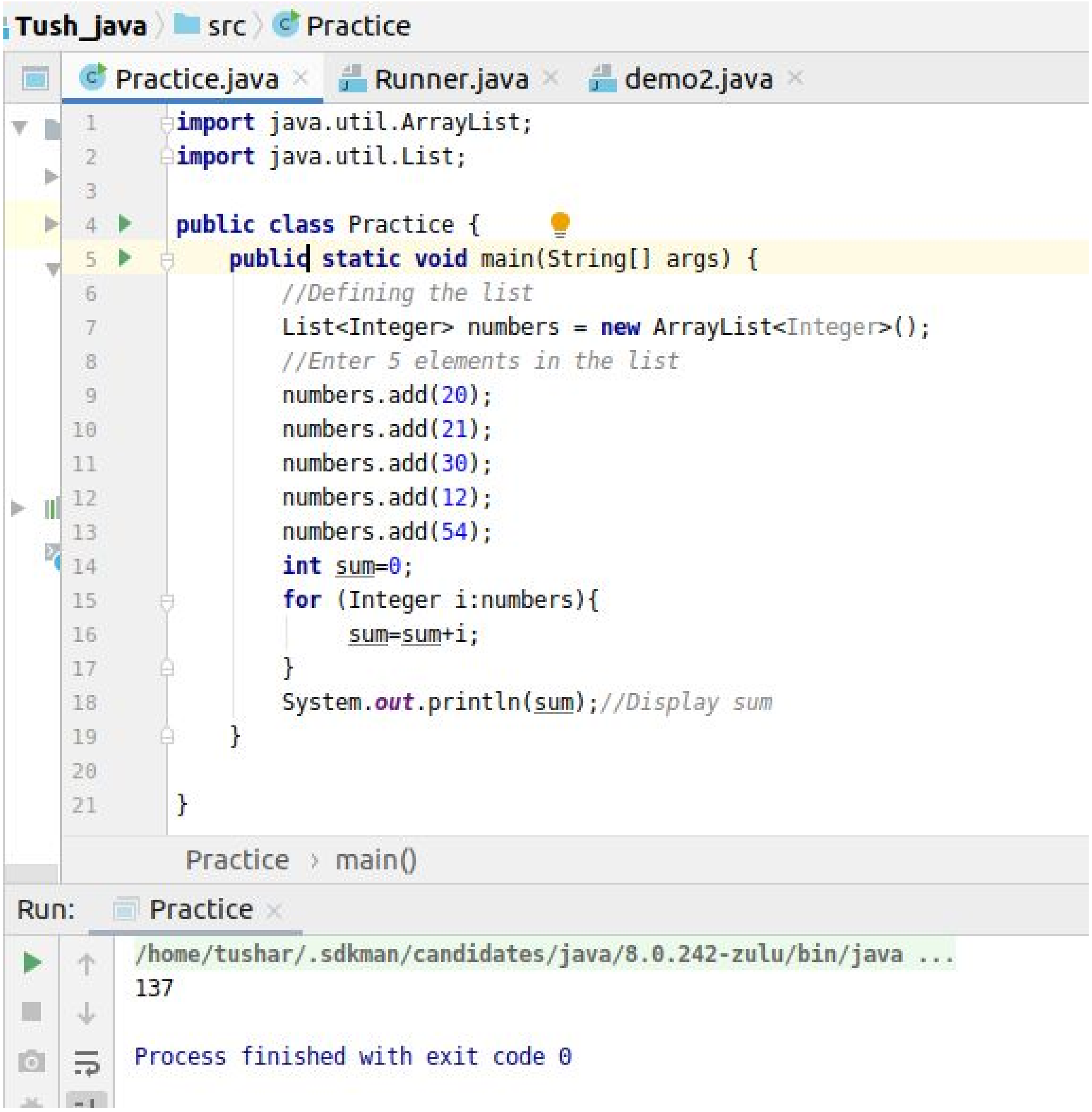
Collections

Q1. Write Java code to define List . Insert 5 floating point numbers in List, and using an iterator, find the sum of the numbers in List.

Answer:

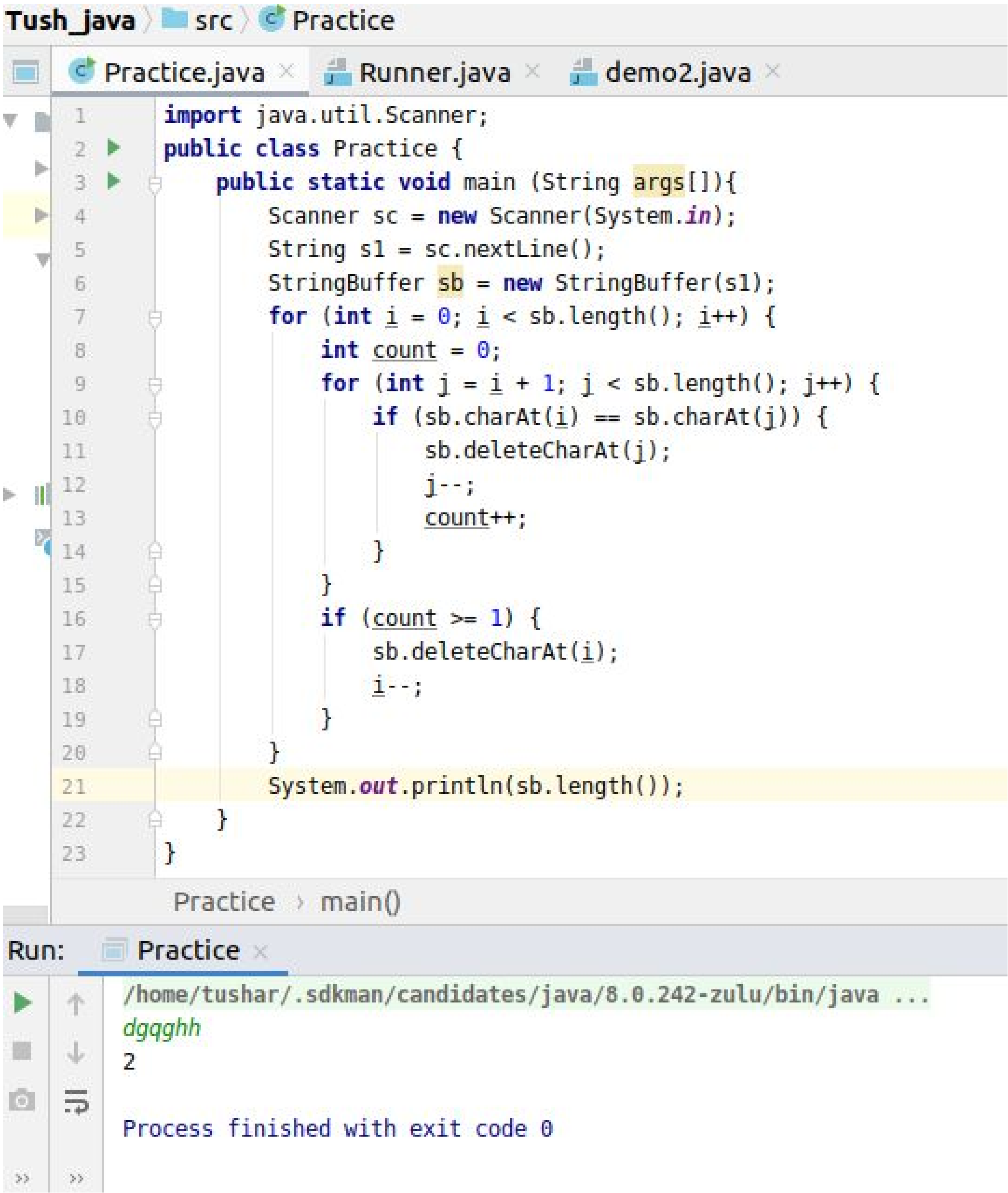


```java
import java.util.ArrayList;
import java.util.List;

public class Practice {
    public static void main(String[] args) {
        //Defining the list
        List<Integer> numbers = new ArrayList<Integer>();
        //Enter 5 elements in the list
        numbers.add(20);
        numbers.add(21);
        numbers.add(30);
        numbers.add(12);
        numbers.add(54);
        int sum=0;
        for (Integer i:numbers){
            sum=sum+i;
        }
        System.out.println(sum);//Display sum
    }

}
```

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
137

Process finished with exit code 0
```

Q2. Write a method that takes a string and returns the number of unique characters in the string.

Answer:

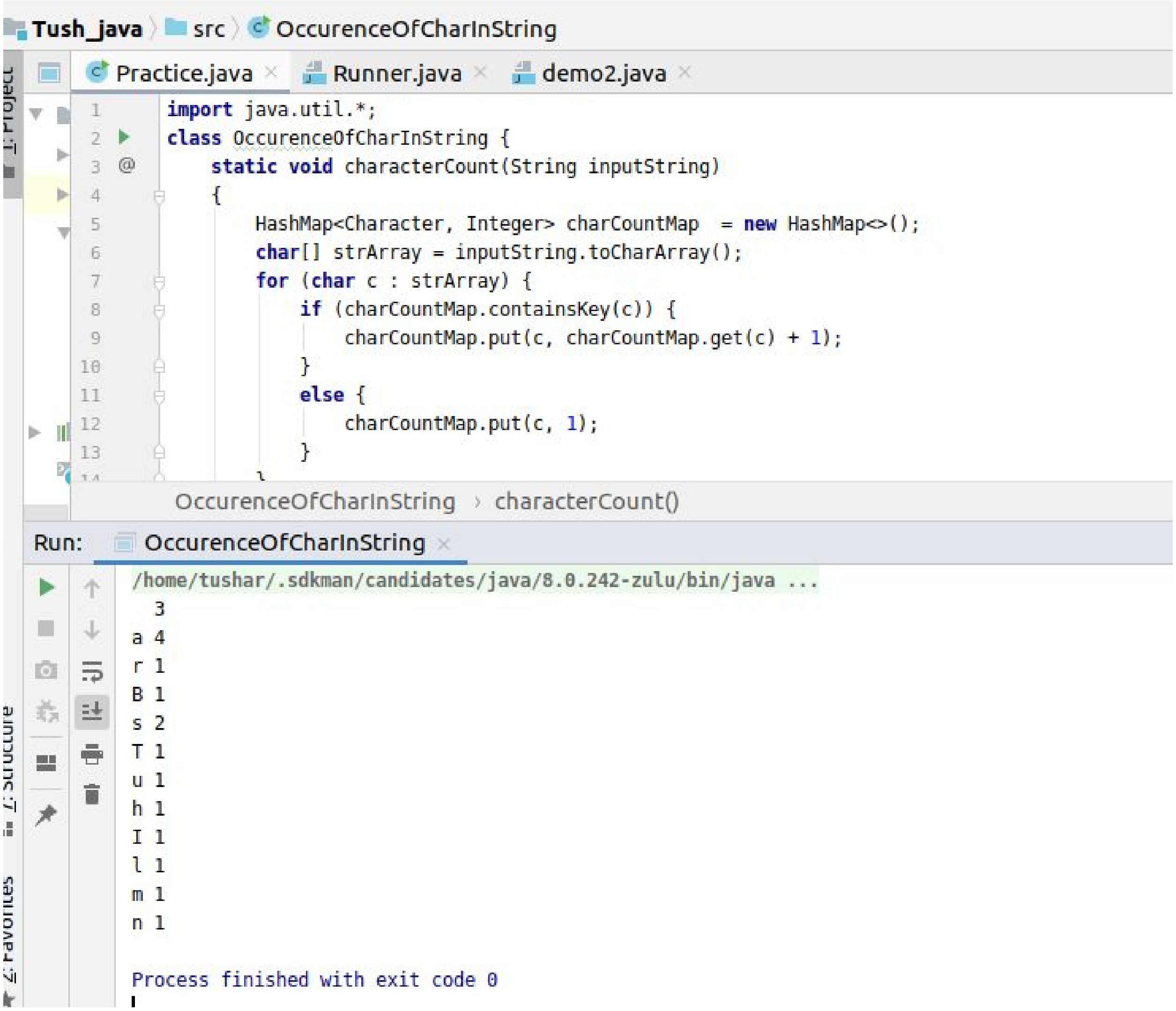


```java
import java.util.Scanner;
public class Practice {
    public static void main (String args[]){
        Scanner sc = new Scanner(System.in);
        String s1 = sc.nextLine();
        StringBuffer sb = new StringBuffer(s1);
        for (int i = 0; i < sb.length(); i++) {
            int count = 0;
            for (int j = i + 1; j < sb.length(); j++) {
                if (sb.charAt(i) == sb.charAt(j)) {
                    sb.deleteCharAt(j);
                    j--;
                    count++;
                }
            }
            if (count >= 1) {
                sb.deleteCharAt(i);
                i--;
            }
        }
        System.out.println(sb.length());
    }
}
```

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
dgqghh
2

Process finished with exit code 0
```

Q3. Write a method that takes a string and print the number of occurrence of each character characters in the string.

Answer:



```
import java.util.*;
class OccurenceOfCharInString {
    static void characterCount(String inputString)
    {
        HashMap<Character, Integer> charCountMap  = new HashMap<>();
        char[] strArray = inputString.toCharArray();
        for (char c : strArray) {
            if (charCountMap.containsKey(c)) {
                charCountMap.put(c, charCountMap.get(c) + 1);
            }
            else {
                charCountMap.put(c, 1);
            }
```

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
  3
a 4
r 1
B 1
s 2
T 1
u 1
h 1
I 1
l 1
m 1
n 1

Process finished with exit code 0
```

Code:

```
import java.util.*;

class OccurenceOfCharInString {

  static void characterCount(String inputString)

  {

     HashMap<Character, Integer> charCountMap  = new HashMap<>();

     char[] strArray = inputString.toCharArray();

     for (char c : strArray) {
            if (charCountMap.containsKey(c)) {
                charCountMap.put(c, charCountMap.get(c) + 1);
            }
            else {
                charCountMap.put(c, 1);
            }
        }
        for (Map.Entry entry : charCountMap.entrySet()) {
            System.out.println(entry.getKey() + " " + entry.getValue());
        }
    }
    public static void main(String[] args)
    {
        String str = "I am Tushar Bansal";
        characterCount(str);
    }
}
```

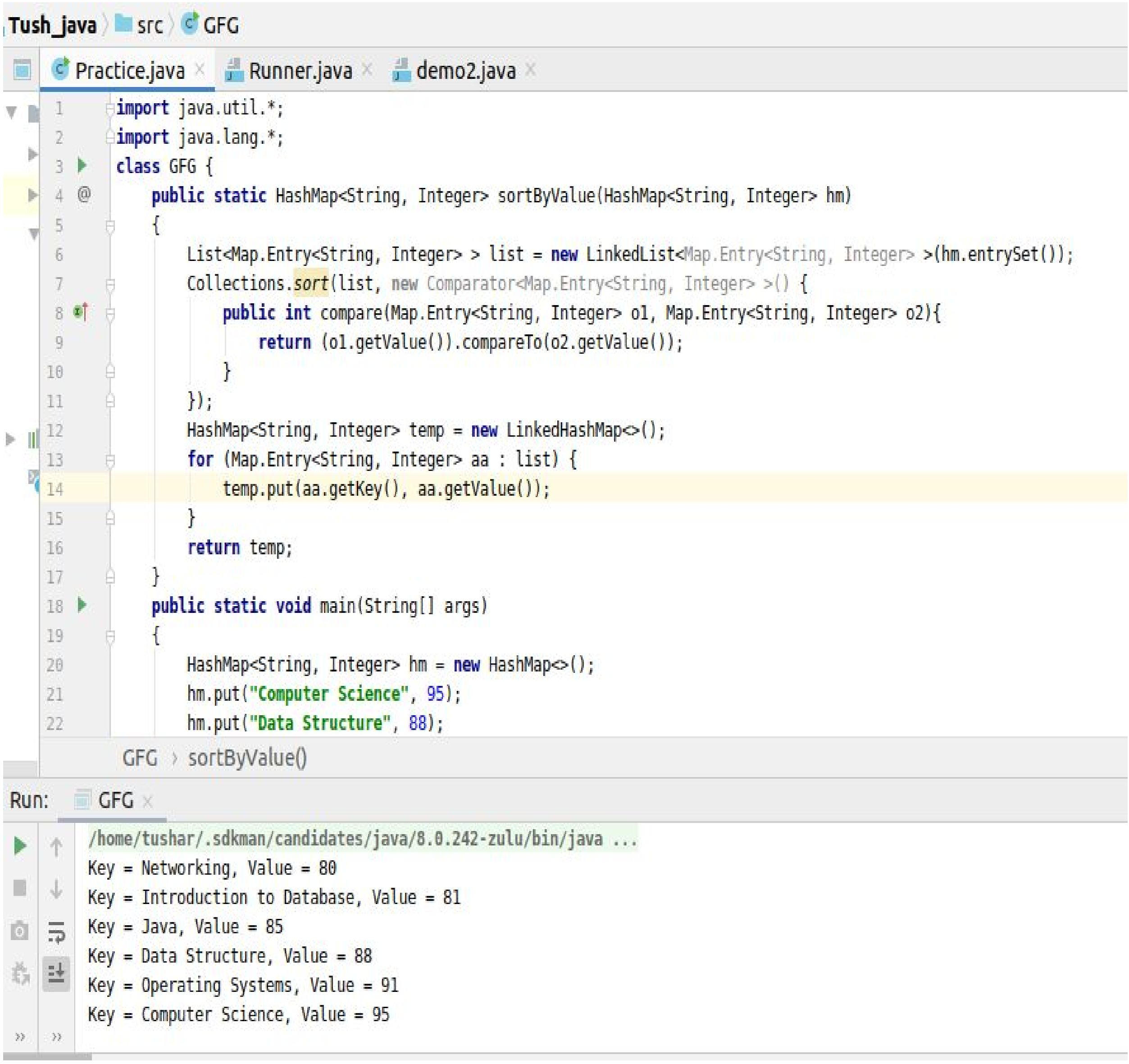
Q4. Write a program to sort HashMap by value.

Answer:

Code:

```
import java.util.*;
import java.lang.*;
```

```
class GFG {
  public static HashMap<String, Integer> sortByValue(HashMap<String, Integer> hm)
  {
    List<Map.Entry<String, Integer> > list = new LinkedList<Map.Entry<String, Integer> >(hm.entrySet());
    Collections.sort(list, new Comparator<Map.Entry<String, Integer> >() {
      public int compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2){
        return (o1.getValue()).compareTo(o2.getValue());
      }
    });
    HashMap<String, Integer> temp = new LinkedHashMap<>();
    for (Map.Entry<String, Integer> aa : list) {
      temp.put(aa.getKey(), aa.getValue());
    }
    return temp;
  }
  public static void main(String[] args)
  {
    HashMap<String, Integer> hm = new HashMap<>();
    hm.put("Computer Science", 95);
    hm.put("Data Structure", 88);
    hm.put("Introduction to Database", 81);
    hm.put("Java", 85);
    hm.put("Operating Systems", 91);
    hm.put("Networking", 80);
    Map<String, Integer> hm1 = sortByValue(hm);
    for (Map.Entry<String, Integer> en : hm1.entrySet()) {
      System.out.println("Key = " + en.getKey() + ", Value = " + en.getValue());
    }
  }
}
```



```
import java.util.*;
import java.lang.*;
class GFG {
    public static HashMap<String, Integer> sortByValue(HashMap<String, Integer> hm)
    {
        List<Map.Entry<String, Integer> > list = new LinkedList<Map.Entry<String, Integer> >(hm.entrySet());
        Collections.sort(list, new Comparator<Map.Entry<String, Integer> >() {
            public int compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2){
                return (o1.getValue()).compareTo(o2.getValue());
            }
        });
        HashMap<String, Integer> temp = new LinkedHashMap<>();
        for (Map.Entry<String, Integer> aa : list) {
            temp.put(aa.getKey(), aa.getValue());
        }
        return temp;
    }
    public static void main(String[] args)
    {
        HashMap<String, Integer> hm = new HashMap<>();
        hm.put("Computer Science", 95);
        hm.put("Data Structure", 88);
```

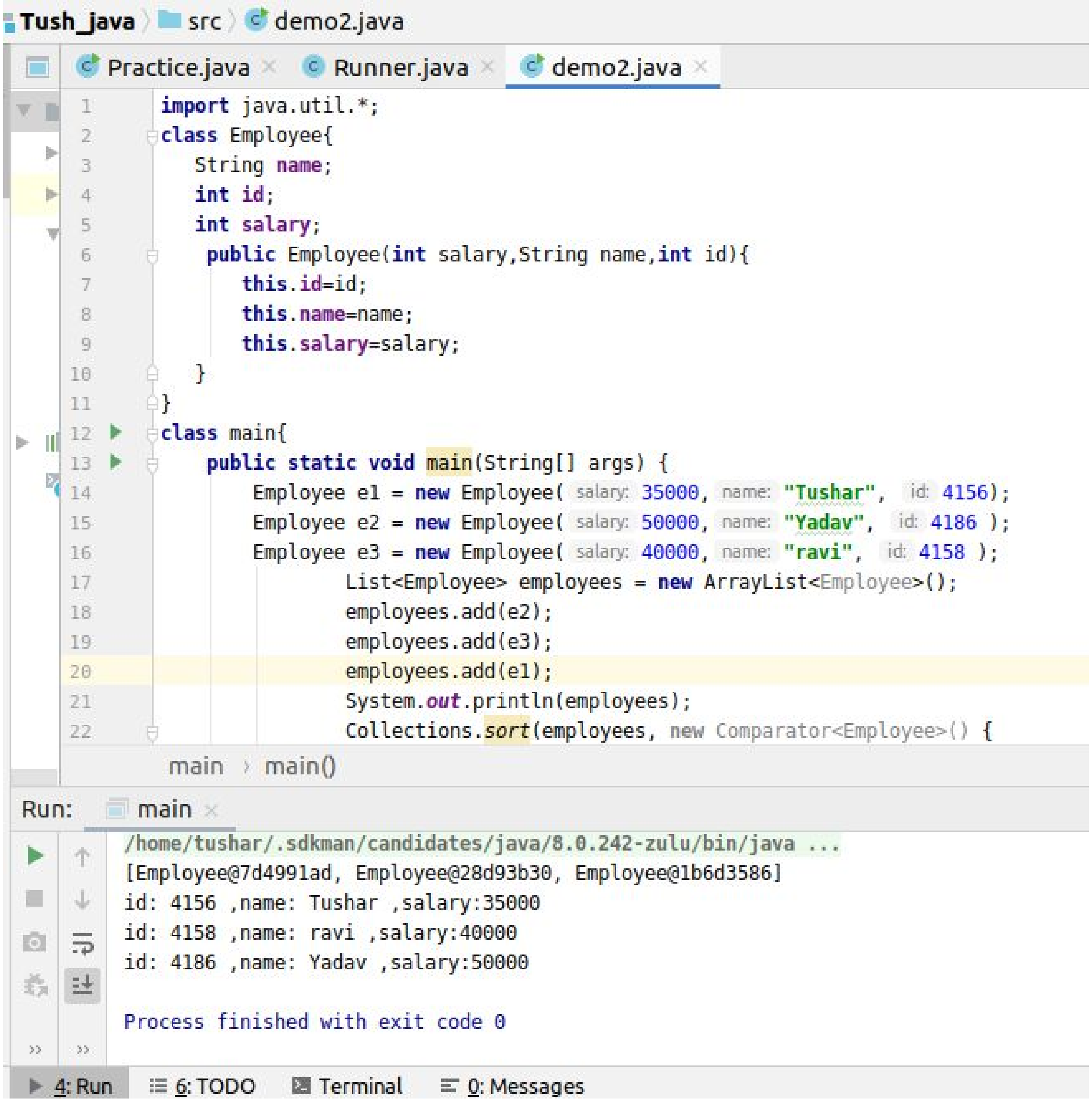
```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Key = Networking, Value = 80
Key = Introduction to Database, Value = 81
Key = Java, Value = 85
Key = Data Structure, Value = 88
Key = Operating Systems, Value = 91
Key = Computer Science, Value = 95
```

Q5.Write a program to sort Employee objects based on highest salary using Comparator. Employee class{ Double Age; Double Salary; String Name}

Answer:

Code:

```
import java.util.*;

class Employee{

  String name;
  int id;
  int salary;
   public Employee(int salary,String name,int id){
      this.id=id;
      this.name=name;
      this.salary=salary;
  }
}
class main{
   public static void main(String[] args) {
       Employee e1 = new Employee(35000,"Tushar", 4156);
       Employee e2 = new Employee(50000,"Yadav", 4186 );
       Employee e3 = new Employee(40000,"ravi", 4158 );
               List<Employee> employees = new ArrayList<Employee>();
               employees.add(e2);
               employees.add(e3);
               employees.add(e1);
               System.out.println(employees);
               Collections.sort(employees, new Comparator<Employee>() {
                   @Override
                   public int compare(Employee o1, Employee o2) {
                       return o1.salary - o2.salary;
                   }
               });
               for (Employee i: employees)
               System.out.println("id: "+i.id+" ,name: "+i.name+" ,salary:"+i.salary);
          }
       }
```



```java
import java.util.*;
class Employee{
   String name;
   int id;
   int salary;
    public Employee(int salary,String name,int id){
       this.id=id;
       this.name=name;
       this.salary=salary;
   }
}
class main{
    public static void main(String[] args) {
        Employee e1 = new Employee( salary: 35000, name: "Tushar",  id: 4156);
        Employee e2 = new Employee( salary: 50000, name: "Yadav",  id: 4186 );
        Employee e3 = new Employee( salary: 40000, name: "ravi",  id: 4158 );
                List<Employee> employees = new ArrayList<Employee>();
                employees.add(e2);
                employees.add(e3);
                employees.add(e1);
                System.out.println(employees);
                Collections.sort(employees, new Comparator<Employee>() {
```

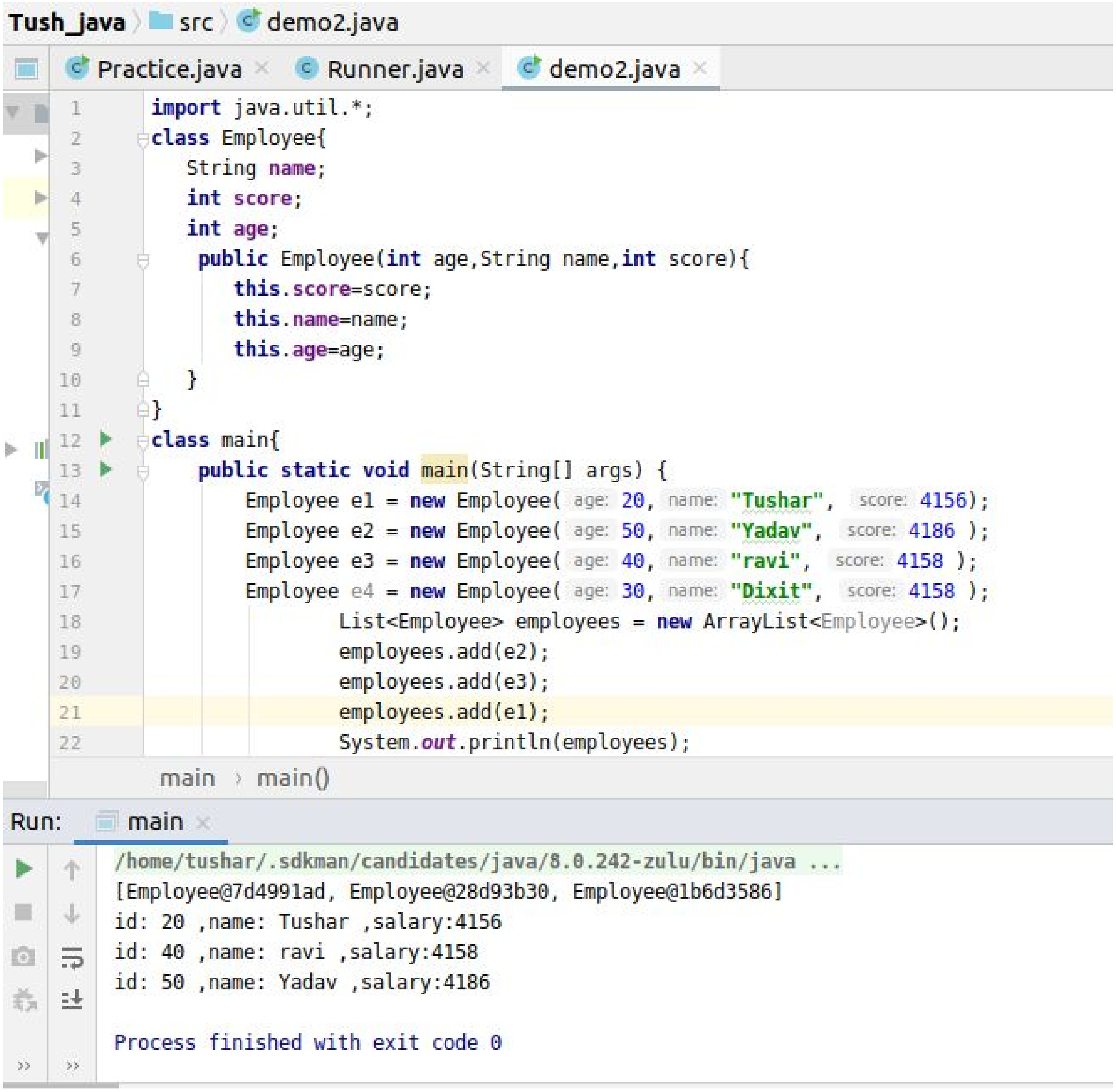
```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
[Employee@7d4991ad, Employee@28d93b30, Employee@1b6d3586]
id: 4156 ,name: Tushar ,salary:35000
id: 4158 ,name: ravi ,salary:40000
id: 4186 ,name: Yadav ,salary:50000

Process finished with exit code 0
```

Q6.Write a program to sort the Student objects based on Score , if the score are same then sort on First Name . Class Student{ String Name; Double Score; Double Age}.

Answer:

Code:

```
import java.util.*;
class Employee{
 String name;
 int score;
 int age;
  public Employee(int age,String name,int score){
    this.score=score;
    this.name=name;
    this.age=age;
 }
}
class main{
  public static void main(String[] args) {
    Employee e1 = new Employee(20,"Tushar", 4156);
    Employee e2 = new Employee(50,"Yadav", 4186 );
    Employee e3 = new Employee(40,"ravi", 4158 );
    Employee e4 = new Employee(30,"Dixit", 4158 );
        List<Employee> employees = new ArrayList<Employee>();
        employees.add(e2);
        employees.add(e3);
        employees.add(e1);
        System.out.println(employees);
        Collections.sort(employees, new Comparator<Employee>() {
          @Override
          public int compare(Employee o1, Employee o2) {
            return o1.score - o2.score;
          }
        });
        for (Employee i: employees)
        System.out.println("id: "+i.age+" ,name: "+i.name+" ,salary:"+i.score);
      }
    }
```



```java
import java.util.*;
class Employee{
   String name;
   int score;
   int age;
    public Employee(int age,String name,int score){
       this.score=score;
       this.name=name;
       this.age=age;
   }
}
class main{
    public static void main(String[] args) {
        Employee e1 = new Employee( age: 20, name: "Tushar",  score: 4156);
        Employee e2 = new Employee( age: 50, name: "Yadav",  score: 4186 );
        Employee e3 = new Employee( age: 40, name: "ravi",  score: 4158 );
        Employee e4 = new Employee( age: 30, name: "Dixit",  score: 4158 );
                List<Employee> employees = new ArrayList<Employee>();
                employees.add(e2);
                employees.add(e3);
                employees.add(e1);
                System.out.println(employees);
```

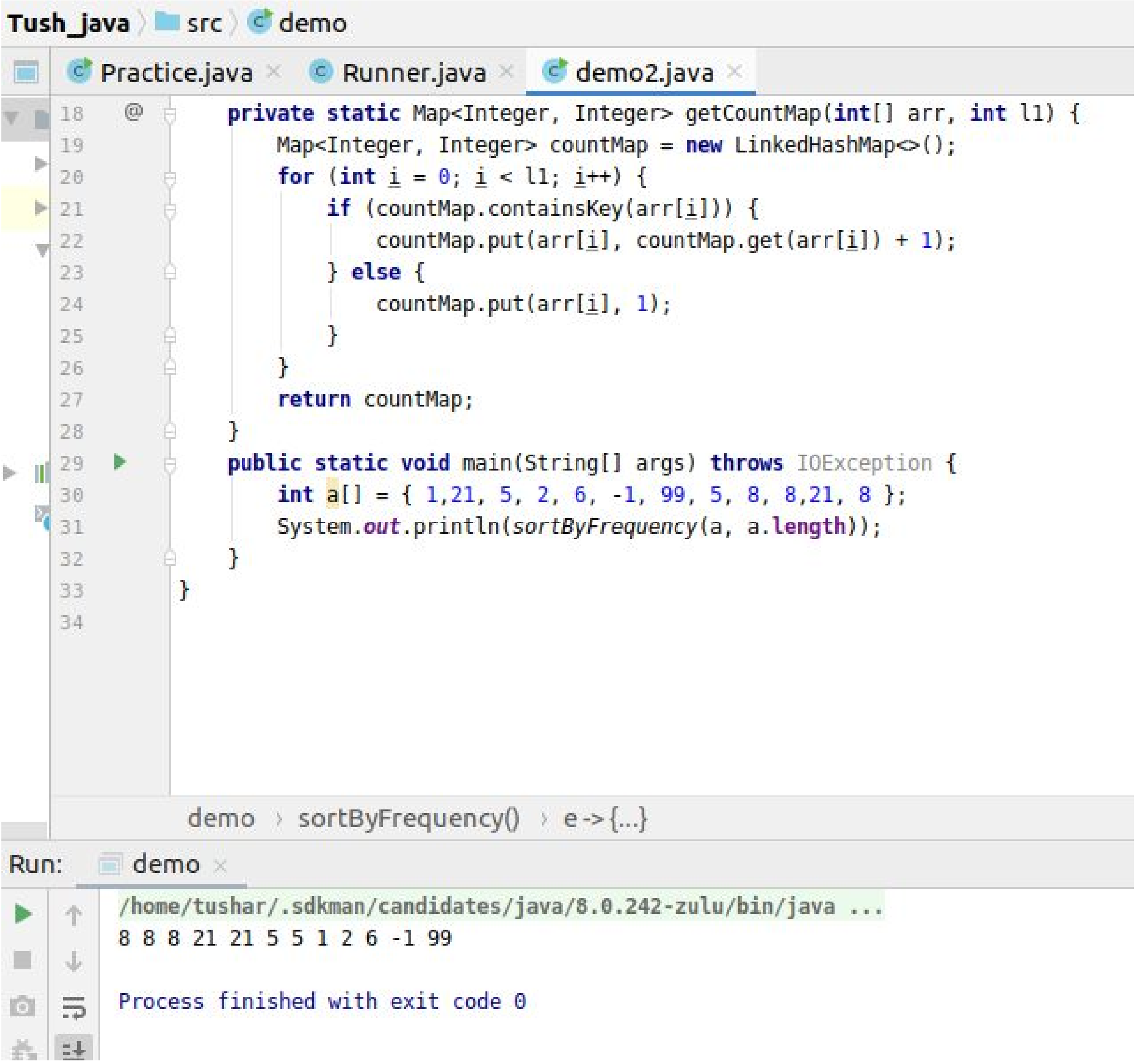
```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
[Employee@7d4991ad, Employee@28d93b30, Employee@1b6d3586]
id: 20 ,name: Tushar ,salary:4156
id: 40 ,name: ravi ,salary:4158
id: 50 ,name: Yadav ,salary:4186

Process finished with exit code 0
```

Q7.Print the elements of an array in the decreasing frequency if 2 numbers have same frequency then print the one which came first.

Answer:

Code:

```
import java.io.IOException;
import java.util.*;
class demo {
  public static StringBuffer sortByFrequency(int arr1[], int l1) {
    Map<Integer, Integer> countMap = getCountMap(arr1, l1);
    StringBuffer result = new StringBuffer();
    countMap.entrySet().stream()
        .sorted(Map.Entry.<Integer, Integer> comparingByValue().reversed())
        .forEach(e -> {
          int key = e.getKey();
          int val = e.getValue();
          for (int i = 0; i < val; i++) {
            result.append(key + " ");
          }
        });
    return result;
  }
  private static Map<Integer, Integer> getCountMap(int[] arr, int l1) {
    Map<Integer, Integer> countMap = new LinkedHashMap<>();
    for (int i = 0; i < l1; i++) {
      if (countMap.containsKey(arr[i])) {
        countMap.put(arr[i], countMap.get(arr[i]) + 1);
      } else {
        countMap.put(arr[i], 1);
      }
    }
    return countMap;
  }
  public static void main(String[] args) throws IOException {
    int a[] = { 1,21, 5, 2, 6, -1, 99, 5, 8, 8,21, 8 };
    System.out.println(sortByFrequency(a, a.length));
  }
}
```



```java
    private static Map<Integer, Integer> getCountMap(int[] arr, int l1) {
        Map<Integer, Integer> countMap = new LinkedHashMap<>();
        for (int i = 0; i < l1; i++) {
            if (countMap.containsKey(arr[i])) {
                countMap.put(arr[i], countMap.get(arr[i]) + 1);
            } else {
                countMap.put(arr[i], 1);
            }
        }
        return countMap;
    }
    public static void main(String[] args) throws IOException {
        int a[] = { 1,21, 5, 2, 6, -1, 99, 5, 8, 8,21, 8 };
        System.out.println(sortByFrequency(a, a.length));
    }
}
```

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
8 8 8 21 21 5 5 1 2 6 -1 99

Process finished with exit code 0
```

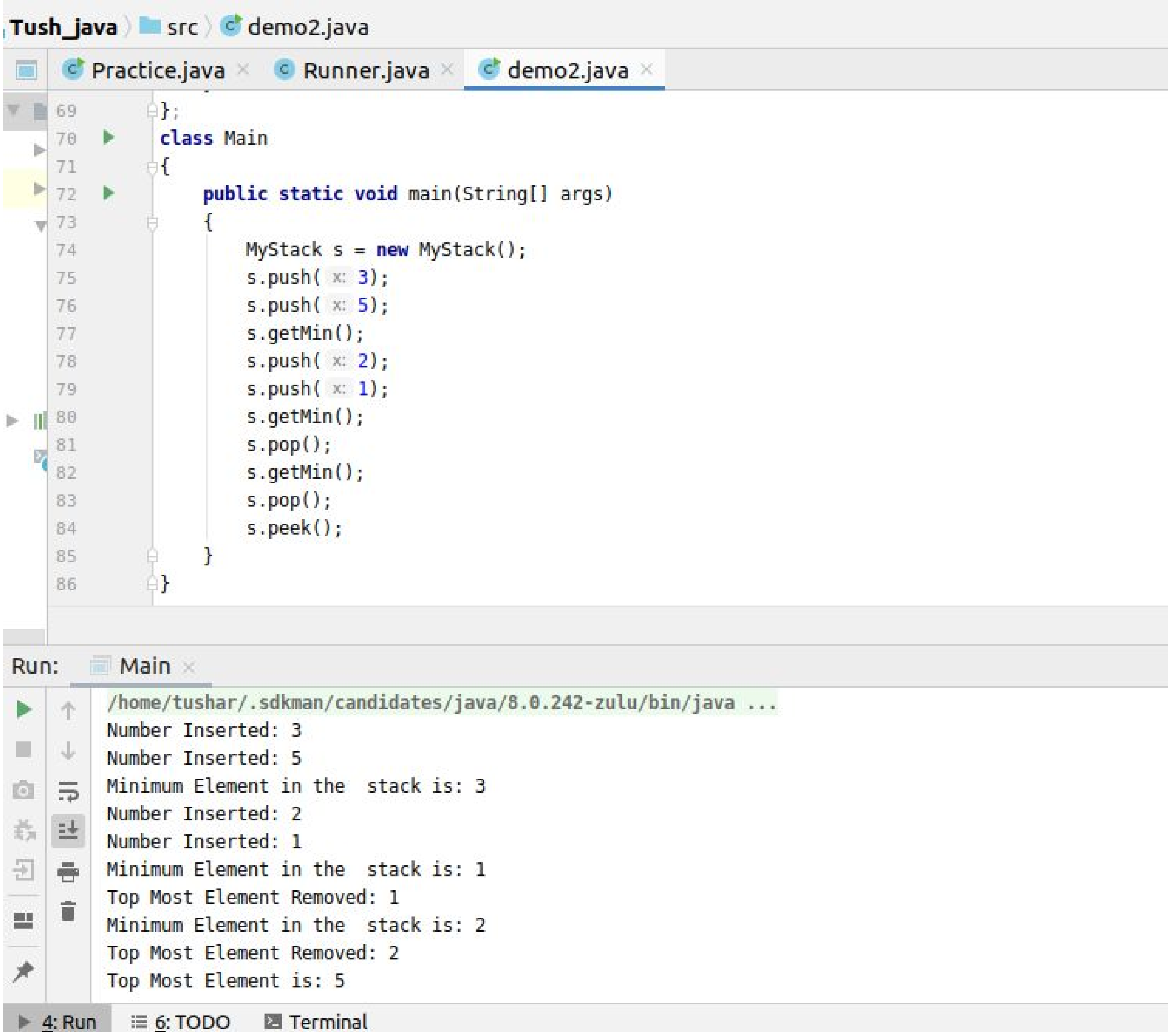
Q8.Design a Data Structure SpecialStack that supports all the stack operations like push(), pop(), isEmpty(), isFull() and an additional operation getMin() which should return minimum element from the SpecialStack.

Answer:

Code:

```java
import java.util.*;
class MyStack
```

```java
{
    Stack<Integer> s;
    Integer minEle;
    MyStack() { s = new Stack<Integer>(); }
    void getMin()
    {
        if (s.isEmpty())
            System.out.println("Stack is empty");
        else
            System.out.println("Minimum Element in the " +" stack is: " + minEle);
    }
    void peek()
    {
        if (s.isEmpty())
        {
            System.out.println("Stack is empty ");
            return;
        }

        Integer t = s.peek(); // Top element.
        System.out.print("Top Most Element is: ");

        if (t < minEle)
            System.out.println(minEle);
        else
            System.out.println(t);
    }
    void pop()
    {
        if (s.isEmpty())
        {
            System.out.println("Stack is empty");
            return;
        }

        System.out.print("Top Most Element Removed: ");
        Integer t = s.pop();
        if (t < minEle)
        {
            System.out.println(minEle);
            minEle = 2*minEle - t;
        }

        else
            System.out.println(t);
    }
    void push(Integer x)
    {
        if (s.isEmpty())
        {
            minEle = x;
            s.push(x);
            System.out.println("Number Inserted: " + x);
            return;
        }
        if (x < minEle)
        {
            s.push(2*x - minEle);
            minEle = x;
        }

        else
            s.push(x);

        System.out.println("Number Inserted: " + x);
    }
};
class Main
{
    public static void main(String[] args)
    {
        MyStack s = new MyStack();
        s.push(3);
        s.push(5);
        s.getMin();
        s.push(2);
        s.push(1);
        s.getMin();
        s.pop();
        s.getMin();
        s.pop();
        s.peek();
    }
}
```



```
};
class Main
{
    public static void main(String[] args)
    {
        MyStack s = new MyStack();
        s.push( x: 3);
        s.push( x: 5);
        s.getMin();
        s.push( x: 2);
        s.push( x: 1);
        s.getMin();
        s.pop();
        s.getMin();
        s.pop();
        s.peek();
    }
}
```

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Number Inserted: 3
Number Inserted: 5
Minimum Element in the  stack is: 3
Number Inserted: 2
Number Inserted: 1
Minimum Element in the  stack is: 1
Top Most Element Removed: 1
Minimum Element in the  stack is: 2
Top Most Element Removed: 2
Top Most Element is: 5
```