Day 4:

Task 1: Array Sorting and Searching

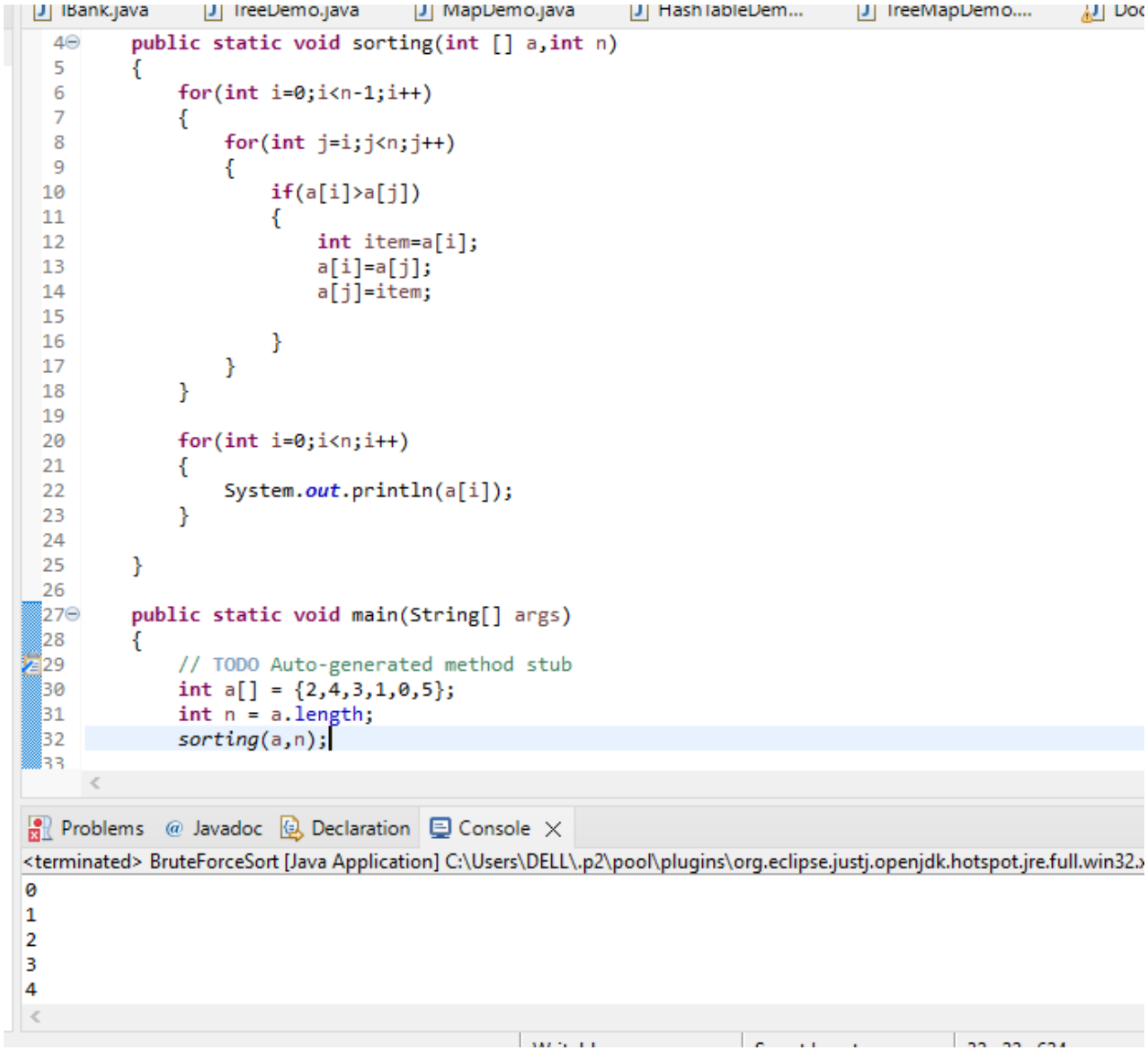
a) Implement a function called BruteForceSort that sorts an array using the brute force approach. Use this function to sort an array created with InitializeArray.

b) Write a function named PerformLinearSearch that searches for a specific element in an array and returns the index of the element if found or -1 if not found.

Bruit Force Algorithum:

-> is a straightforward problem solving approach that tries all possible solution to find the correct solution.

Implementation a) :



Linear Search : if searching one by one elements;

```java
package Array;

public class Linear_Search {
    public static int  search(int []a,int n)
    {

        for (int i = 0; i < a.length; i++)
        {
              if (a[i] == n)
              {
                    return i;
              }
        }
        return -1;
        }


    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int [] a= {1,2,4,5,3};

        int n=2;

        int result=search(a,n);
        if(result !=1)
        {
            System.out.println("Not Found");
        }
        else
        {
            System.out.println("Found");

```

Problems @ Javadoc Declaration Console

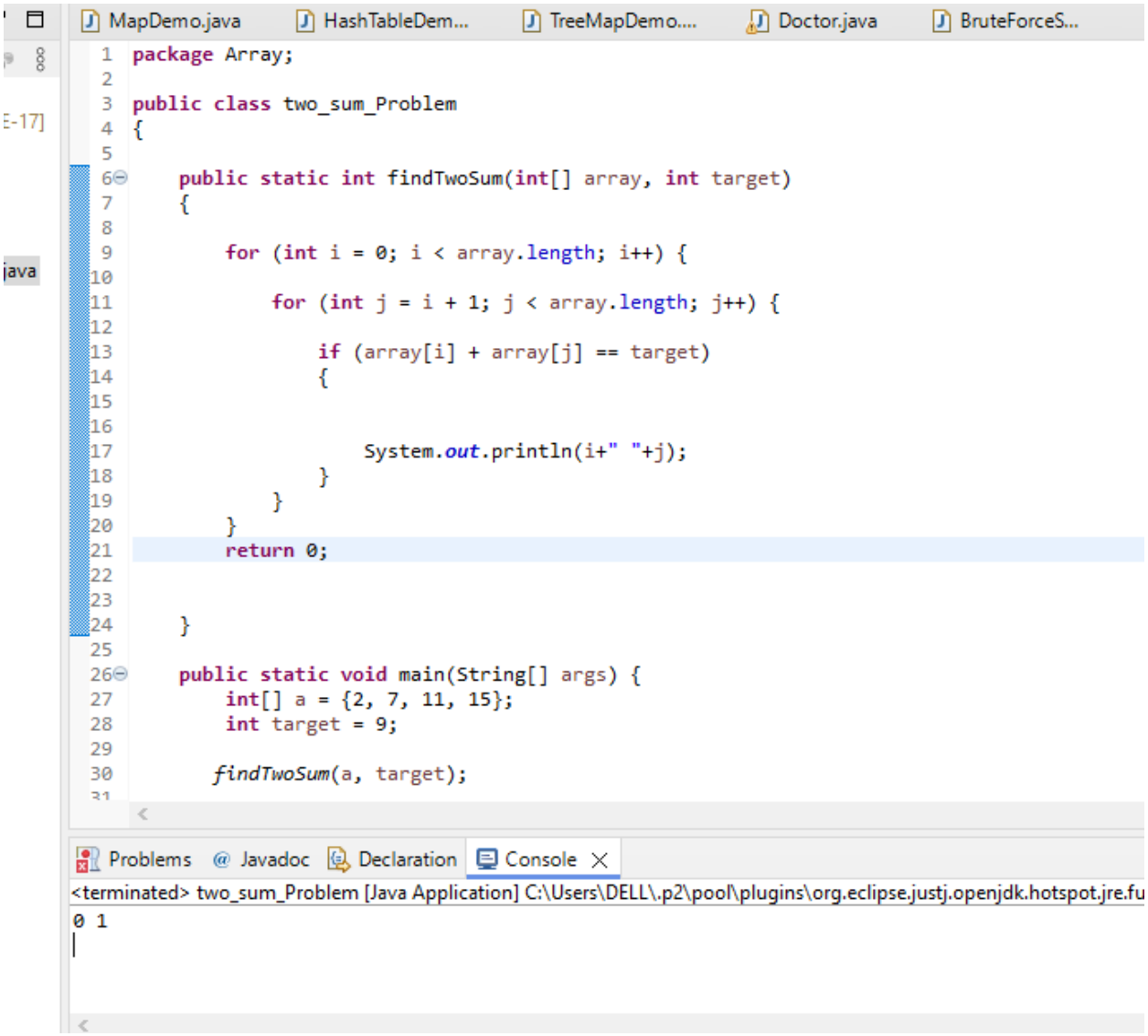
<terminated> Linear_Search [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.hots

Found

Task 2: Two-Sum Problem

a) Given an array of integers, write a program that finds if there are two numbers that add up to a specific target. You may assume that each input would have exactly one solution, and you may not use the same element twice. Optimize the solution for time complexity.

Below are implementation of two-sum problem:



Task 3: Understanding Functions through Arrays

a) Write a recursive function named SumArray that calculates and returns the sum of elements in an array, demonstarte with example.

- Fun called itself is called recursive function:

```java
package Array;

public class Recurtion {
    public static int SumArray(int[] array, int index)
    {
        if (index == array.length - 1)
        {
            return array[index];
        }
        return array[index] + SumArray(array, index + 1);
    }

    public static void main(String[] args) {
        int[] arr = {12, 3, 4, 15};
        int n = arr.length;
        System.out.println("Sum of given array is " + SumArray(arr, 0));
    }

}
```

Problems @ Javadoc Declaration Console

terminated> Recurtion [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32

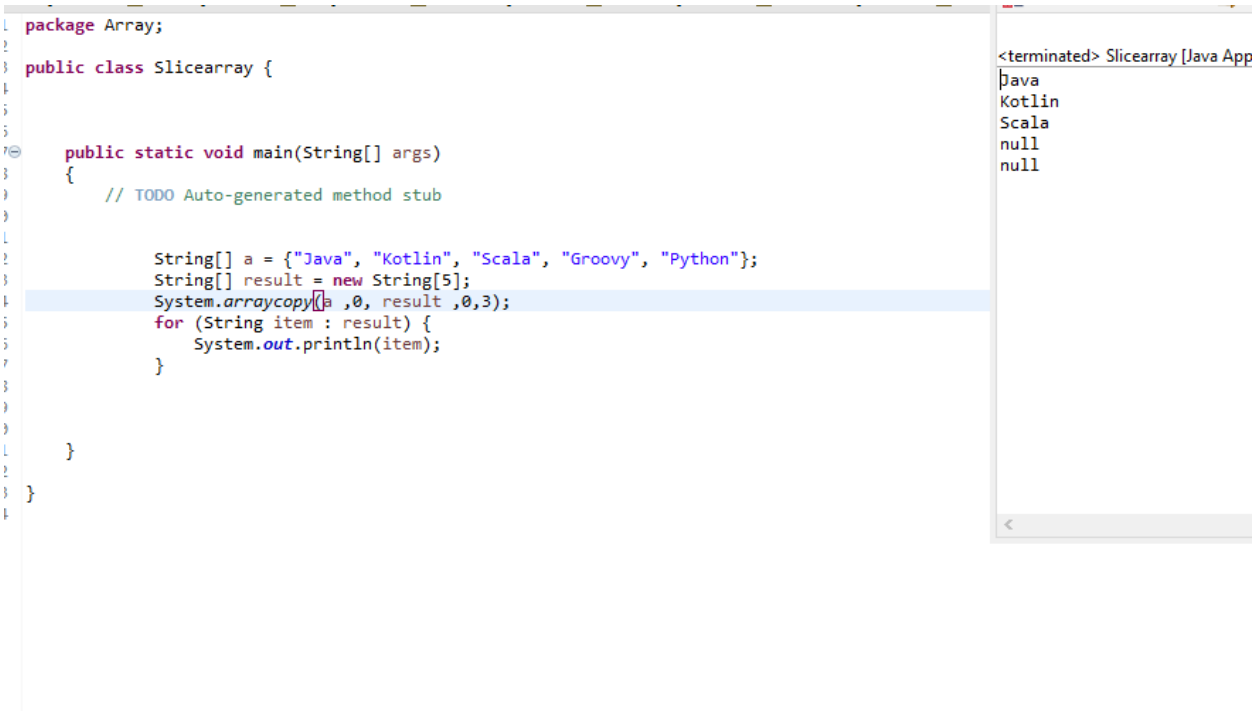
Sum of given array is 34

Task 4: Advanced Array Operations

a) Implement a method SliceArray that takes an array, a starting index, and an end index, then returns a new array containing the elements from the start to the end index.

b) Create a recursive function to find the nth element of a Fibonacci sequence and store the first n elements in an array.

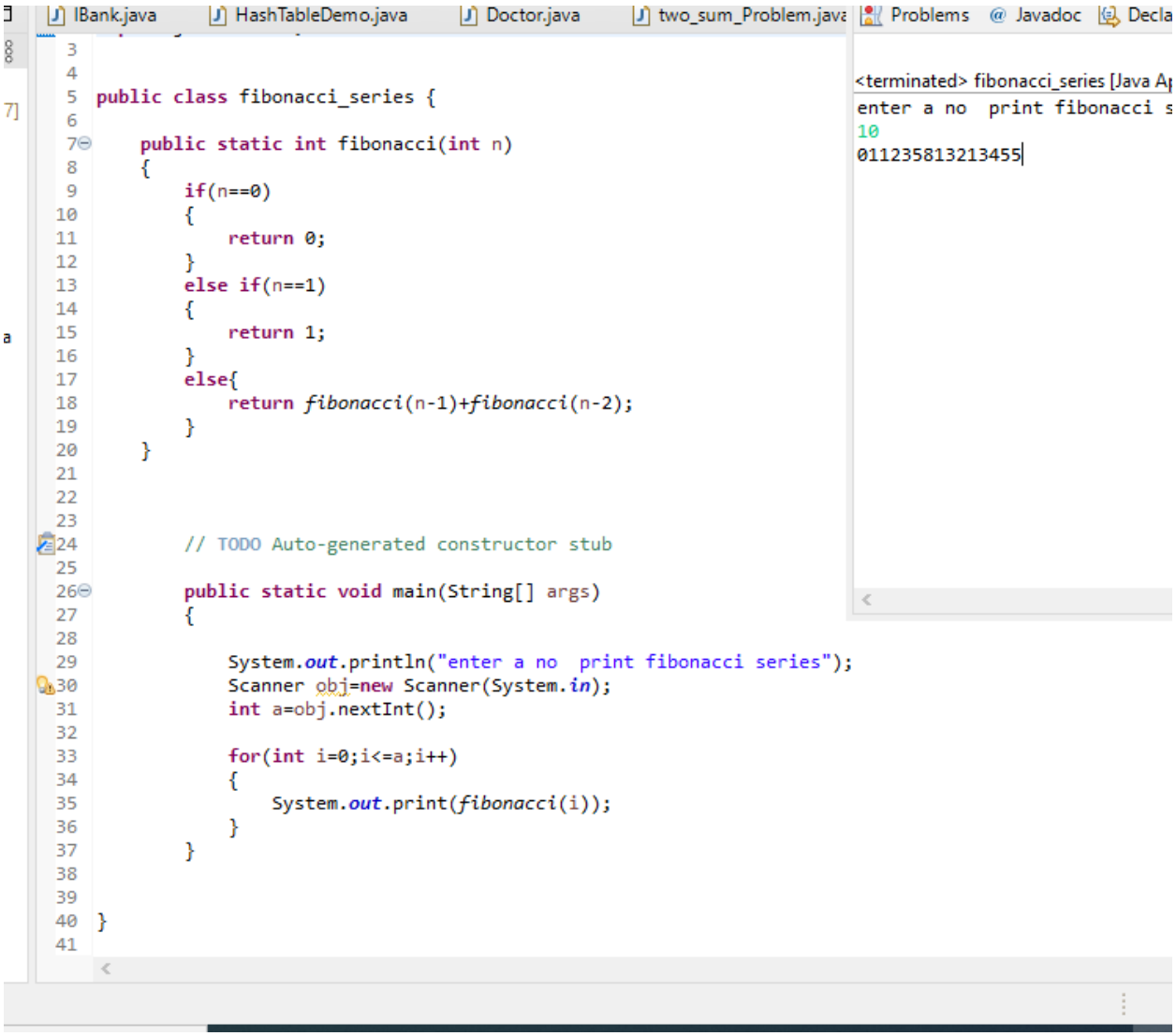
public static void arraycopy(Object Source, int sourceStartInd Object Destination, int DestinationStartIndex, int size)



```
package Array;

public class Slicearray {



    public static void main(String[] args)
    {
        // TODO Auto-generated method stub


            String[] a = {"Java", "Kotlin", "Scala", "Groovy", "Python"};
            String[] result = new String[5];
            System.arraycopy(a ,0, result ,0,3);
            for (String item : result) {
                System.out.println(item);
            }



    }

}
```

```
<terminated> Slicearray [Java App
Java
Kotlin
Scala
null
null
```

Implementation of second Task (B);


IBank.java
HashTableDemo.java
Doctor.java
two_sum_Problem.java
Problems
Javadoc
Decla
<terminated> fibonacci_series [Java A
enter a no print fibonacci s
10
011235813213455
public class fibonacci_series {
public static int fibonacci(int n)
{
if(n==0)
{
return 0;
}
else if(n==1)
{
return 1;
}
else{
return fibonacci(n-1)+fibonacci(n-2);
}
}
// TODO Auto-generated constructor stub
public static void main(String[] args)
{
System.out.println("enter a no print fibonacci series");
Scanner obj=new Scanner(System.in);
int a=obj.nextInt();
for(int i=0;i<=a;i++)
{
System.out.print(fibonacci(i));
}
}
}