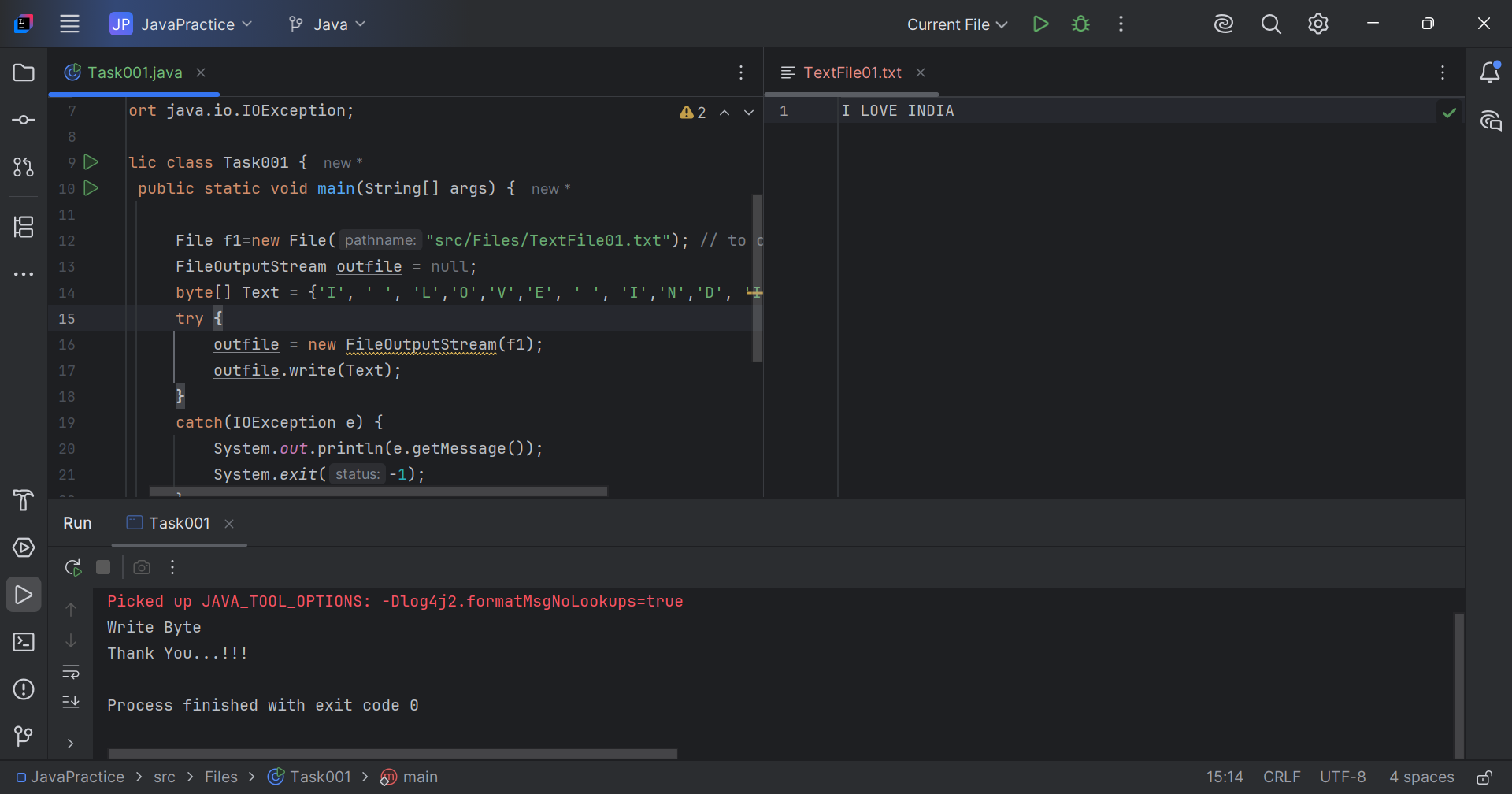
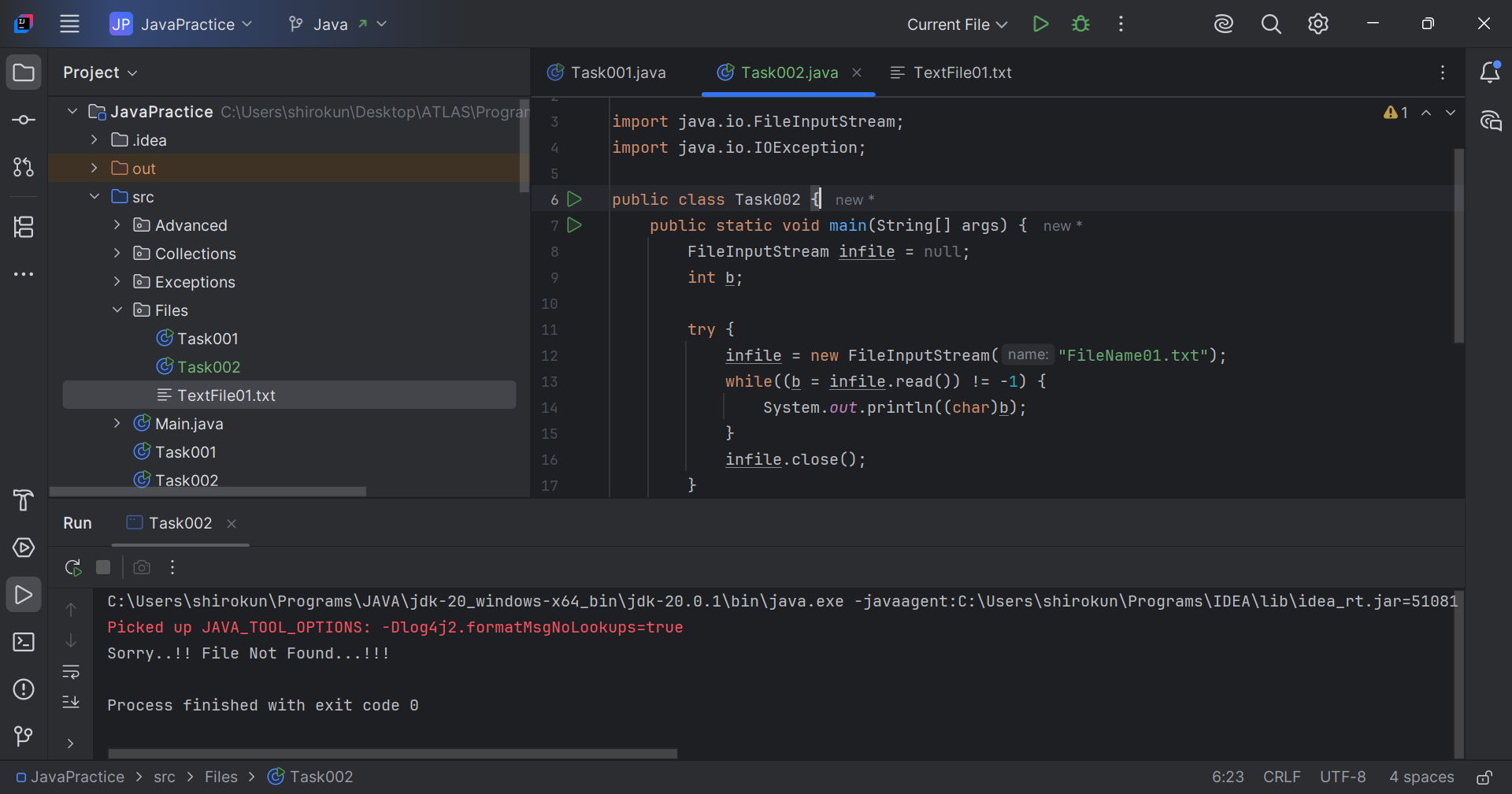
Day 11 – 21/06/2025

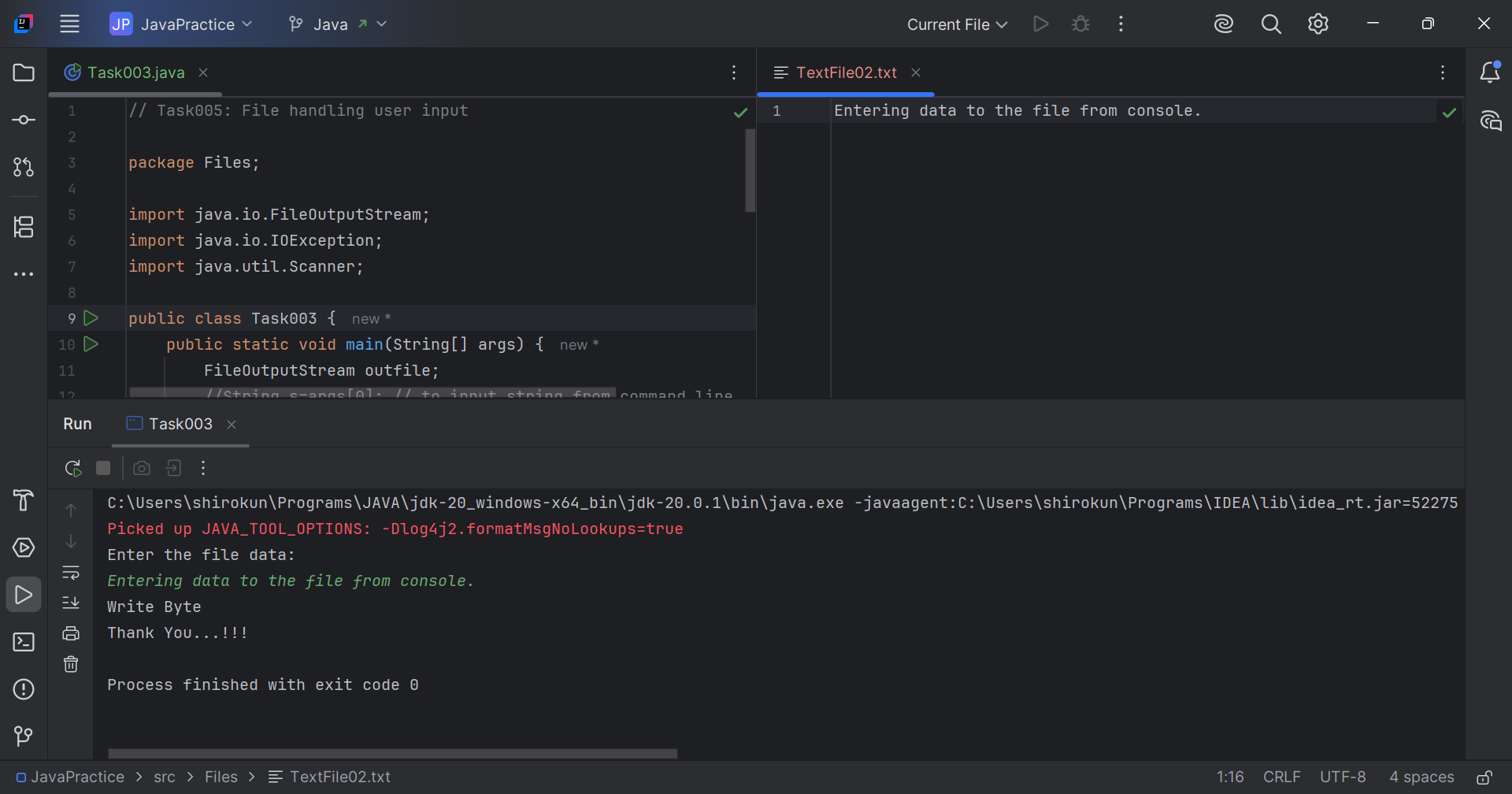
// Task001: File handling.  
  
package Files;  
  
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.IOException;  
  
public class Task001 {  
 public static void main(String[] args) {  
  
 File f1=new File("src/Files/TextFile01.txt"); // to create new file  
 FileOutputStream outfile = null;  
 byte[] Text = {'I', ' ', 'L','O','V','E', ' ', 'I','N','D', 'I', 'A'};  
 try {  
 outfile = new FileOutputStream(f1);  
 outfile.write(Text);  
 }  
 catch(IOException e) {  
 System.*out*.println(e.getMessage());  
 System.*exit*(-1);  
 }  
  
 System.*out*.println("Write Byte");  
 System.*out*.println("Thank You...!!!");  
 }  
}



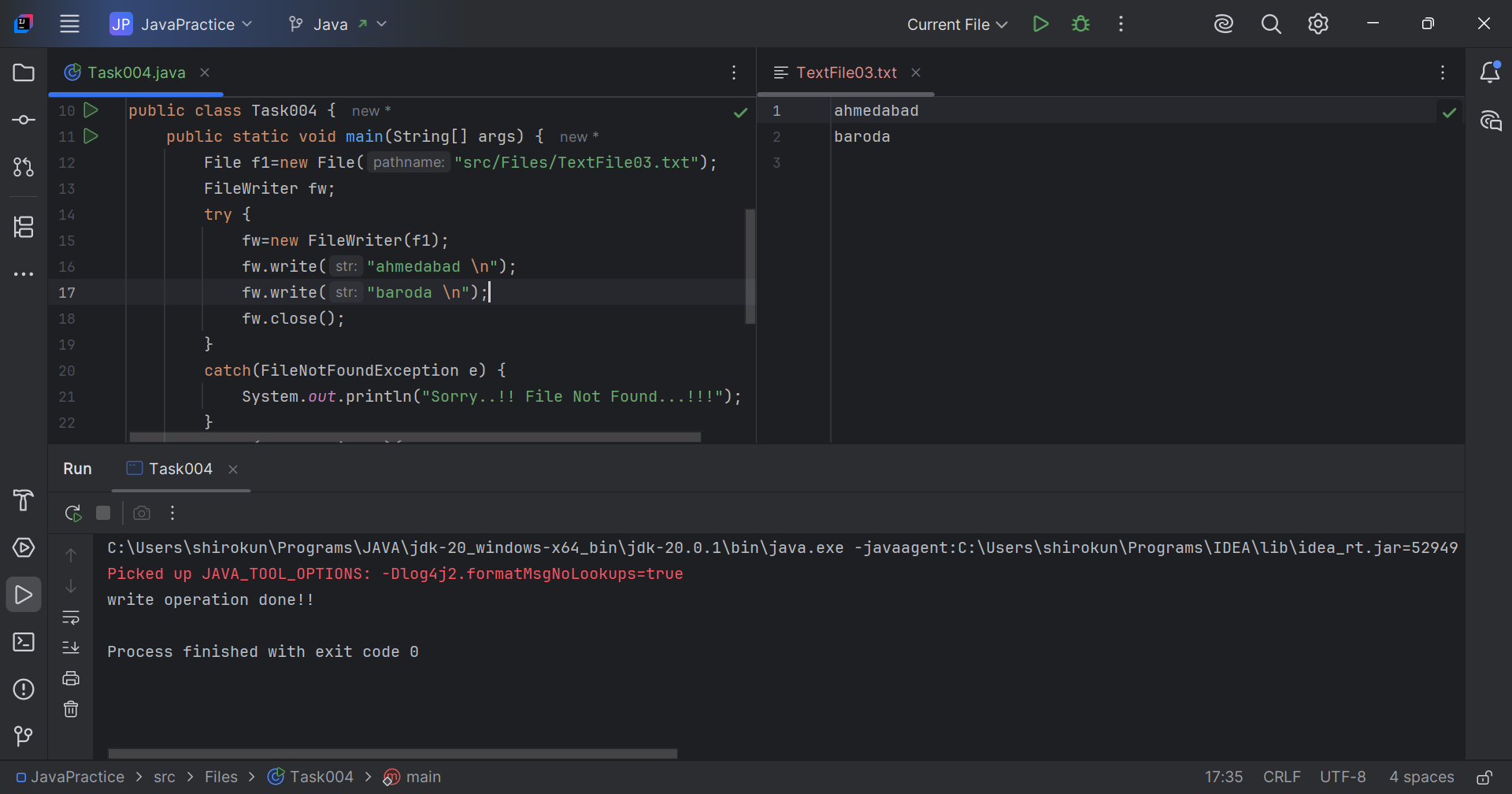
// Task002: File not found  
  
package Files;  
  
import java.io.FileInputStream;  
import java.io.IOException;  
  
public class Task002 {  
 public static void main(String[] args) {  
 FileInputStream infile = null;  
 int b;  
  
 try {  
 infile = new FileInputStream("FileName01.txt");  
 while((b = infile.read()) != -1) {  
 System.*out*.println((char)b);  
 }  
 infile.close();  
 }  
  
 catch(IOException e) {  
 System.*out*.println("Sorry..!! File Not Found...!!!");  
 }  
 }  
}



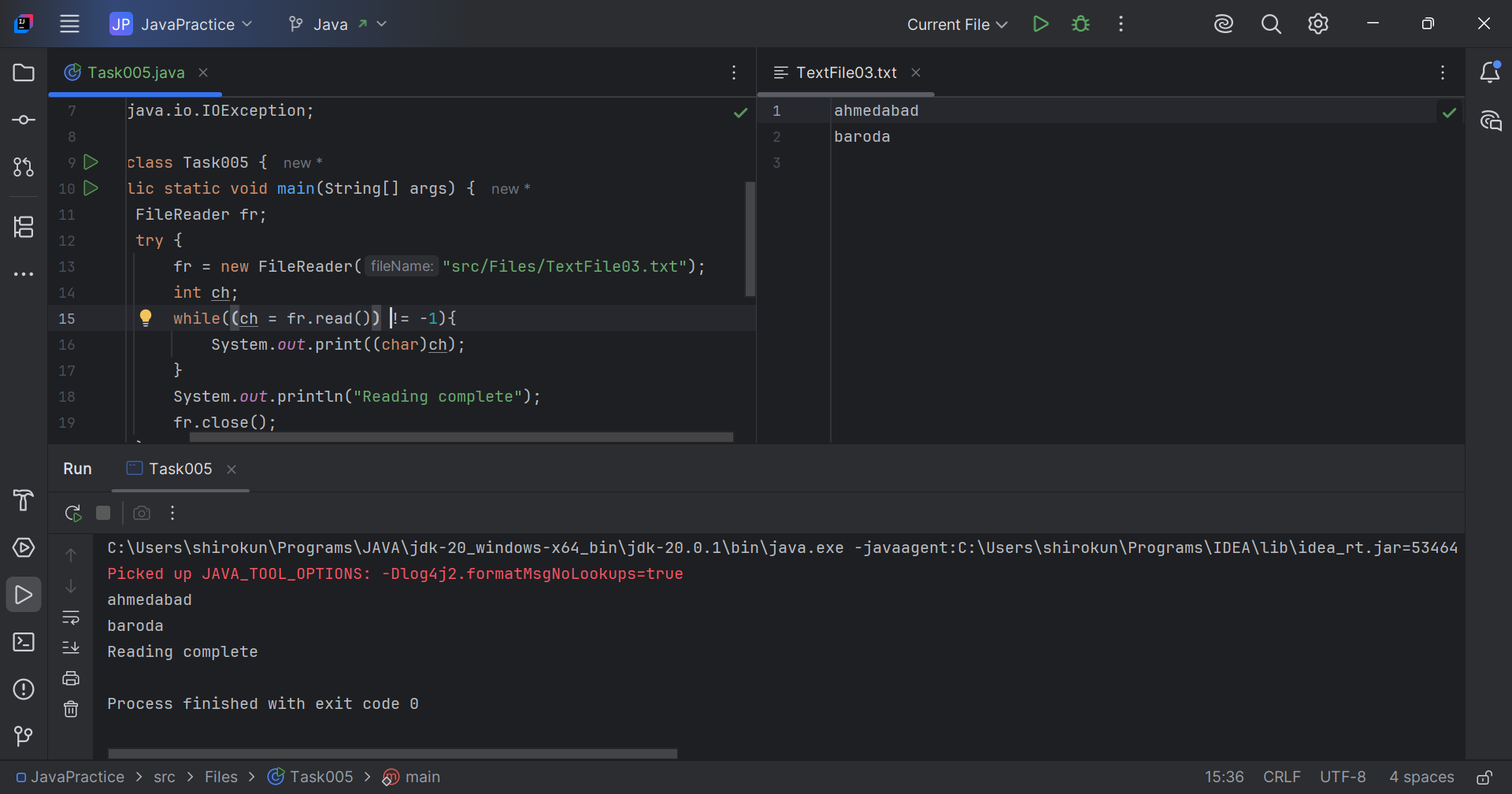
// Task003: File handling user input  
  
package Files;  
  
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.util.Scanner;  
  
public class Task003 {  
 public static void main(String[] args) {  
 FileOutputStream outfile;  
 //String s=args[0]; // to input string from command line  
 Scanner sc=new Scanner(System.*in*);  
 System.*out*.println("Enter the file data: ");  
 String s=sc.nextLine();  
 byte[] b1 = s.getBytes();  
 try{  
 //noinspection resource  
 outfile = new FileOutputStream("src/Files/TextFile02.txt");  
 outfile.write(b1);  
 }  
 catch(IOException e){  
 System.*out*.println(e.getMessage());  
 System.*exit*(-1);  
 }  
  
 System.*out*.println("Write Byte");  
 System.*out*.println("Thank You...!!!");  
 }  
}



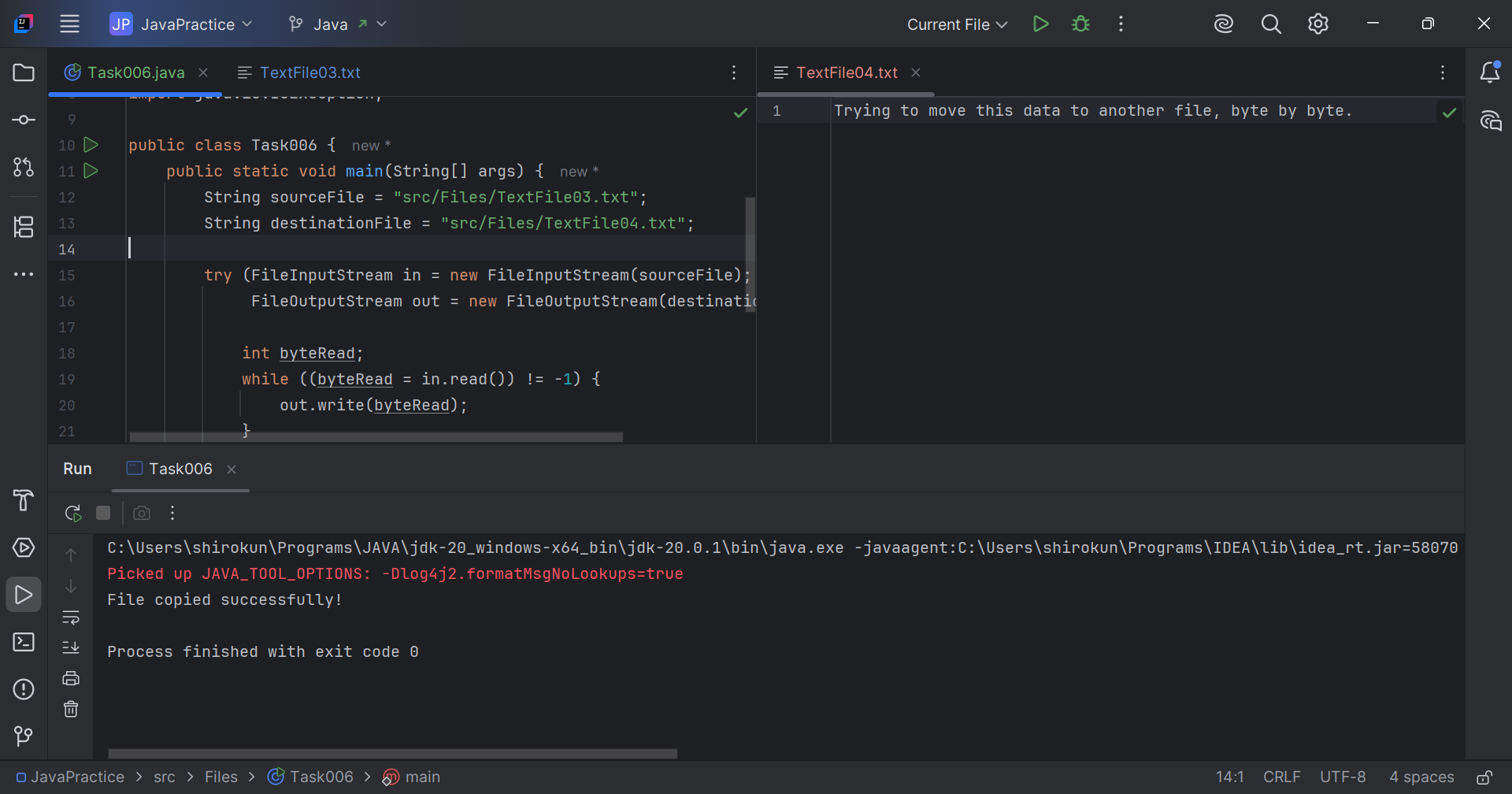
// Task004: Create a file and writes character into that file.  
  
package Files;  
  
import java.io.File;  
import java.io.FileNotFoundException;  
import java.io.FileWriter;  
import java.io.IOException;  
  
public class Task004 {  
 public static void main(String[] args) {  
 File f1=new File("src/Files/TextFile03.txt");  
 FileWriter fw;  
 try {  
 fw=new FileWriter(f1);  
 fw.write("ahmedabad \n");  
 fw.write("baroda \n");  
 fw.close();  
 }  
 catch(FileNotFoundException e) {  
 System.*out*.println("Sorry..!! File Not Found...!!!");  
 }  
 catch(IOException e){  
 System.*out*.println(e.getMessage());  
 }  
  
 System.*out*.println("write operation done!!");  
 }  
}



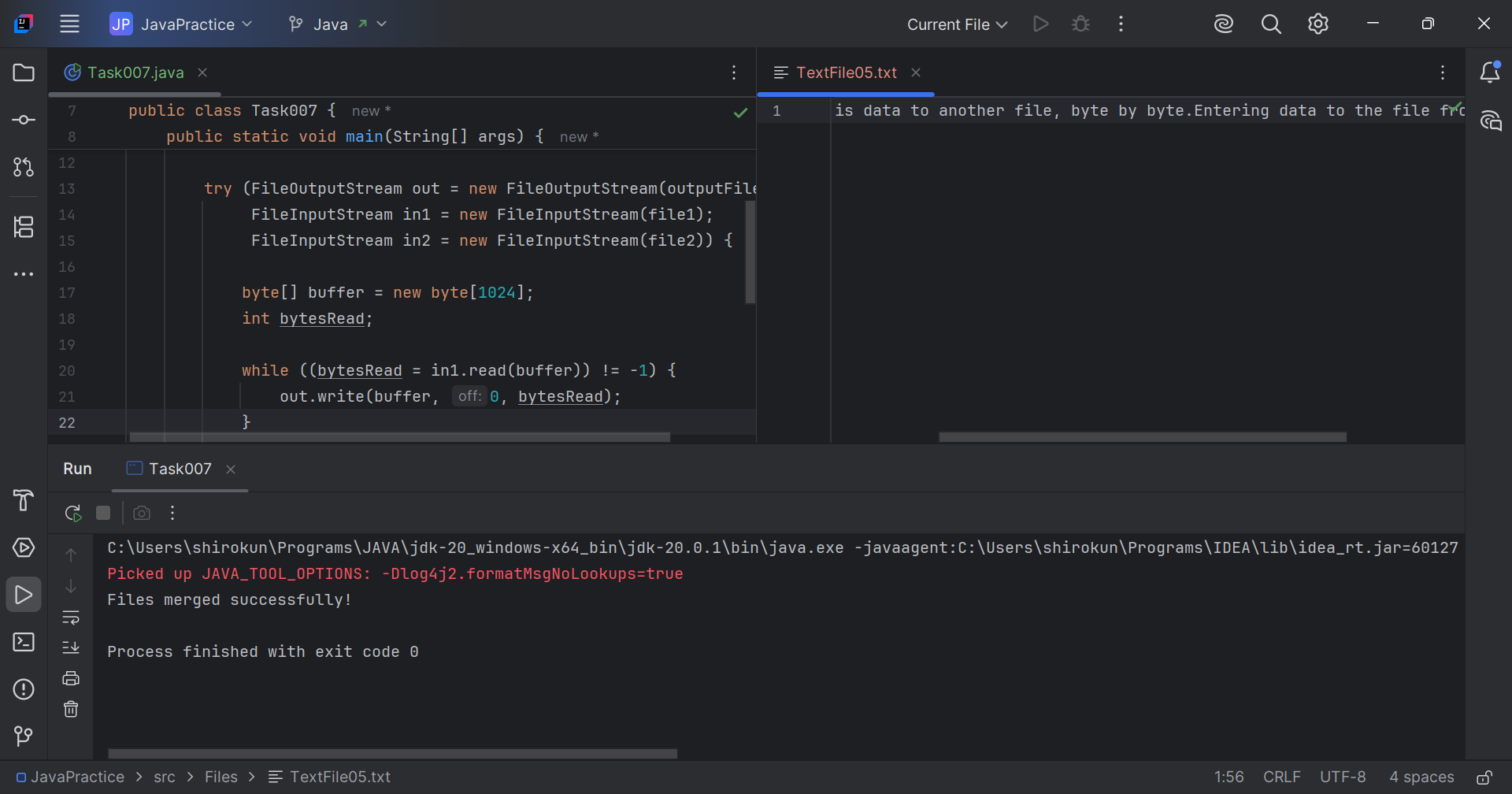
// Task005: Read data from a file.  
  
package Files;  
  
import java.io.FileNotFoundException;  
import java.io.FileReader;  
import java.io.IOException;  
  
public class Task005 {  
 public static void main(String[] args) {  
 FileReader fr;  
 try {  
 fr = new FileReader("src/Files/TextFile03.txt");  
 int ch;  
 while((ch = fr.read()) != -1){  
 System.*out*.print((char)ch);  
 }  
 System.*out*.println("Reading complete");  
 fr.close();  
 }  
 catch(FileNotFoundException e){  
 System.*out*.println("Sorry..!! File Not Found...!!!");  
 }  
  
 catch(IOException e){  
 System.*out*.println(e.getMessage());  
 }  
 }  
}



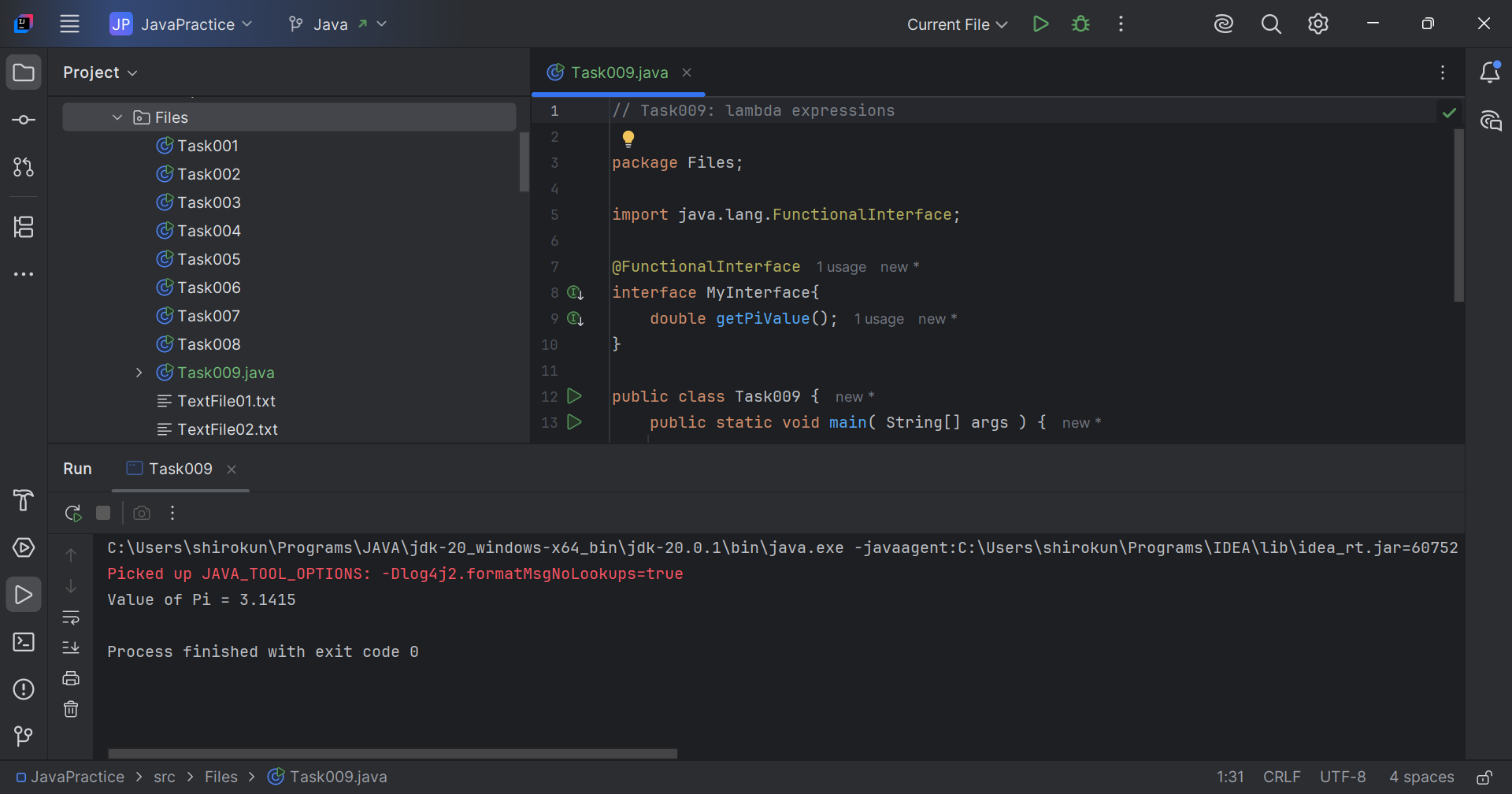
// Task006: Read one byte at a time from a file  
// and copy it into another file immediately.  
  
package Files;  
  
import java.io.FileInputStream;  
import java.io.FileOutputStream;  
import java.io.IOException;  
  
public class Task006 {  
 public static void main(String[] args) {  
 String sourceFile = "src/Files/TextFile03.txt";  
 String destinationFile = "src/Files/TextFile04.txt";  
  
 try (FileInputStream in = new FileInputStream(sourceFile);  
 FileOutputStream out = new FileOutputStream(destinationFile)) {  
  
 int byteRead;  
 while ((byteRead = in.read()) != -1) {  
 out.write(byteRead);  
 }  
  
 System.*out*.println("File copied successfully!");  
  
 } catch (IOException e) {  
 System.*err*.println("Error copying file: " + e.getMessage());  
 }  
 }  
}



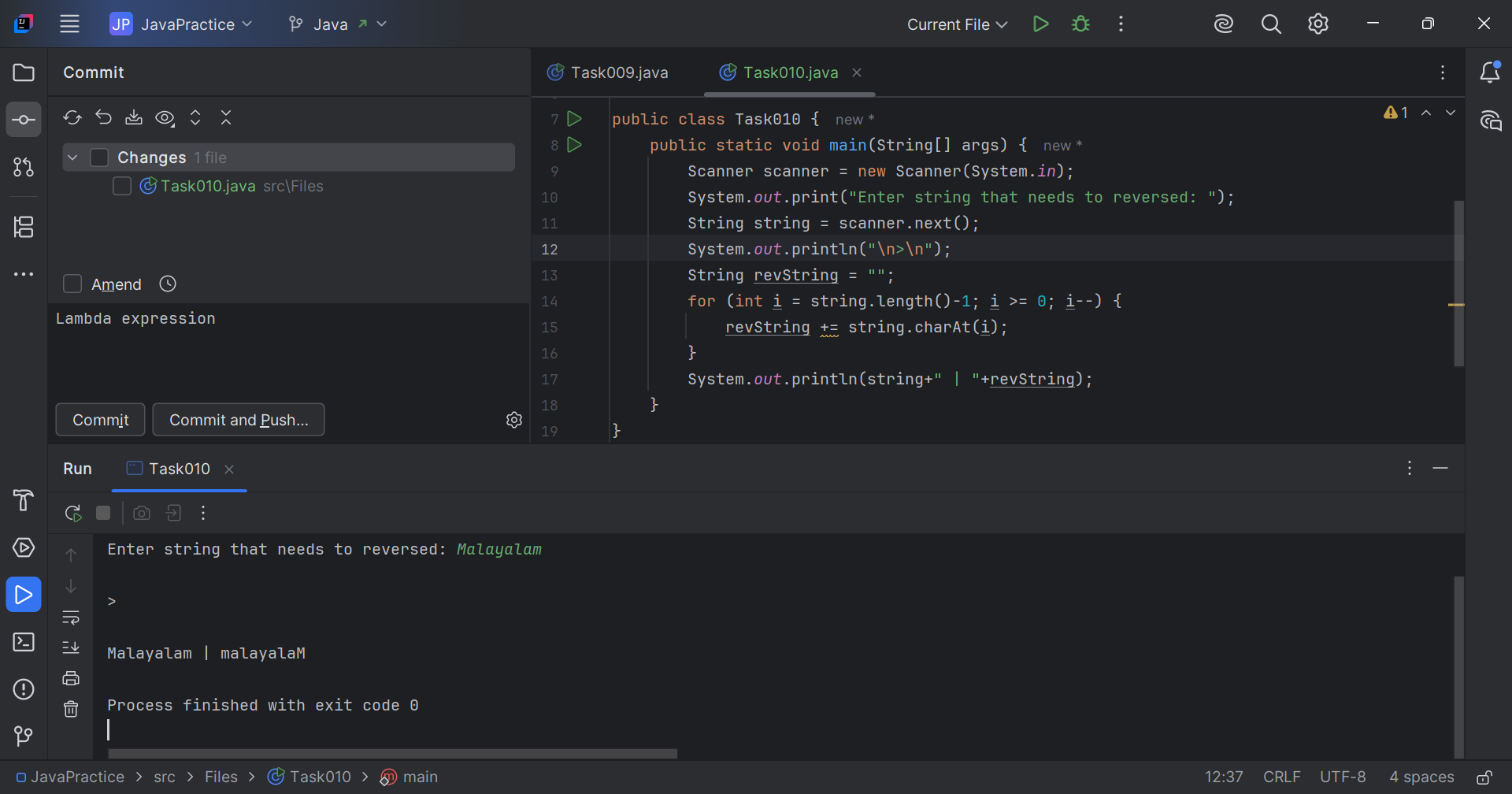
// Task007: Merge two file into a new file.  
  
package Files;  
  
import java.io.\*;  
  
public class Task007 {  
 public static void main(String[] args) {  
 String file1 = "src/Files/TextFile03.txt";  
 String file2 = "src/Files/TextFile02.txt";  
 String outputFile = "src/Files/TextFile05.txt";  
  
 try (FileOutputStream out = new FileOutputStream(outputFile);  
 FileInputStream in1 = new FileInputStream(file1);  
 FileInputStream in2 = new FileInputStream(file2)) {  
  
 byte[] buffer = new byte[1024];  
 int bytesRead;  
  
 while ((bytesRead = in1.read(buffer)) != -1) {  
 out.write(buffer, 0, bytesRead);  
 }  
  
 while ((bytesRead = in2.read(buffer)) != -1) {  
 out.write(buffer, 0, bytesRead);  
 }  
  
 System.*out*.println("Files merged successfully!");  
 } catch(IOException e){  
 System.*out*.println("Sorry..!! File Not Found...!!!");  
 }  
 }  
}



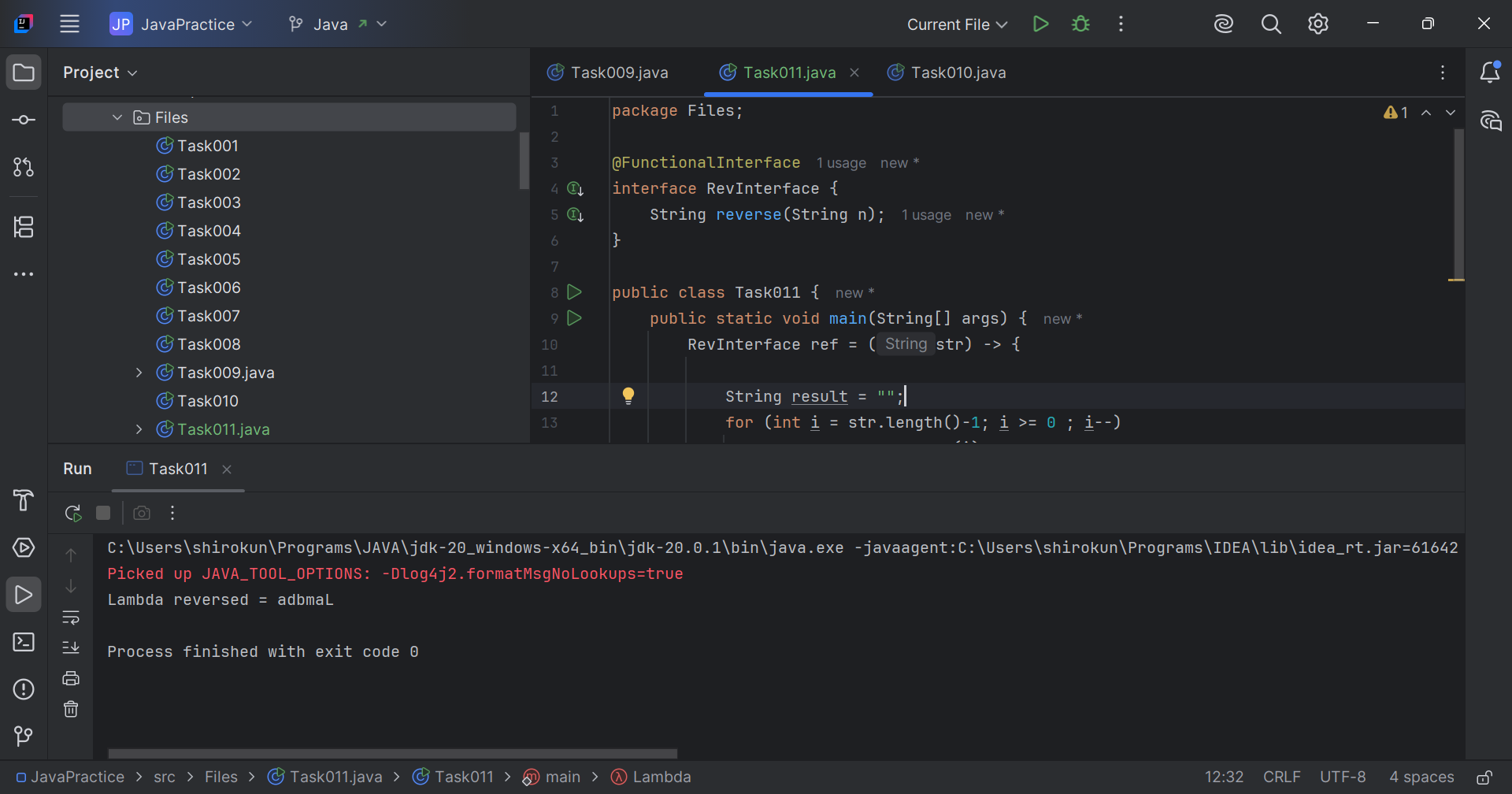
// Task009: lambda expressions  
  
package Files;  
  
import java.lang.FunctionalInterface;  
  
@FunctionalInterface  
interface MyInterface{  
 double getPiValue();  
}  
  
public class Task009 {  
 public static void main( String[] args ) {  
 MyInterface ref;  
 ref = () -> 3.1415; // lambda exp  
 System.*out*.println("Value of Pi = " + ref.getPiValue());  
 }  
}



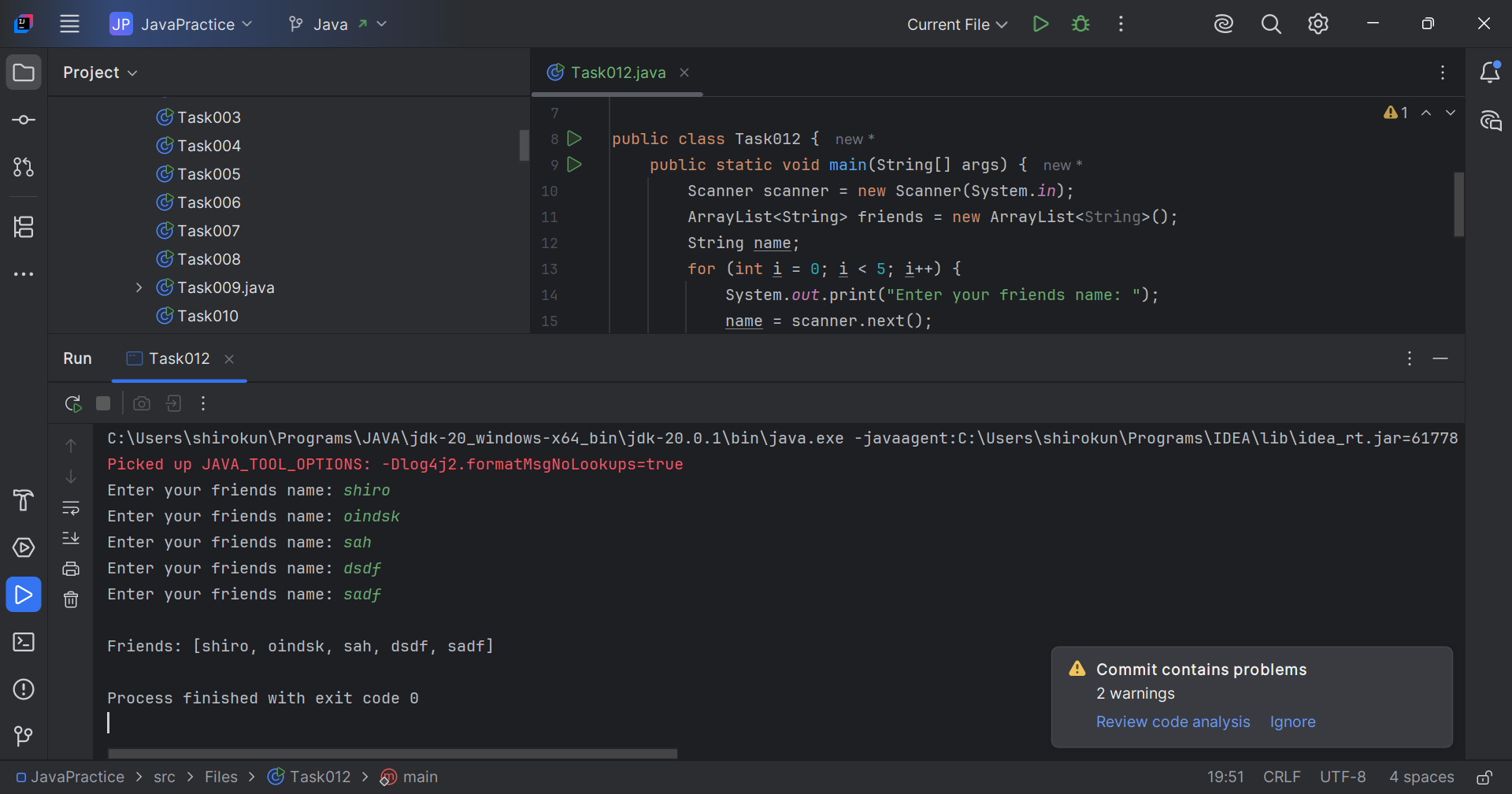
// Task010: Reverse a string.  
  
package Files;  
  
import java.util.Scanner;  
  
public class Task010 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter string that needs to reversed: ");  
 String string = scanner.next();  
 System.*out*.println("\n>\n");  
 String revString = "";  
 for (int i = string.length()-1; i >= 0; i--) {  
 revString += string.charAt(i);  
 }  
 System.*out*.println(string+" | "+revString);  
 }  
}



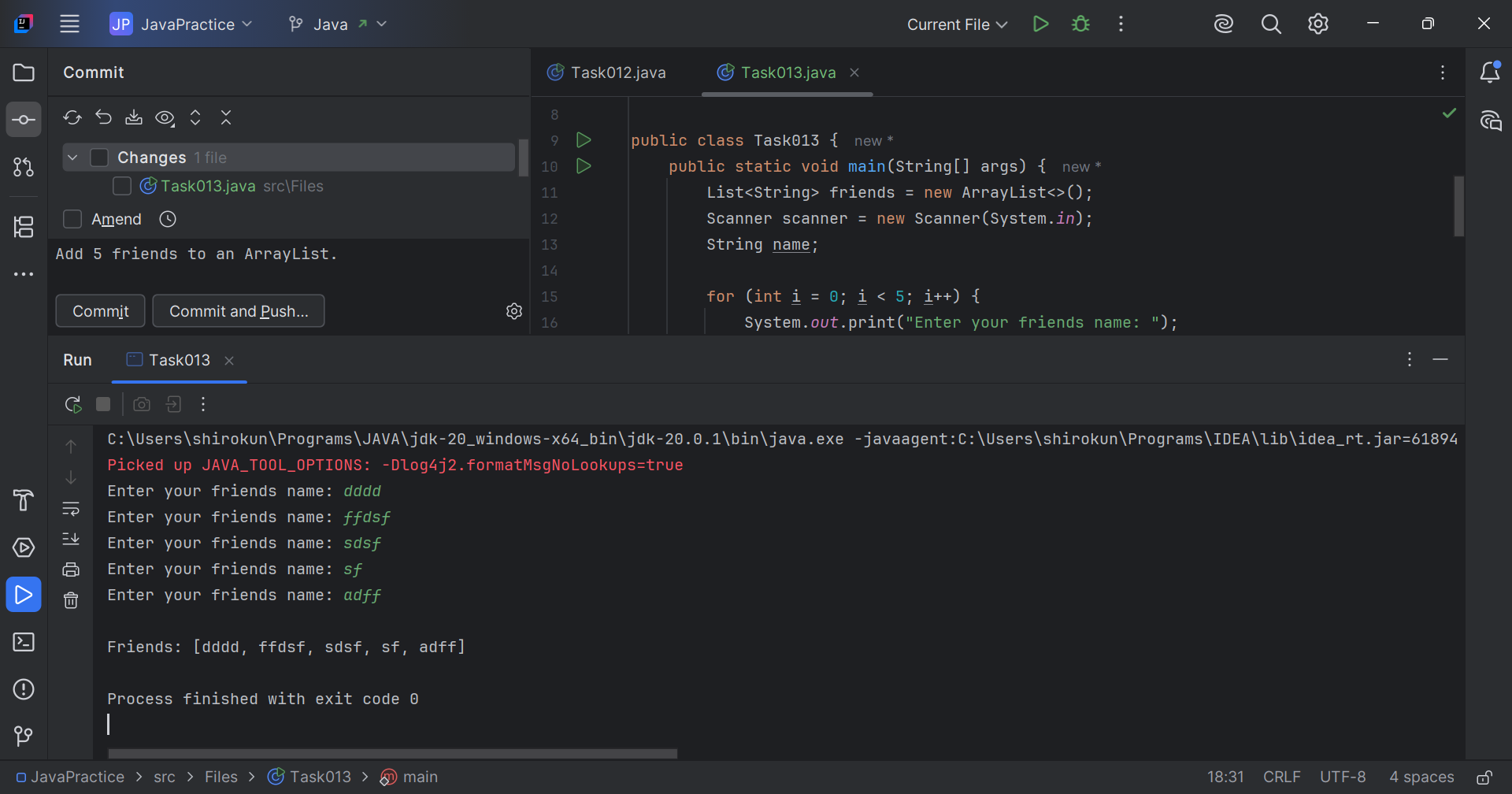
// Task011: Lambda reverse.   
  
package Files;  
  
@FunctionalInterface  
interface RevInterface {  
 String reverse(String n);  
}  
  
public class Task011 {  
 public static void main(String[] args) {  
 RevInterface ref = (str) -> {  
  
 String result = "";  
 for (int i = str.length()-1; i >= 0 ; i--)  
 result += str.charAt(i);  
 return result;  
 };  
  
 System.*out*.println("Lambda reversed = " + ref.reverse("Lambda"));  
  
 }  
}



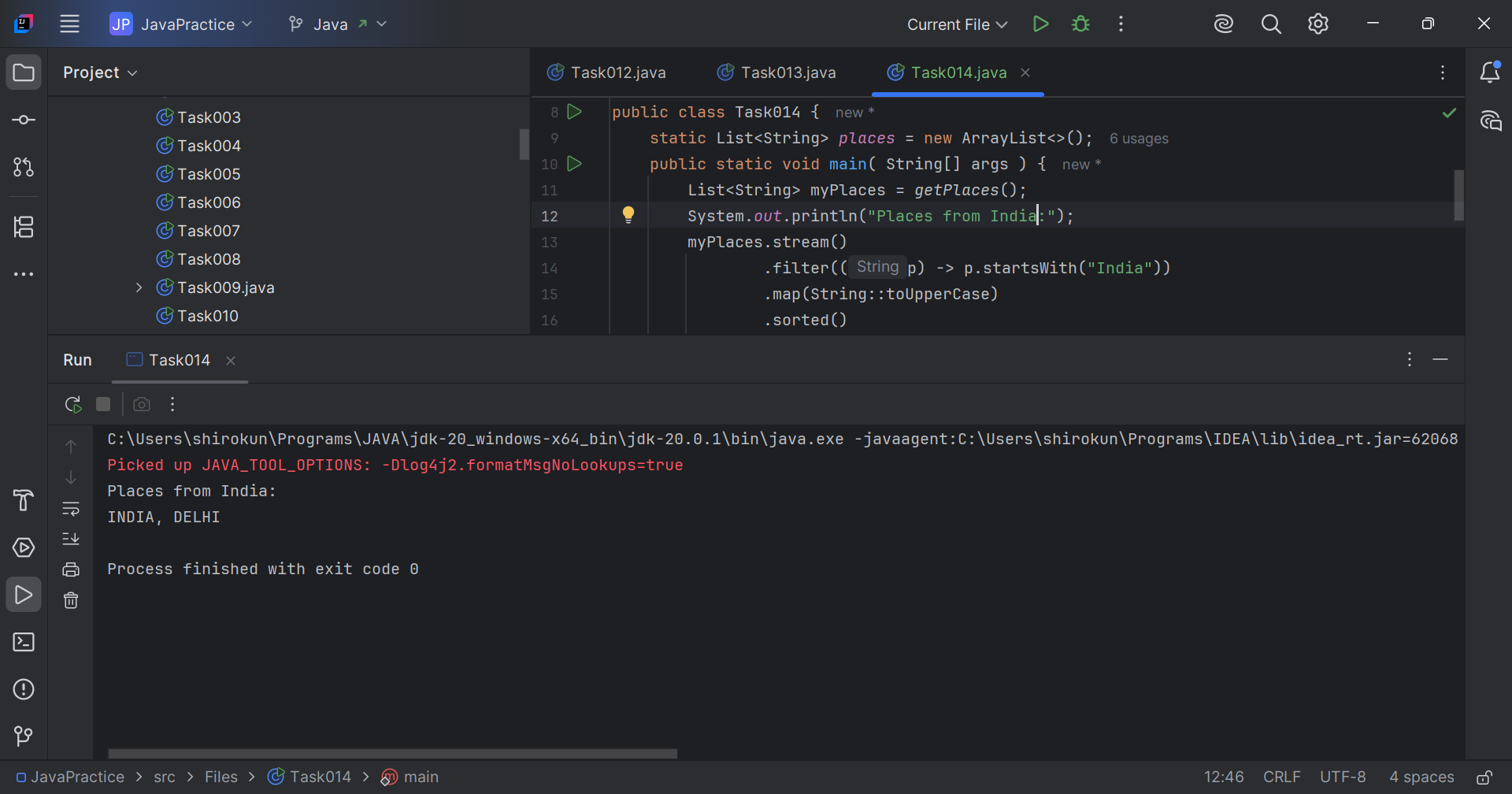
// Task012: Create an ArrayList with 5 friends names.  
  
package Files;  
  
import java.util.ArrayList;  
import java.util.Scanner;  
  
public class Task012 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 ArrayList<String> friends = new ArrayList<String>();  
 String name;  
 for (int i = 0; i < 5; i++) {  
 System.*out*.print("Enter your friends name: ");  
 name = scanner.next();  
 friends.add(name);  
 }  
  
 System.*out*.println("\nFriends: "+friends);  
 scanner.close();  
 }  
}



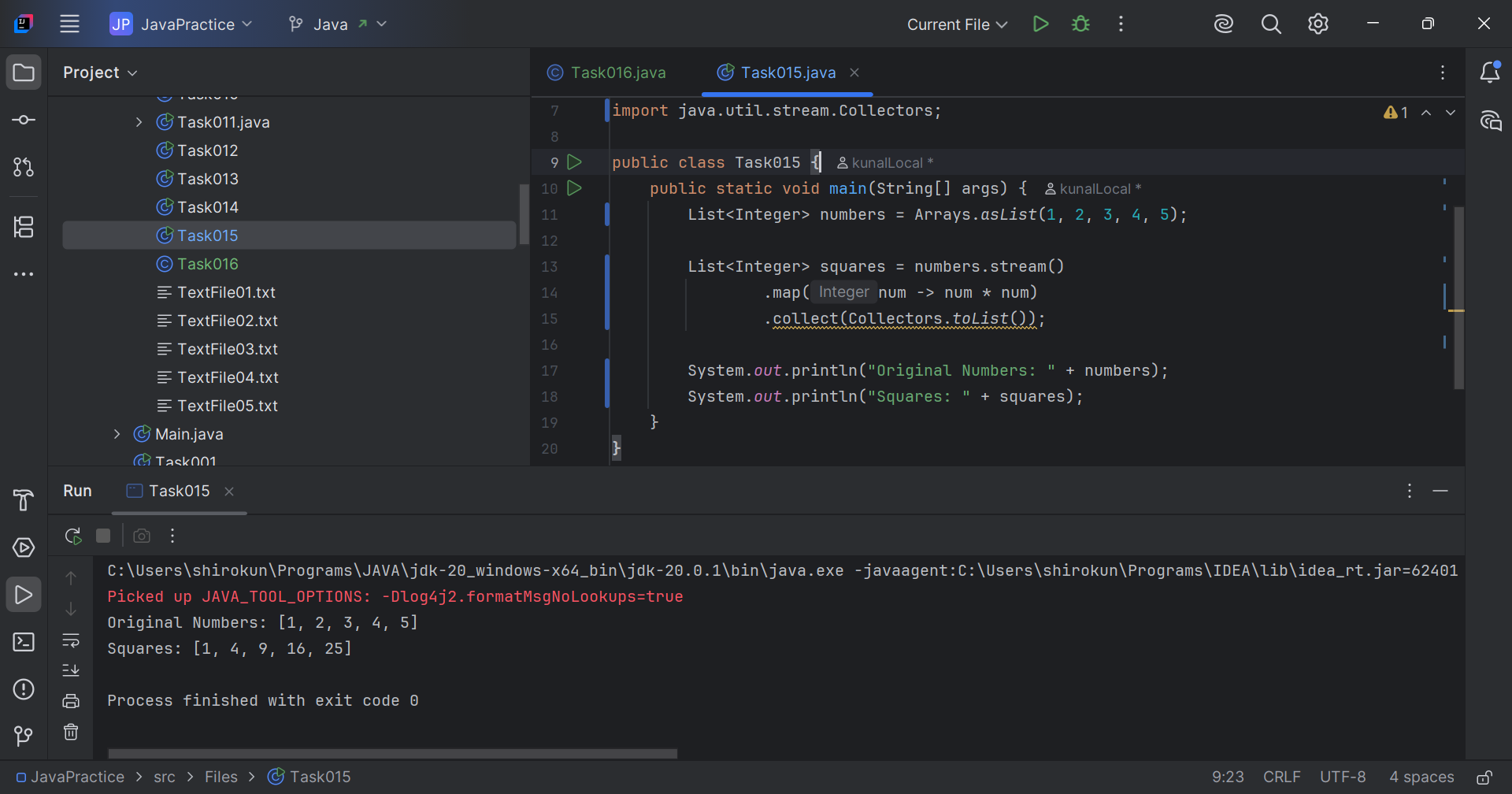
// Task012: Create a List with 5 friends names.  
  
package Files;  
  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Scanner;  
  
public class Task013 {  
 public static void main(String[] args) {  
 List<String> friends = new ArrayList<>();  
 Scanner scanner = new Scanner(System.*in*);  
 String name;  
  
 for (int i = 0; i < 5; i++) {  
 System.*out*.print("Enter your friends name: ");  
 name = scanner.next();  
 friends.add(name);  
 }  
  
 System.*out*.println("\nFriends: "+friends);  
 scanner.close();  
 }  
}



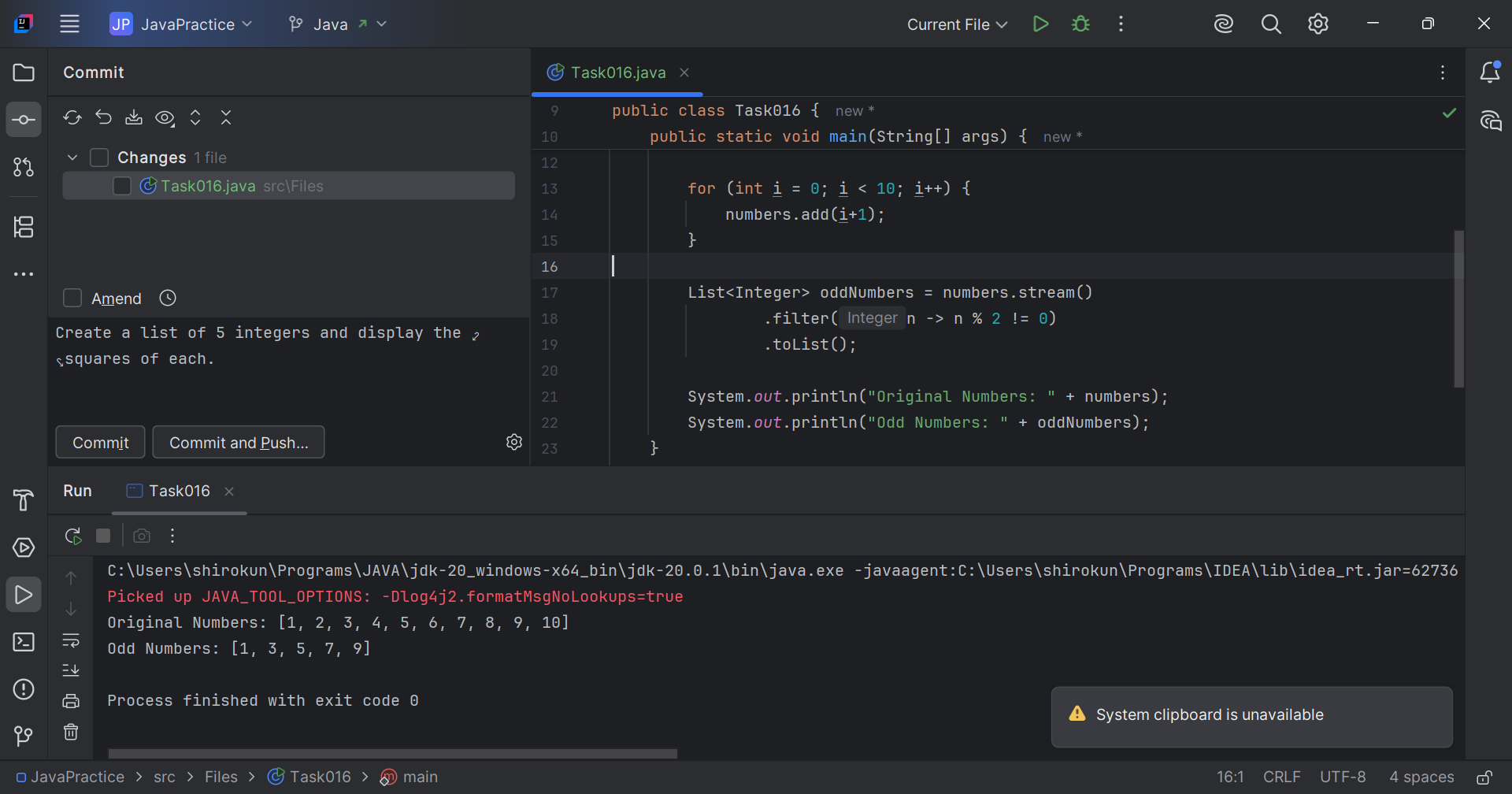
// Task014: Streams filter.  
  
package Files;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class Task014 {  
 static List<String> *places* = new ArrayList<>();  
 public static void main( String[] args ) {  
 List<String> myPlaces = *getPlaces*();  
 System.*out*.println("Places from India:");  
 myPlaces.stream()  
 .filter((p) -> p.startsWith("India"))  
 .map(String::toUpperCase)  
 .sorted()  
 .forEach(System.*out*::println);  
 }  
  
 public static List<String> getPlaces(){  
 *places*.add("Nepal, Kathmandu");  
 *places*.add("Nepal, Pokhara");  
 *places*.add("India, Delhi");  
 *places*.add("USA, New York");  
 *places*.add("Africa, Nigeria");  
 return *places*;  
 }  
}



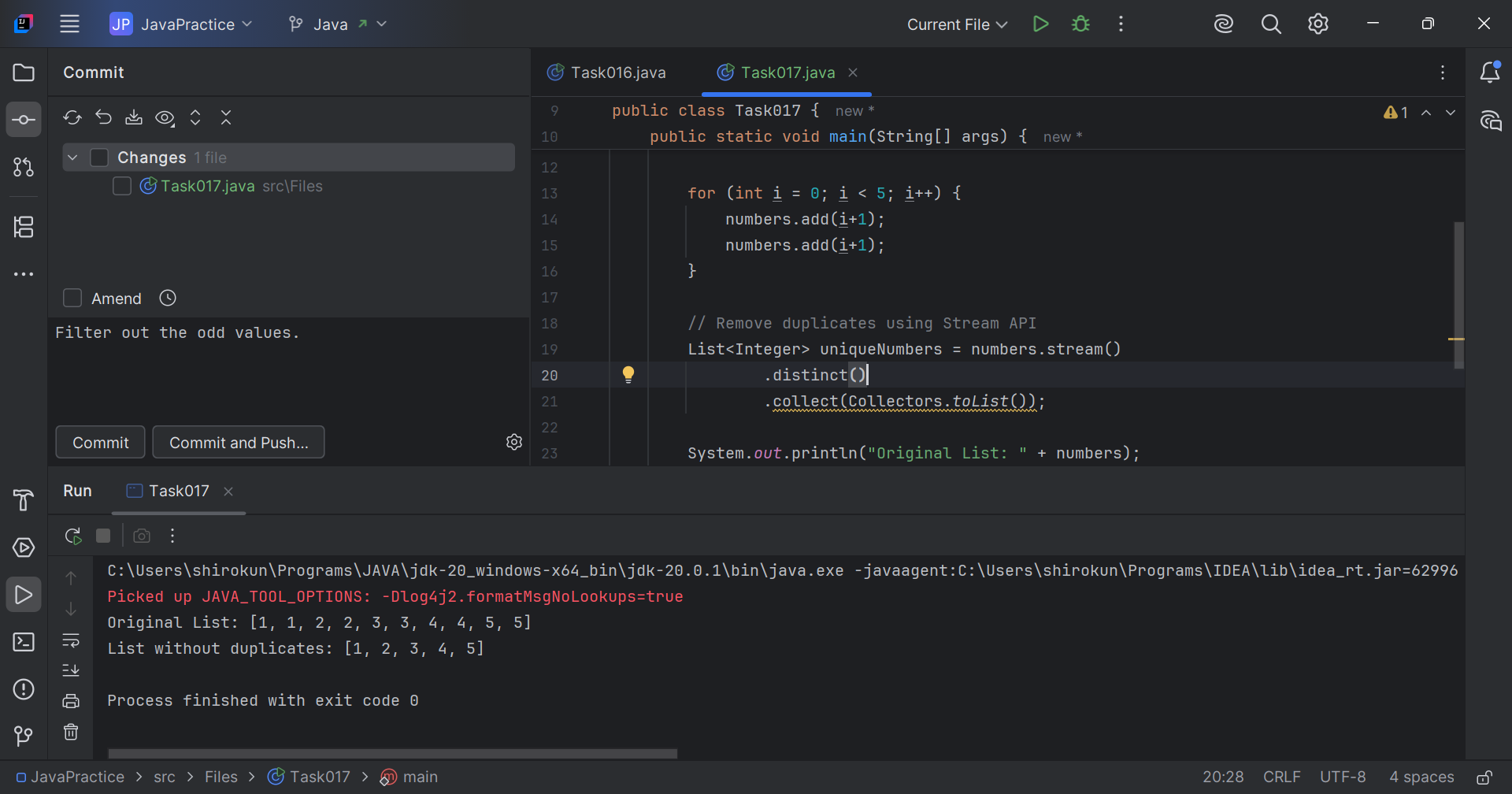
// Task015: Create a list of 5 integers and display the squares of each.  
  
package Files;  
  
import java.util.Arrays;  
import java.util.List;  
  
  
public class Task015 {  
 public static void main(String[] args) {  
 List<Integer> numbers = Arrays.*asList*(1, 2, 3, 4, 5);  
  
 List<Integer> squares = numbers.stream()  
 .map(num -> num \* num)  
 .toList();  
  
 System.*out*.println("Original Numbers: " + numbers);  
 System.*out*.println("Squares: " + squares);  
 }  
}



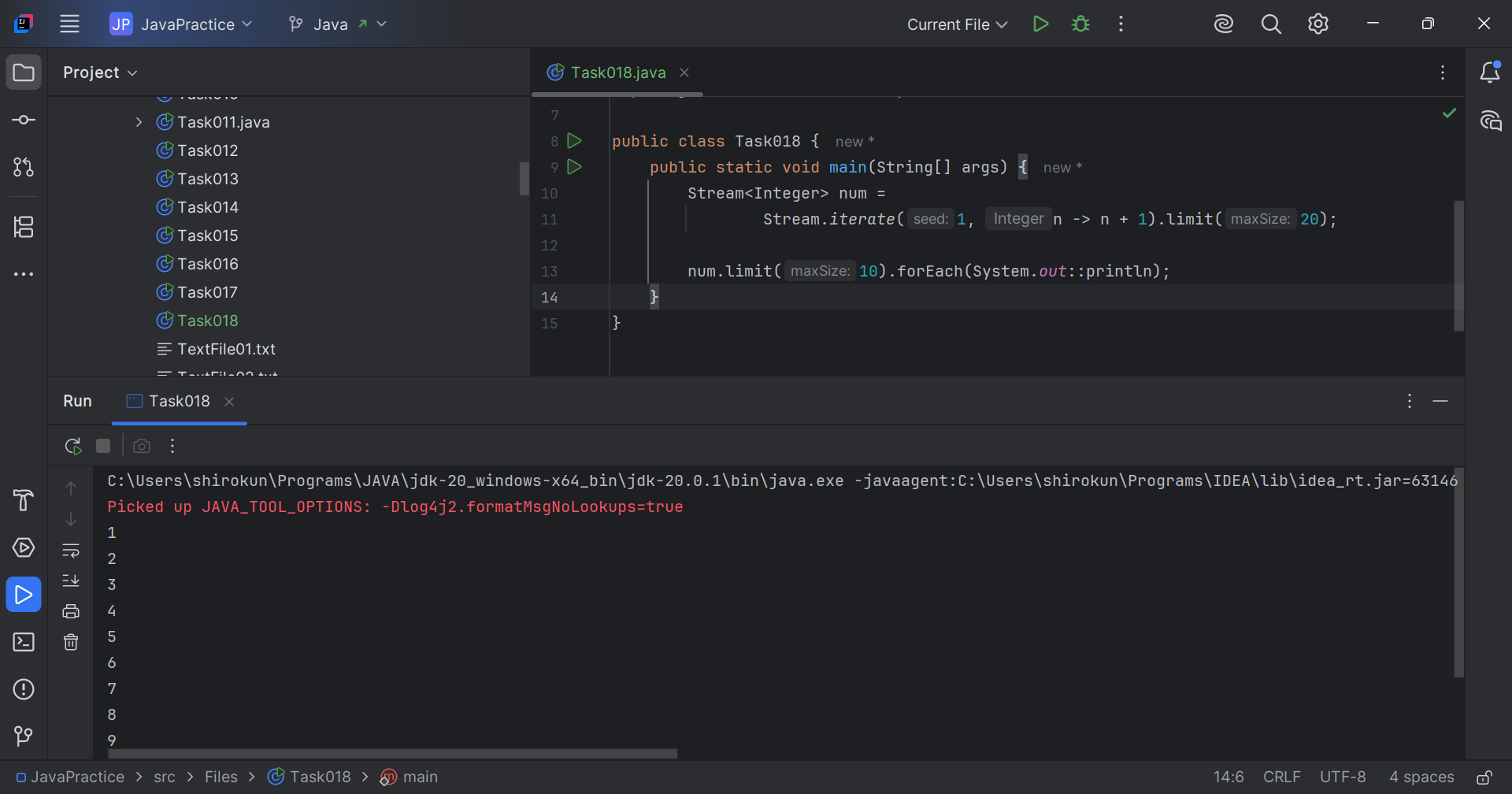
// Task016: Create an array list and filter the  
// values which are odd numbers and display  
  
package Files;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class Task016 {  
 public static void main(String[] args) {  
 ArrayList<Integer> numbers = new ArrayList<>();  
  
 for (int i = 0; i < 10; i++) {  
 numbers.add(i+1);  
 }  
  
 List<Integer> oddNumbers = numbers.stream()  
 .filter(n -> n % 2 != 0)  
 .toList();  
  
 System.*out*.println("Original Numbers: " + numbers);  
 System.*out*.println("Odd Numbers: " + oddNumbers);  
 }  
}



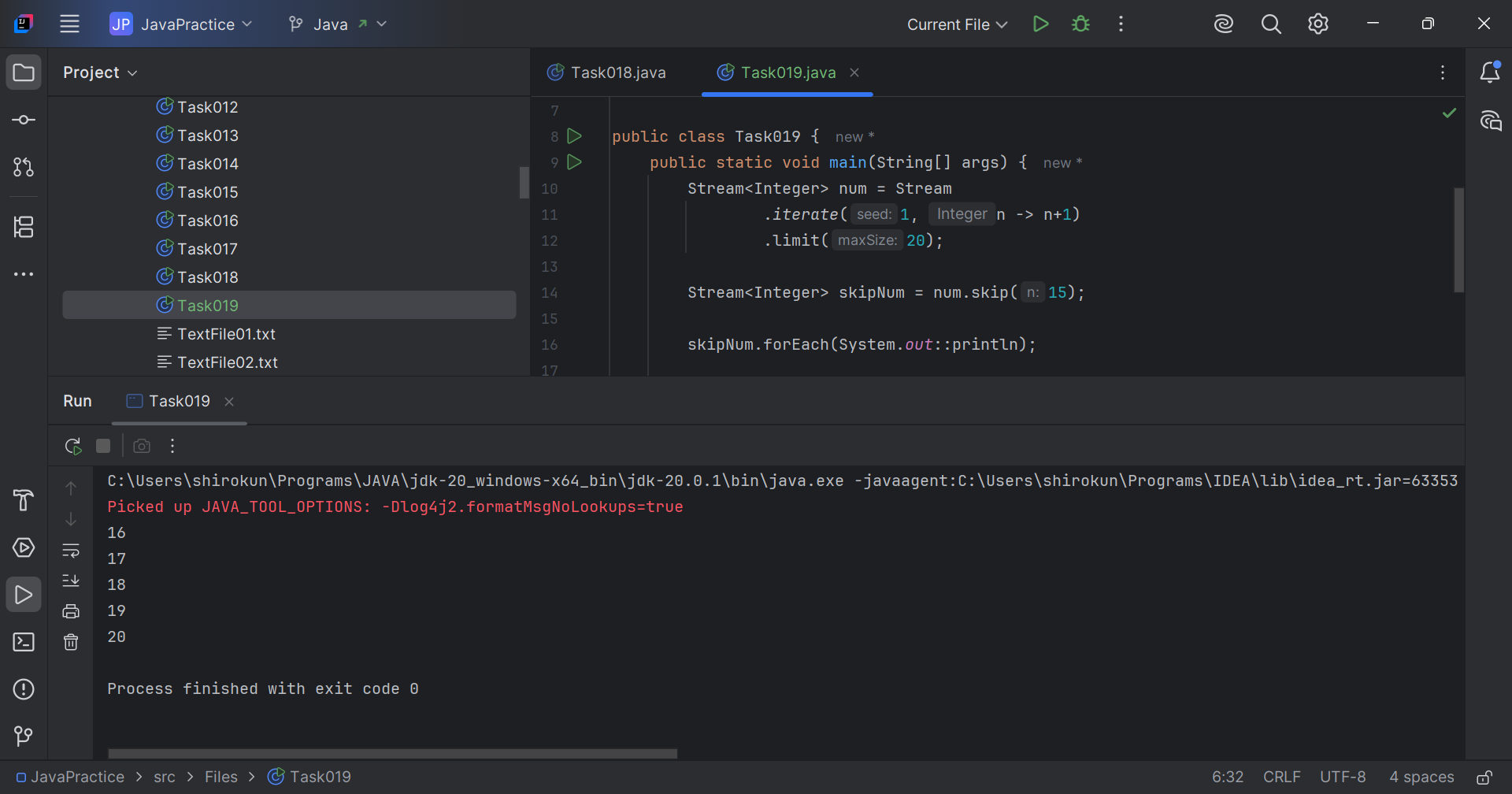
// Task017: Remove duplicates.  
  
package Files;  
  
import java.util.ArrayList;  
import java.util.List;  
import java.util.stream.Collectors;  
  
public class Task017 {  
 public static void main(String[] args) {  
 ArrayList<Integer> numbers = new ArrayList<>();  
  
 for (int i = 0; i < 5; i++) {  
 numbers.add(i+1);  
 numbers.add(i+1);  
 }  
  
 // Remove duplicates using Stream API  
 List<Integer> uniqueNumbers = numbers.stream()  
 .distinct()  
 .collect(Collectors.*toList*());  
  
 System.*out*.println("Original List: " + numbers);  
 System.*out*.println("List without duplicates: " + uniqueNumbers);  
 }  
}



// Task018: WAP to run a loop / iterate() and limit it to 20 values (1 to 2)  
//While displaying use for each to limit till 10 numbers.  
  
package Files;  
  
import java.util.stream.Stream;  
  
public class Task018 {  
 public static void main(String[] args) {  
 Stream<Integer> num =  
 Stream.*iterate*(1, n -> n + 1).limit(20);  
  
 num.limit(10).forEach(System.*out*::println);  
 }  
}



// Task019: Create an array List skip 15 numbers  
// and print the output using foreach loop.  
  
package Files;  
  
import java.util.stream.Stream;  
  
public class Task019 {  
 public static void main(String[] args) {  
 Stream<Integer> num = Stream  
 .*iterate*(1, n -> n+1)  
 .limit(20);  
  
 Stream<Integer> skipNum = num.skip(15);  
  
 skipNum.forEach(System.*out*::println);  
  
 }  
}



// Task020: reduce()  
  
package Files;  
  
import java.util.Arrays;  
import java.util.List;  
import java.util.Optional;  
  
public class Task020 {  
 public static void main(String[] args) {  
 List<Integer> numbers = Arrays.*asList*(1, 2, 3, 4, 5);  
   
 Optional<Integer> sum = numbers.stream().reduce(Integer::*sum*);  
 System.*out*.println("Sum of all elements: " + sum.orElse(0));  
  
 Optional<Integer> max = numbers.stream().reduce(Integer::*max*);  
 System.*out*.println("Maximum element: " + max.orElse(0));  
  
 List<String> strings = Arrays.*asList*("Hello", " ", "world", "!");  
 Optional<String> concatenatedString = strings.stream().reduce((x, y) -> x + y);  
   
 System.*out*.println("Concatenated string: " + concatenatedString.orElse(""));  
  
 }  
}

