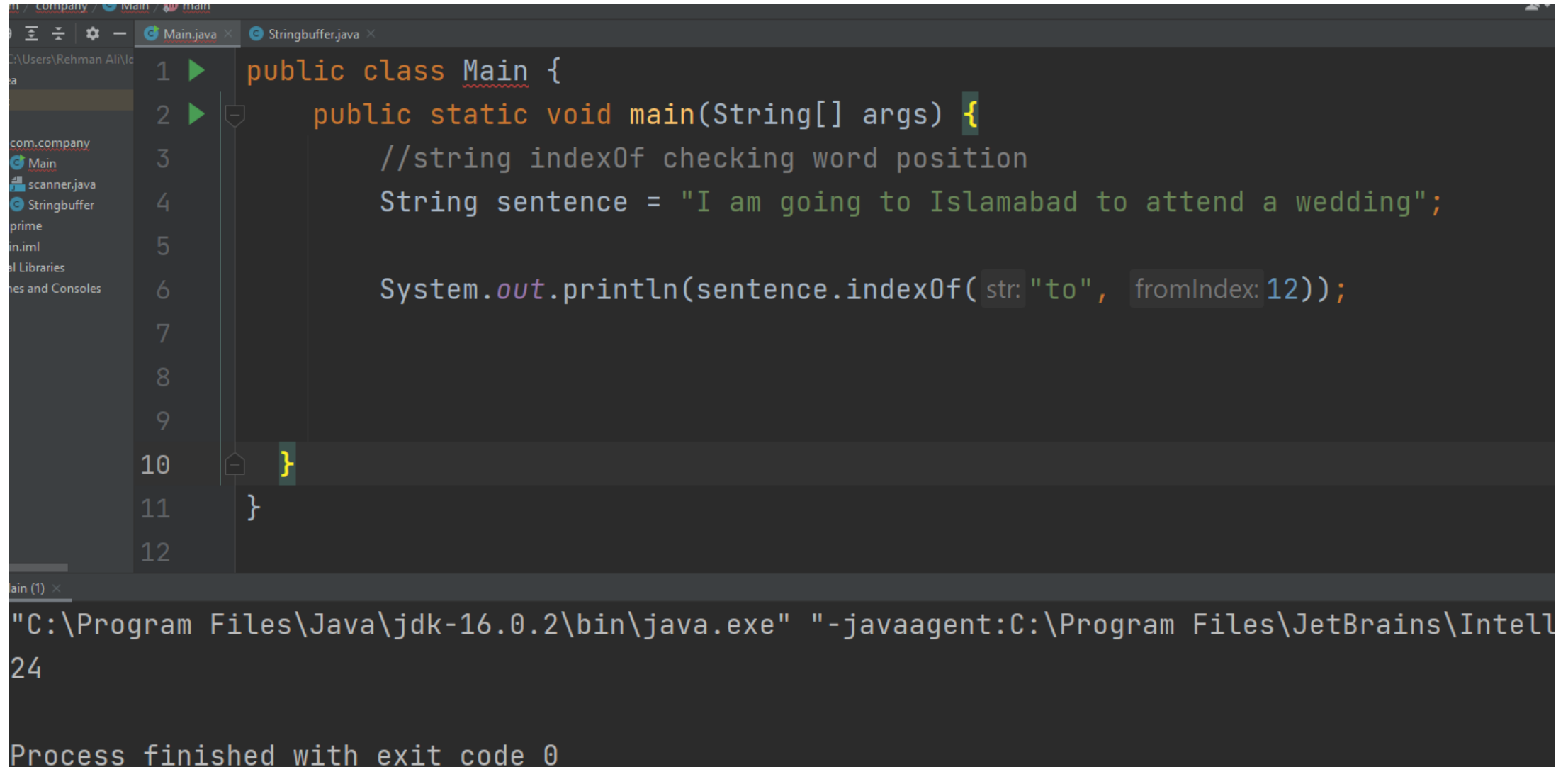


The screenshot shows an IDE with two tabs: 'Main.java' and 'Stringbuffer.java'. The 'Main.java' tab is active, displaying the following code:

```
1 public class Main {
2     public static void main(String[] args) {
3         //string indexOf checking word position
4         String sentence = "I am going to Islamabad";
5
6         System.out.println(sentence.indexOf("to"));
7
8
9
10    }
11 }
12
```

The output window at the bottom shows the command executed: `"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\` and the output: `11`. Below the output, it states: `Process finished with exit code 0`.

If same word appear multiple times



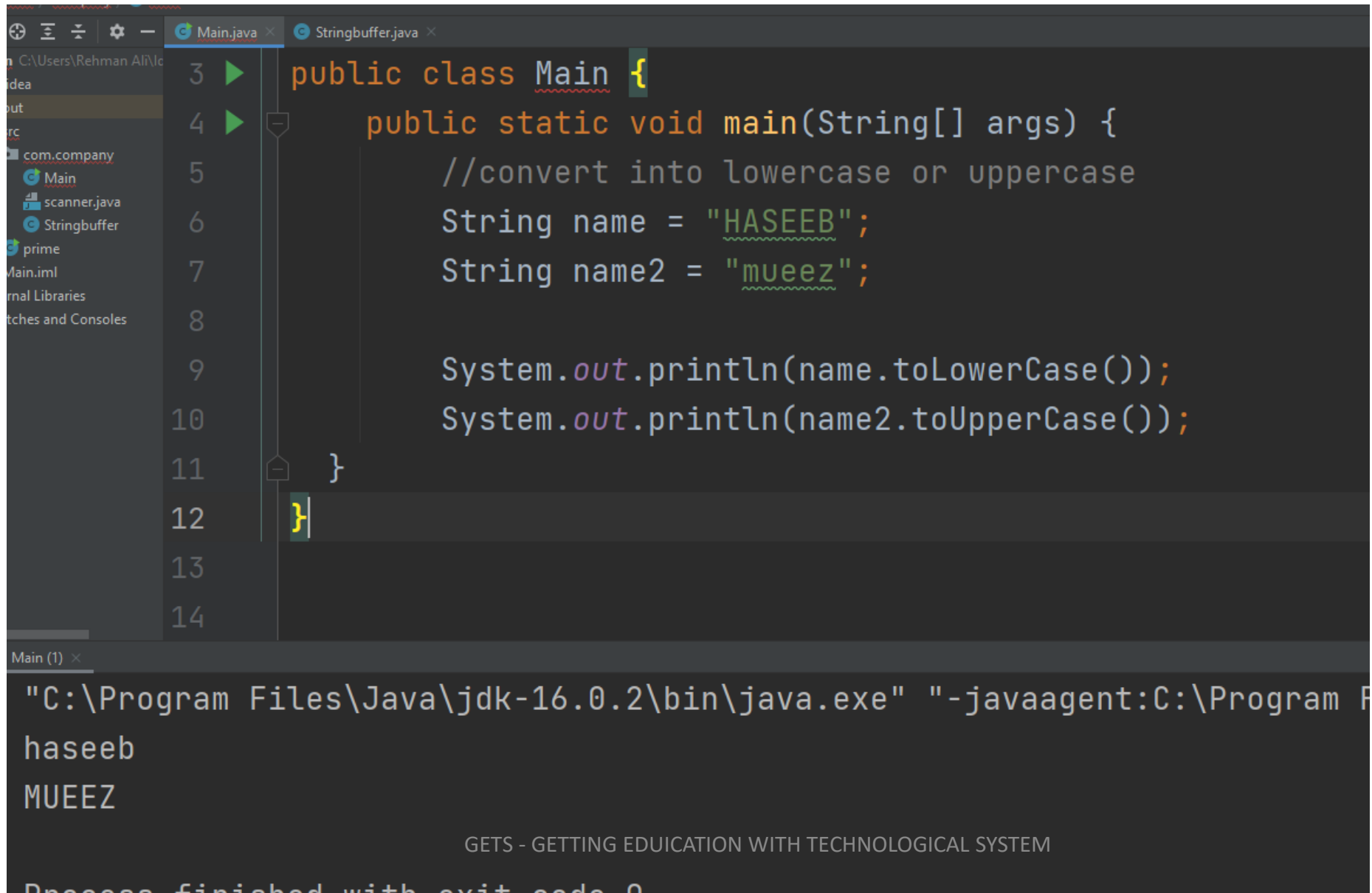
The screenshot shows an IDE with two tabs: 'Main.java' and 'Stringbuffer.java'. The 'Main.java' tab is active, displaying the following code:

```
1 public class Main {  
2     public static void main(String[] args) {  
3         //string indexOf checking word position  
4         String sentence = "I am going to Islamabad to attend a wedding";  
5  
6         System.out.println(sentence.indexOf(str: "to", fromIndex: 12));  
7  
8  
9  
10    }  
11 }  
12
```

The output console at the bottom shows the command executed and the result:

```
main (1) ×  
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ  
24  
  
Process finished with exit code 0
```

If want to change words into upper or lower case



The screenshot shows an IDE with two tabs: `Main.java` and `Stringbuffer.java`. The `Main.java` tab is active, displaying the following code:

```
3 public class Main {
4     public static void main(String[] args) {
5         //convert into lowercase or uppercase
6         String name = "HASEEB";
7         String name2 = "mueez";
8
9         System.out.println(name.toLowerCase());
10        System.out.println(name2.toUpperCase());
11    }
12 }
13
14
```

The output console at the bottom shows the execution results:

```
Main (1) x
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program F
haseeb
MUEEZ
Process finished with exit code 0
```

GETS - GETTING EDUCATION WITH TECHNOLOGICAL SYSTEM

If want to convert int into String using valueOf() method



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

```
3 public class Main {
4     public static void main(String[] args) {
5         //String valueOf() convert int into String
6         int number = 10;
7         String numintostring = String.valueOf(number);
8
9         System.out.println(numintostring);
10        System.out.println(numintostring+20);
11    }
12 }
13
14
```

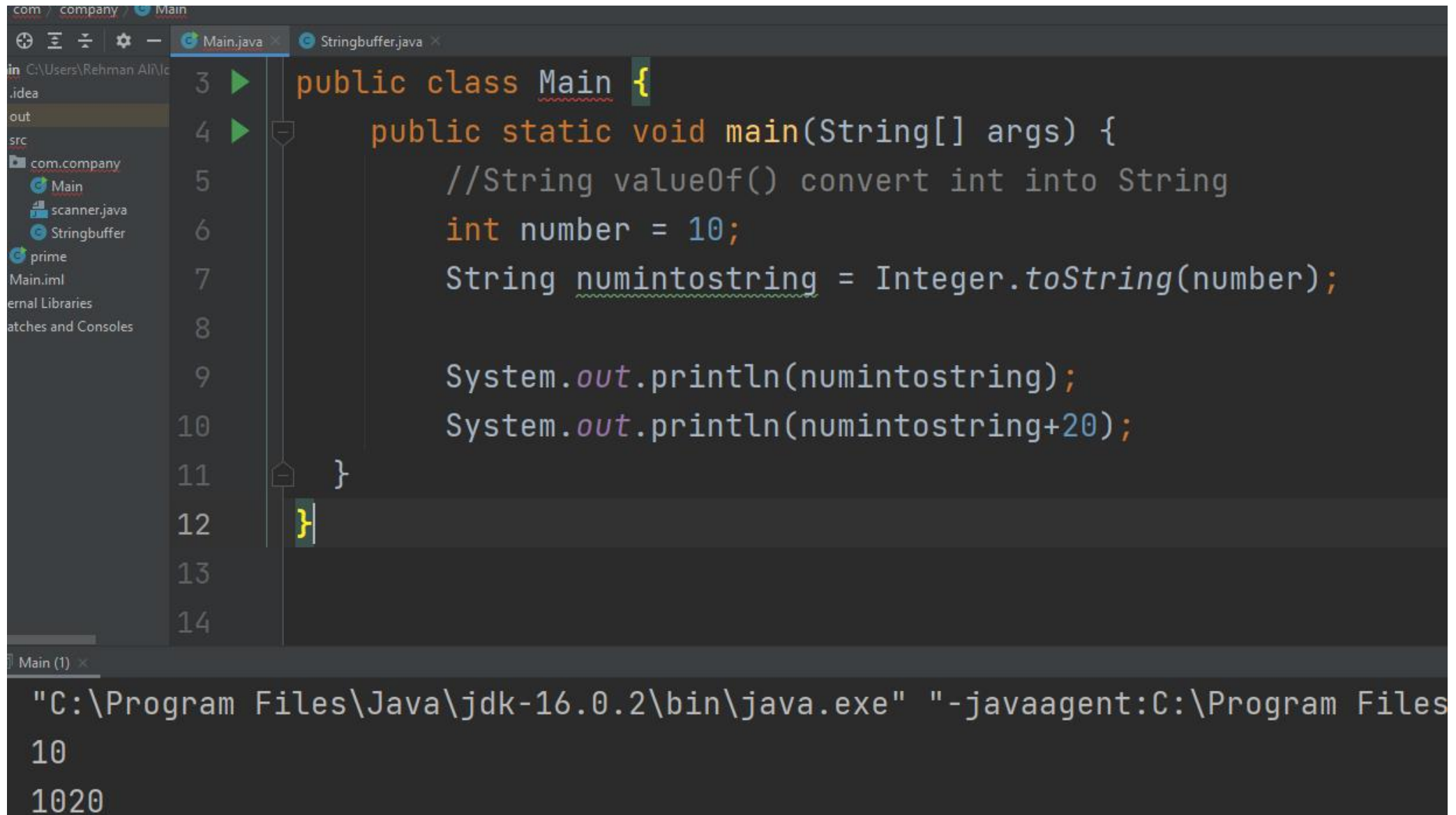
The output console at the bottom shows the execution results:

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Fi
10
1020

Process finished with exit code 0
```

GETS - GETTING EDUCATION WITH TECHNOLOGICAL SYSTEM

If want to convert int into String using toString() method



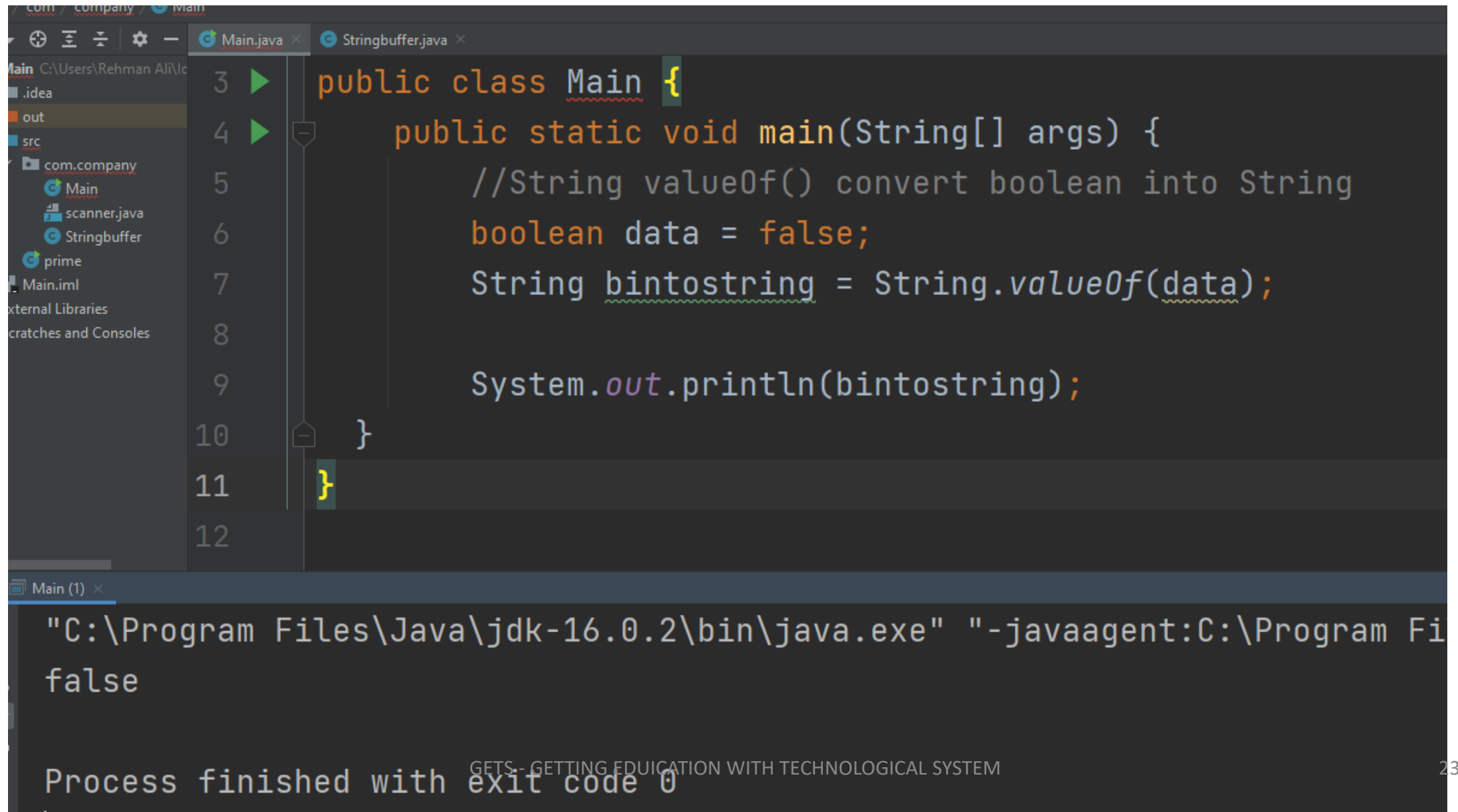
The screenshot shows an IDE with two tabs: `Main.java` and `Stringbuffer.java`. The `Main.java` tab is active, displaying the following code:

```
3 public class Main {  
4     public static void main(String[] args) {  
5         //String.valueOf() convert int into String  
6         int number = 10;  
7         String numintostring = Integer.toString(number);  
8  
9         System.out.println(numintostring);  
10        System.out.println(numintostring+20);  
11    }  
12 }  
13  
14
```

The console output at the bottom shows the command to run the program and its results:

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files  
10  
1020
```

If want to convert Boolean into String using `valueOf()` method

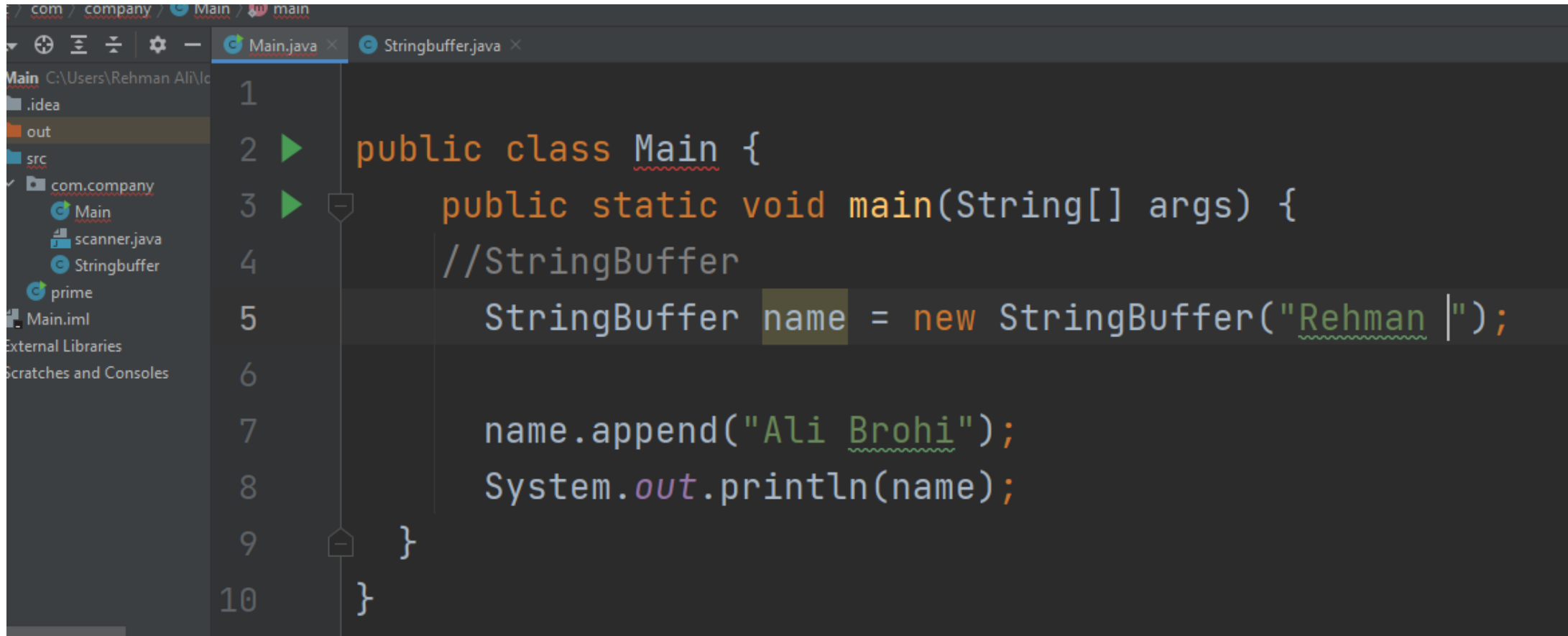


```
3 public class Main {
4     public static void main(String[] args) {
5         //String valueOf() convert boolean into String
6         boolean data = false;
7         String bintostring = String.valueOf(data);
8
9         System.out.println(bintostring);
10    }
11 }
12
```

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Fi
false

Process finished with exit code 0
```

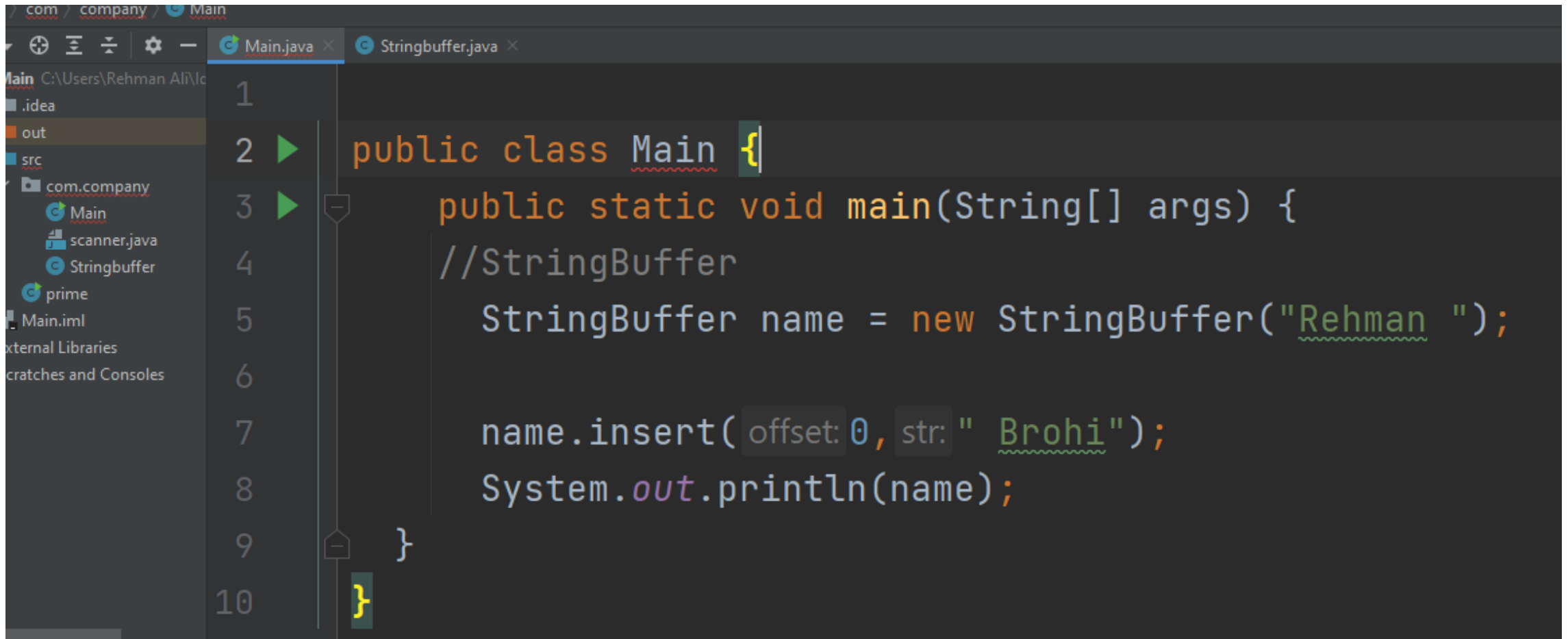
StringBuffer using append() method

A screenshot of an IDE window showing a Java file named Main.java. The code defines a public class Main with a main method. Inside the main method, a StringBuffer object named 'name' is created with the initial value "Rehman |". Then, the append method is used to add "Ali Brohi" to the end of the string. Finally, the contents of the StringBuffer are printed to the console using System.out.println. The IDE's project explorer on the left shows the file structure, and the bottom panel shows the command used to run the program.

```
1  
2 public class Main {  
3     public static void main(String[] args) {  
4         //StringBuffer  
5         StringBuffer name = new StringBuffer("Rehman |");  
6  
7         name.append("Ali Brohi");  
8         System.out.println(name);  
9     }  
10 }
```

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program  
Rehman Ali Brohi
```


StringBuffer with insert() method



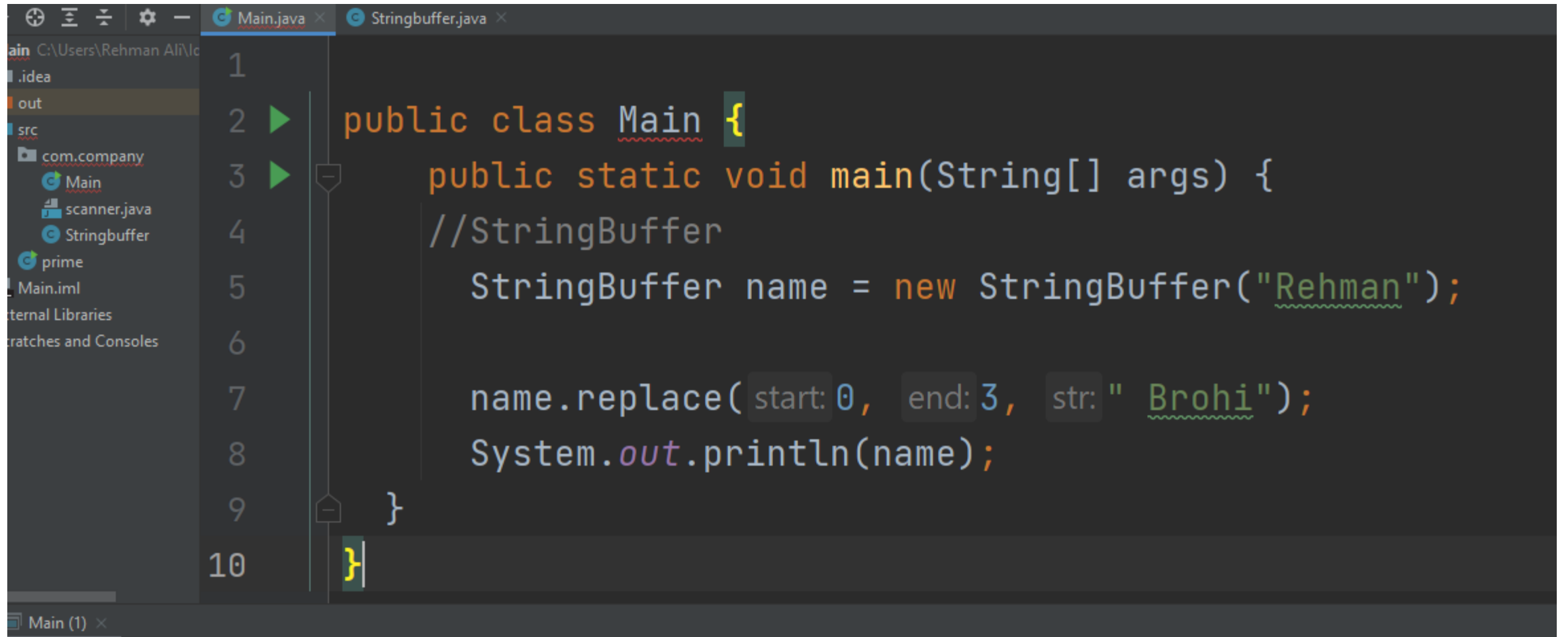
The screenshot shows an IDE with two tabs: 'Main.java' and 'Stringbuffer.java'. The 'Main.java' tab is active, displaying the following code:

```
1  
2 public class Main {  
3     public static void main(String[] args) {  
4         //StringBuffer  
5         StringBuffer name = new StringBuffer("Rehman ");  
6  
7         name.insert(offset: 0, str: " Brohi");  
8         System.out.println(name);  
9     }  
10 }
```

The code demonstrates the use of the `insert()` method on a `StringBuffer` object. The initial string is "Rehman ". The `insert()` method is called with an offset of 0 and the string " Brohi", resulting in the output "BrohiRehman".

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program  
BrohiRehman
```

StringBuffer with replace() method



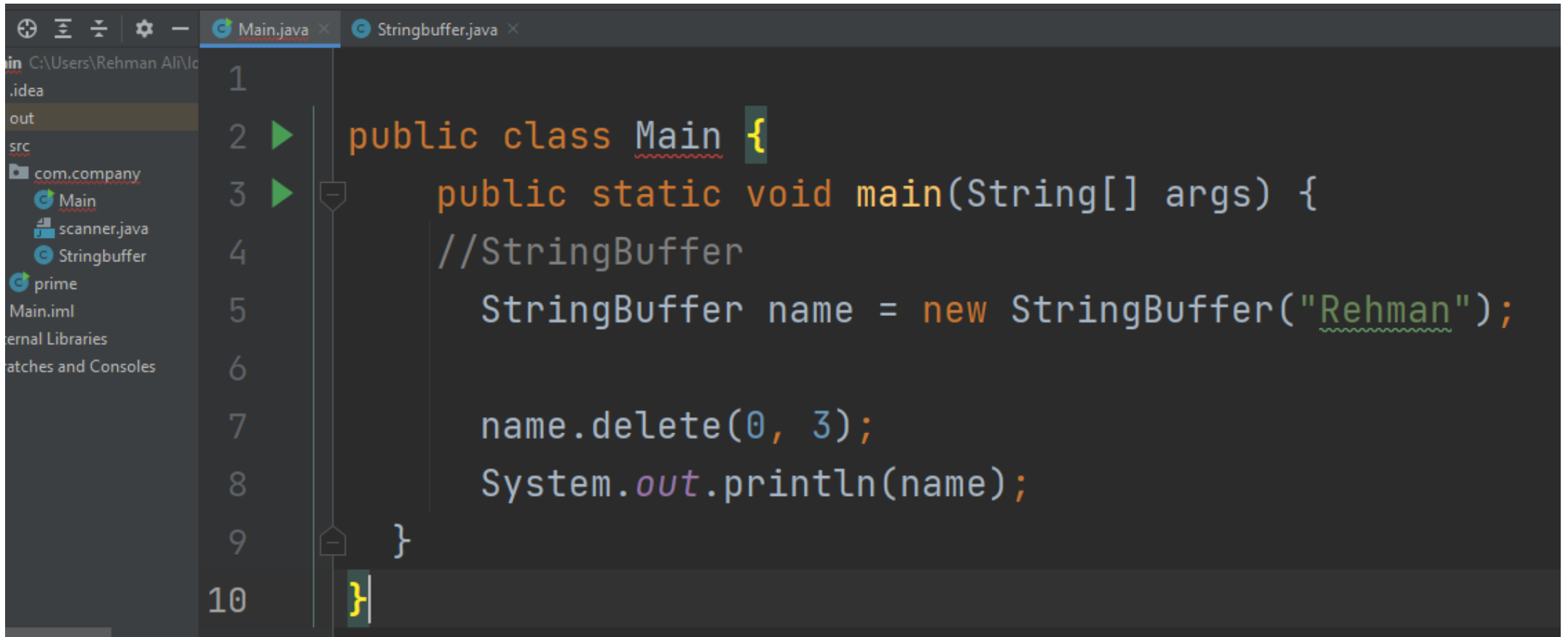
The screenshot shows an IDE with two tabs: 'Main.java' and 'Stringbuffer.java'. The 'Main.java' tab is active, displaying the following code:

```
1  
2 public class Main {  
3     public static void main(String[] args) {  
4         //StringBuffer  
5         StringBuffer name = new StringBuffer("Rehman");  
6  
7         name.replace(start: 0, end: 3, str: " Brohi");  
8         System.out.println(name);  
9     }  
10 }
```

The left sidebar shows a project structure with a package 'com.company' containing files 'Main', 'scanner.java', and 'Stringbuffer'. The bottom status bar shows 'Main (1) x'.

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program F  
Brohiman
```

StringBuffer with delete() method



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

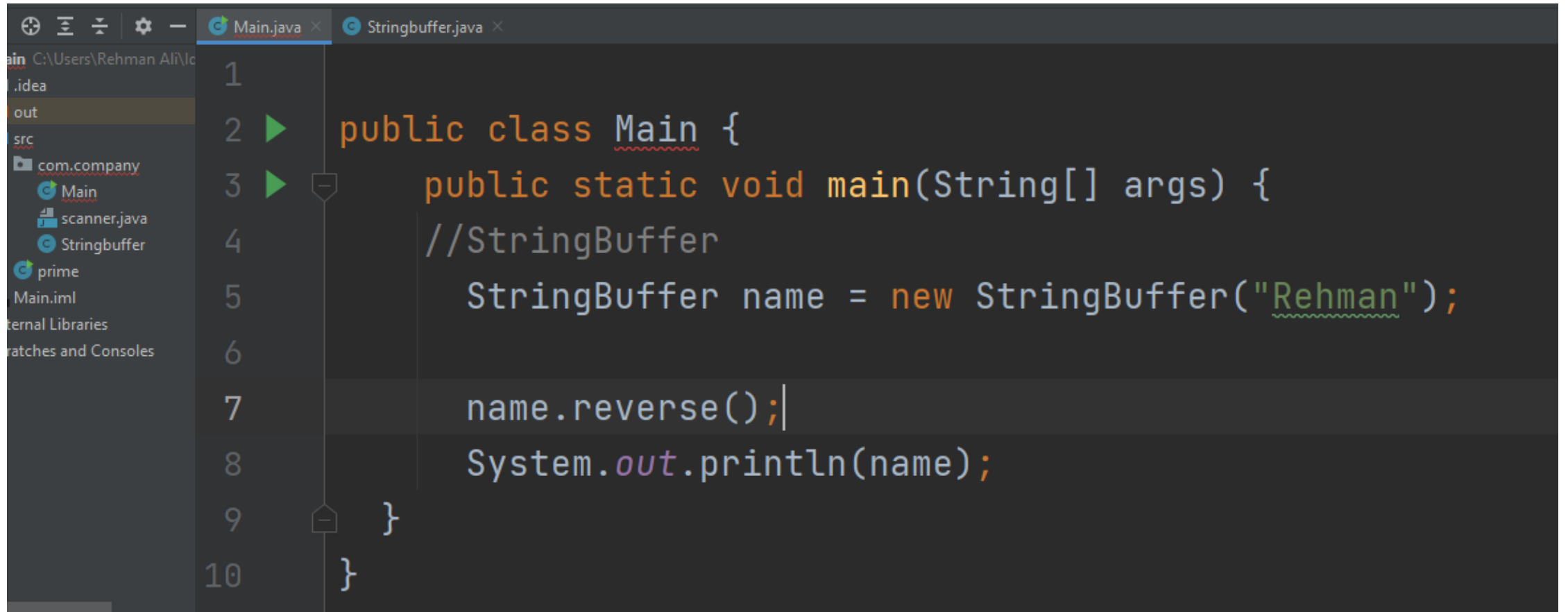
```
1  
2 public class Main {  
3     public static void main(String[] args) {  
4         //StringBuffer  
5         StringBuffer name = new StringBuffer("Rehman");  
6  
7         name.delete(0, 3);  
8         System.out.println(name);  
9     }  
10 }
```

The left sidebar shows a project structure with a package 'com.company' containing files 'Main', 'scanner.java', and 'Stringbuffer'. The bottom status bar shows 'Main (1)'.

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Progra  
man
```

```
Process finished with exit code 0
```

StringBuffer with reverse() method



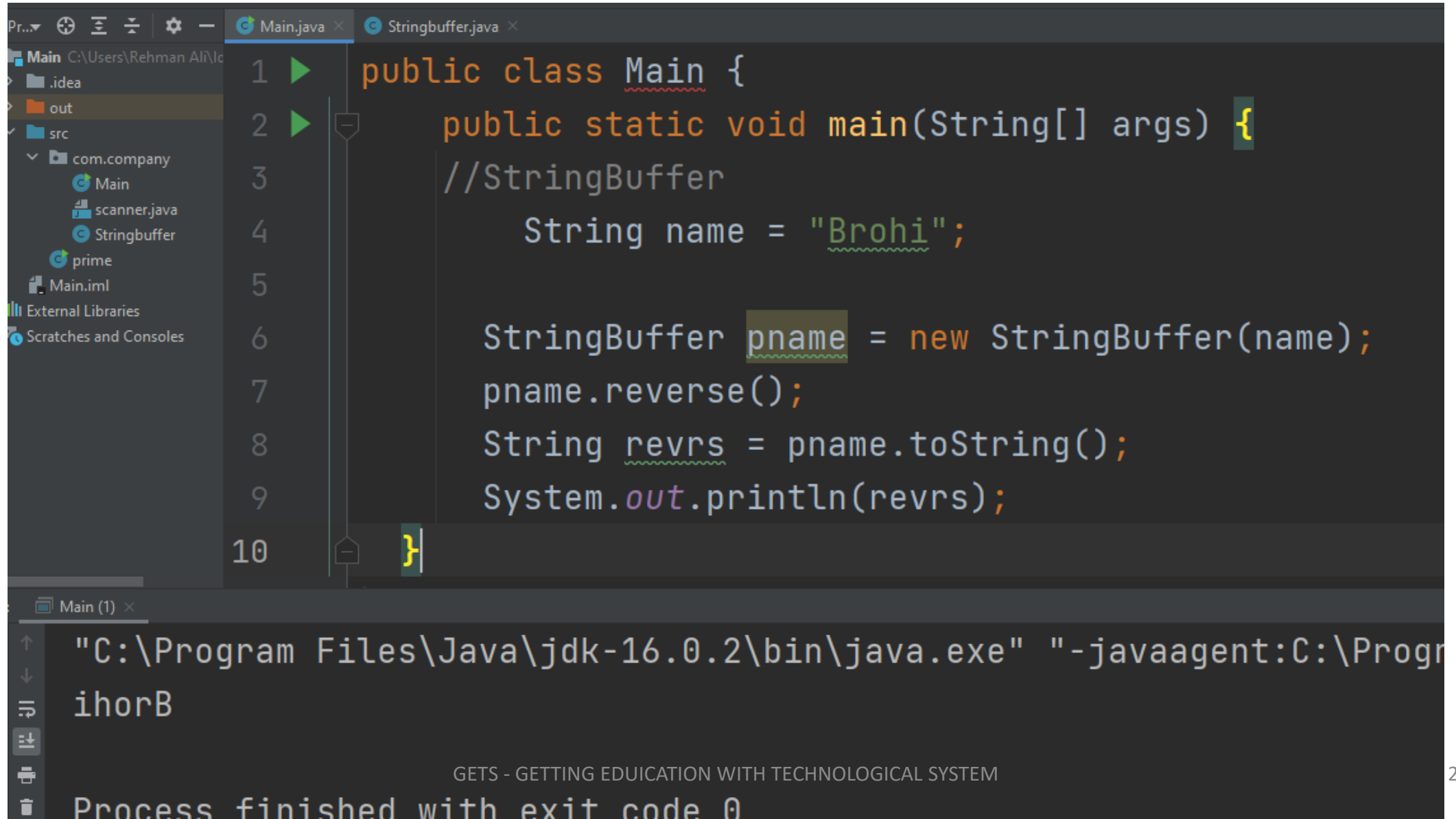
```
1  
2 public class Main {  
3     public static void main(String[] args) {  
4         //StringBuffer  
5         StringBuffer name = new StringBuffer("Rehman");  
6  
7         name.reverse();  
8         System.out.println(name);  
9     }  
10 }
```

Main (1) ×

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program F  
namheR
```

Process finished with exit code 0

StringBuffer with reverse() method in string singular



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

```
1 public class Main {  
2     public static void main(String[] args) {  
3         //StringBuffer  
4         String name = "Brohi";  
5  
6         StringBuffer pname = new StringBuffer(name);  
7         pname.reverse();  
8         String revrs = pname.toString();  
9         System.out.println(revrs);  
10    }
```

The left sidebar shows a project structure with a package 'com.company' containing 'Main', 'scanner.java', and 'Stringbuffer'. The bottom console shows the execution output:

```
Main (1) x  
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Progr  
ihorB  
Process finished with exit code 0
```

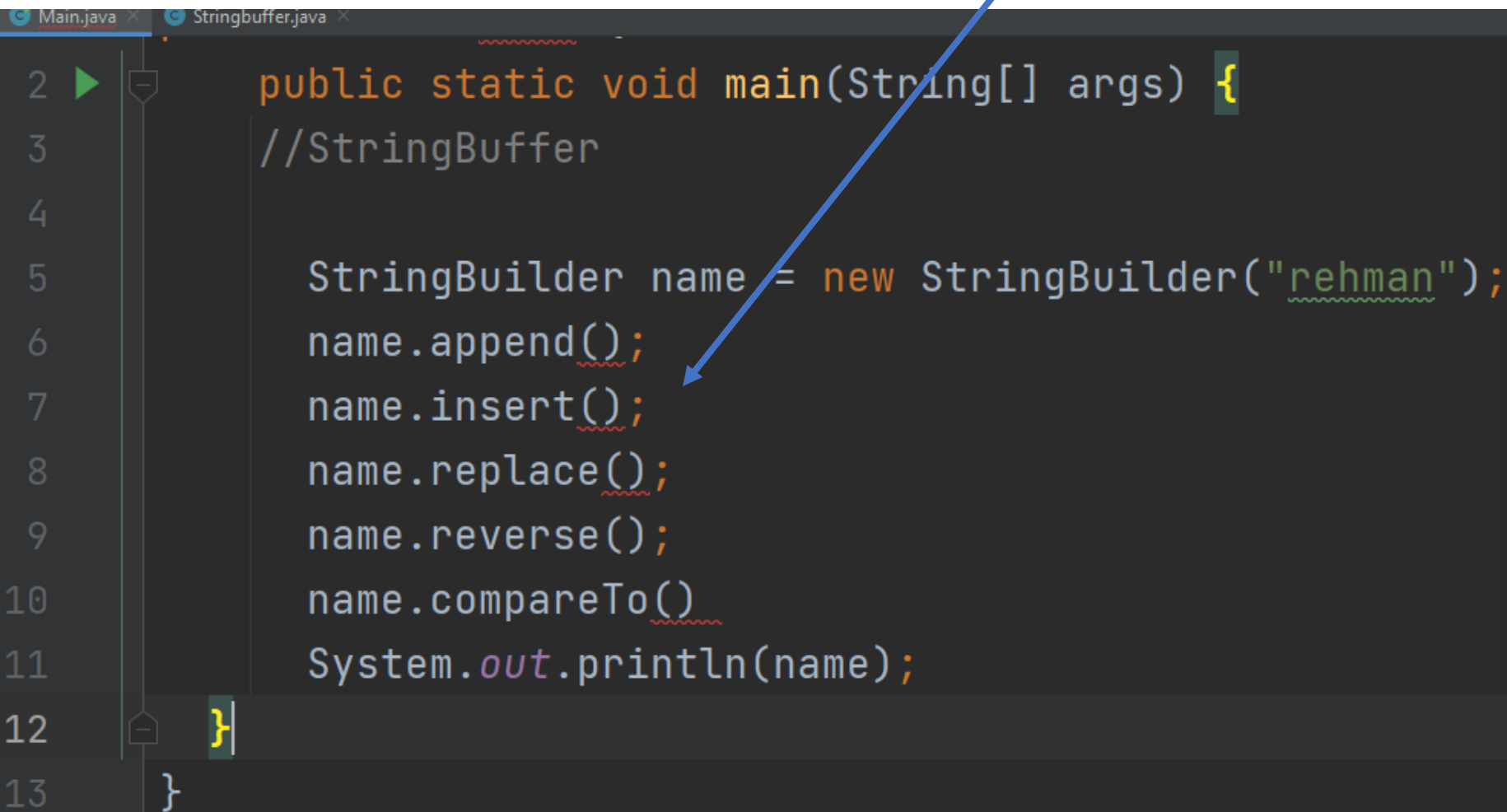
At the bottom of the console, the text 'GETS - GETTING EDUCATION WITH TECHNOLOGICAL SYSTEM' is visible.

What is Different Between **StringBuffer** VS **StringBuilder**

StringBuffer is
less efficient than
StringBuilder

StringBuilder is
more efficient
than StringBuffer.

You can use all method in **StringBuilder**



```
2 public static void main(String[] args) {
3     //StringBuilder
4
5     StringBuilder name = new StringBuilder("rehman");
6     name.append();
7     name.insert();
8     name.replace();
9     name.reverse();
10    name.compareTo()
11    System.out.println(name);
12 }
13 }
```

A screenshot of a Java IDE with two tabs: 'Main.java' and 'Stringbuffer.java'. The 'Stringbuffer.java' tab is active, showing a Java program. The code defines a 'main' method that creates a 'StringBuilder' object named 'name' with the initial value 'rehman'. It then lists several methods: 'append()', 'insert()', 'replace()', 'reverse()', and 'compareTo()'. The 'append()' method on line 6 is underlined with a red squiggly line, and a blue arrow points from the title 'You can use all method in StringBuilder' to this line. The 'compareTo()' method on line 10 is also underlined with a red squiggly line. The program ends with 'System.out.println(name);' and closing braces. The IDE interface includes a line number margin on the left, a gutter with icons, and a bottom status bar with 'Build' and 'Build Output' buttons.

In this program we will use differentiate time

```
1 public class Main {
2     public static void main(String[] args) {
3         long starttime = System.currentTimeMillis();
4         StringBuffer name = new StringBuffer("JAVA");
5         for(int i = 0; i<= 10000; i++) {
6             name.append("hello");
7         }
8         System.out.println("time taken to execute the StringBuffer is: "+(System.currentTimeMillis()-starttime)+
9
10            starttime = System.currentTimeMillis();
11            StringBuilder name2 = new StringBuilder("JAVA");
12            for(int i = 0; i<= 10000; i++) {
13                name.append("hello");
14            }
15            System.out.println("time taken by StringBuilder is: "+(System.currentTimeMillis()-starttime) + "ms");
16        }
17    }
18 }
```

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.1.3\
time taken to execute the StringBuffer is: 2ms
time taken by StringBuilder is: 1ms
```