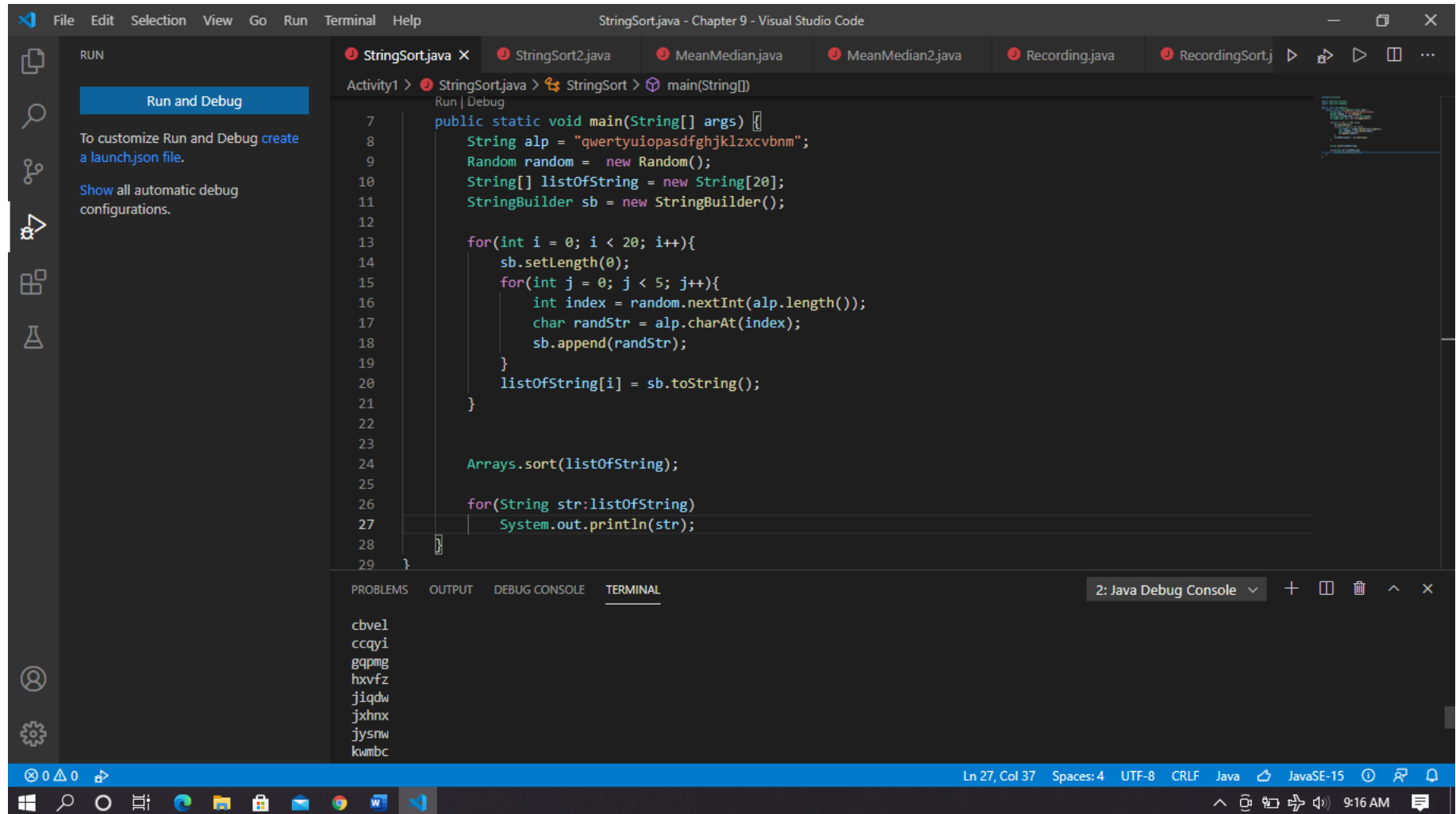
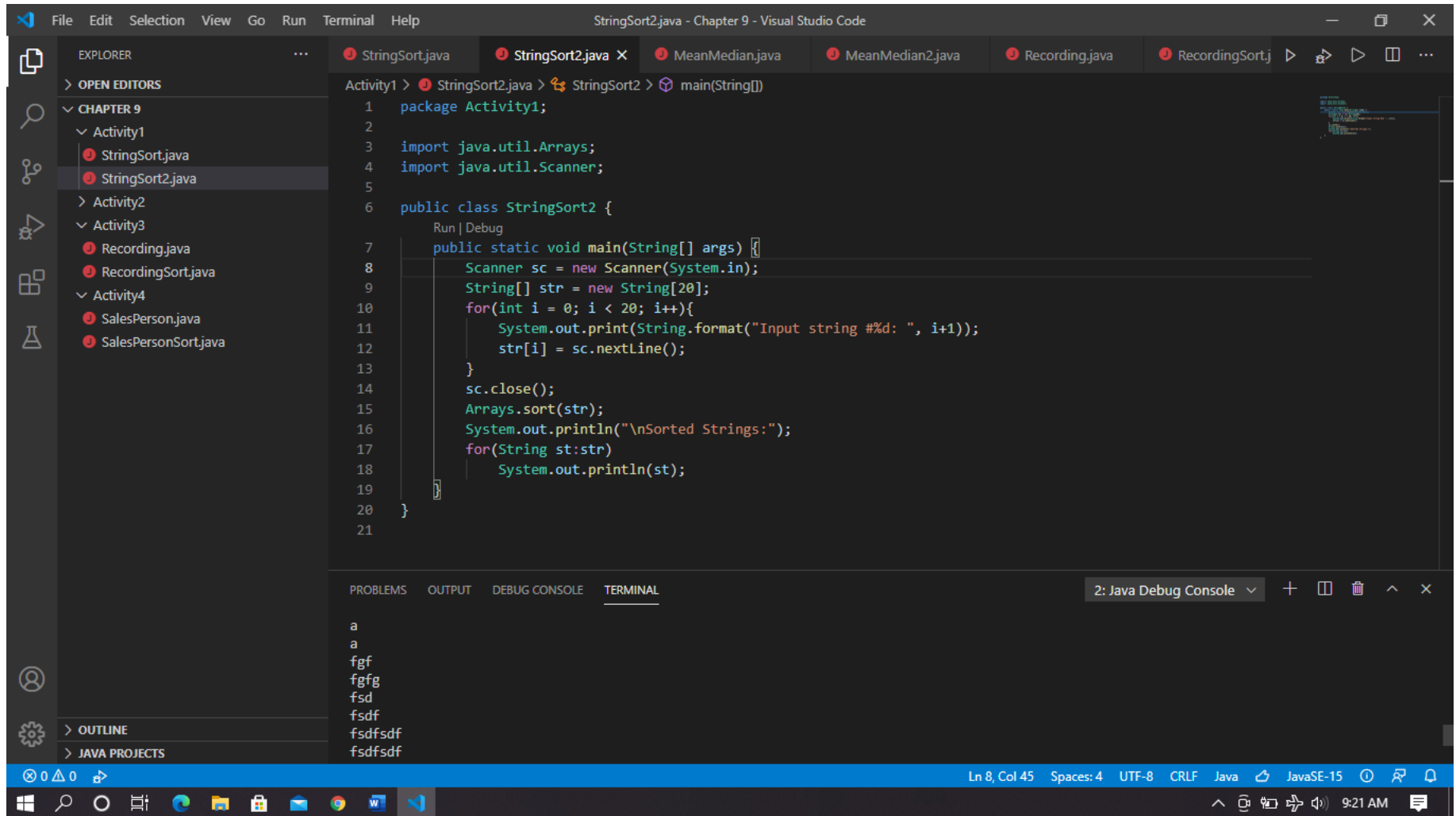


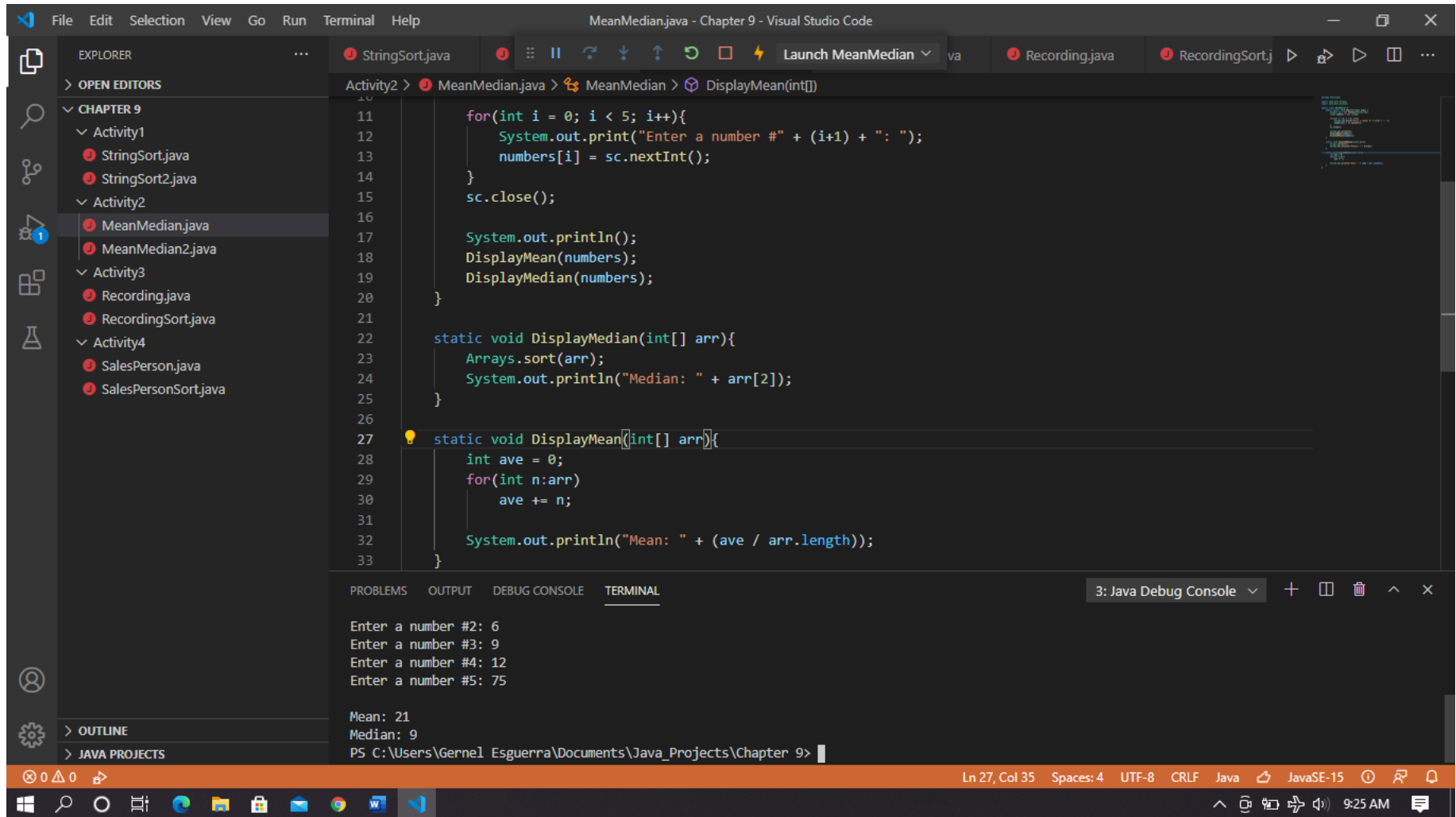
## 1. Chapter 9 – StringSort.java



## 1. Chapter 9 – StringSort2.java



## 2. Chapter 9 – MeanMedian.java



The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** Shows the project structure with 'CHAPTER 9' expanded, containing 'Activity1', 'Activity2', 'Activity3', and 'Activity4'. 'Activity2' is selected, showing 'MeanMedian.java'.
- EDITOR:** Displays the code for 'MeanMedian.java'. The code includes a loop to collect 5 numbers, a call to 'DisplayMean', and two static methods: 'DisplayMedian' and 'DisplayMean'.
- TERMINAL:** Shows the output of the program, including the prompts 'Enter a number #2: 6', 'Enter a number #3: 9', 'Enter a number #4: 12', 'Enter a number #5: 75', and the results 'Mean: 21' and 'Median: 9'.

```
11 for(int i = 0; i < 5; i++){
12     System.out.print("Enter a number #" + (i+1) + ": ");
13     numbers[i] = sc.nextInt();
14 }
15 sc.close();
16
17 System.out.println();
18 DisplayMean(numbers);
19 DisplayMedian(numbers);
20 }
21
22 static void DisplayMedian(int[] arr){
23     Arrays.sort(arr);
24     System.out.println("Median: " + arr[2]);
25 }
26
27 static void DisplayMean(int[] arr){
28     int ave = 0;
29     for(int n:arr)
30         ave += n;
31
32     System.out.println("Mean: " + (ave / arr.length));
33 }
```

Enter a number #2: 6  
Enter a number #3: 9  
Enter a number #4: 12  
Enter a number #5: 75  
  
Mean: 21  
Median: 9  
PS C:\Users\Gernel Esguerra\Documents\Java\_Projects\Chapter 9>

## 2. Chapter 9 – MeanMedian2.java

The screenshot shows the Visual Studio Code editor with the file `MeanMedian2.java` open. The Explorer sidebar on the left shows the project structure, including `CHAPTER 9` and its sub-activities. The main editor displays the code for `MeanMedian2.java`, which includes a `main` method and two static methods: `DisplayMedian` and `DisplayMean`. The terminal at the bottom shows the program's execution, where it prompts for six numbers and then displays the calculated mean and median.

```
File Edit Selection View Go Run Terminal Help
MeanMedian2.java - Chapter 9 - Visual Studio Code

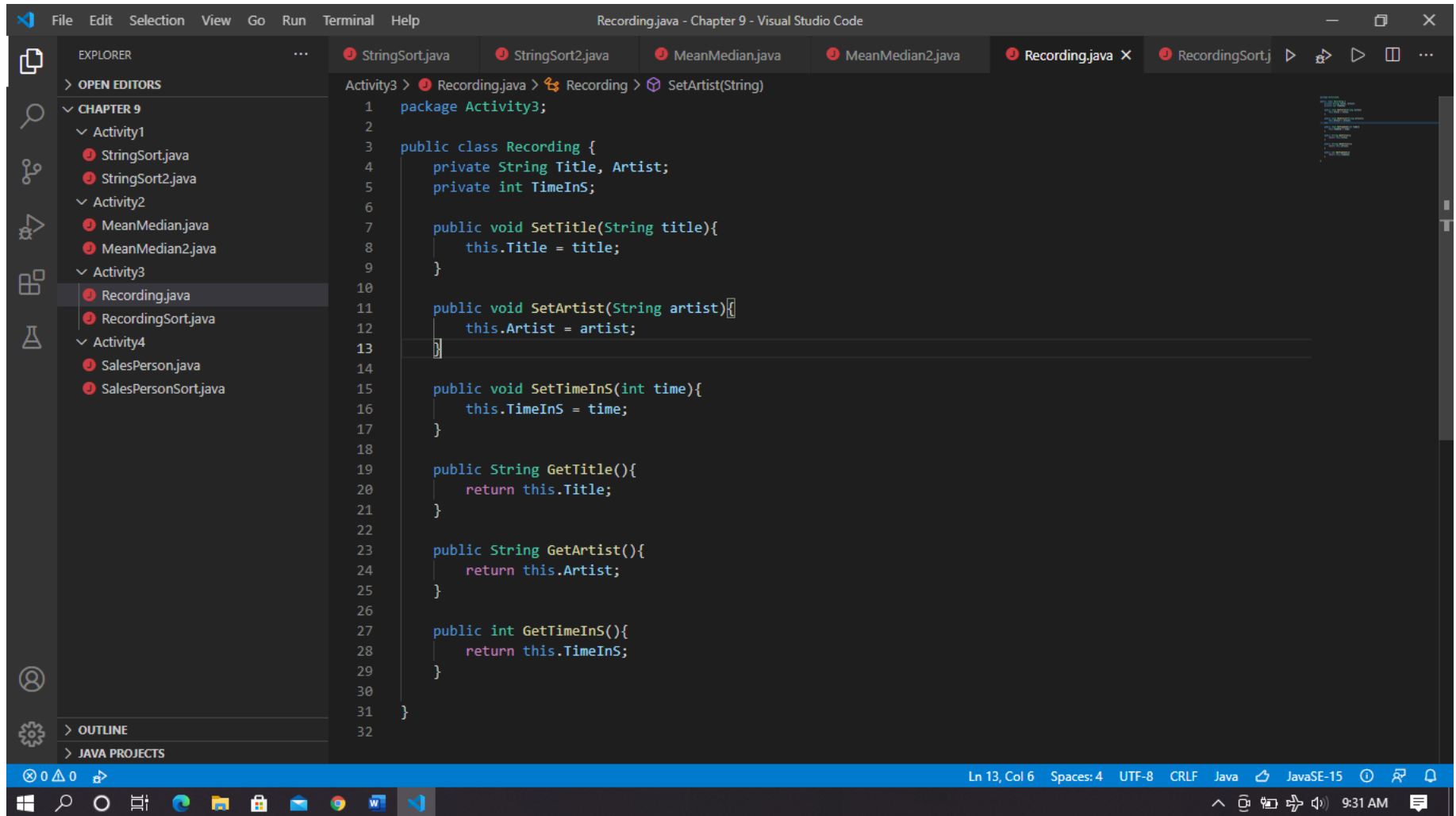
EXPLORER
> OPEN EDITORS
  CHAPTER 9
    Activity1
      StringSort.java
      StringSort2.java
    Activity2
      MeanMedian.java
      MeanMedian2.java
    Activity3
      Recording.java
      RecordingSort.java
    Activity4
      SalesPerson.java
      SalesPersonSort.java

Activity2 > MeanMedian2.java > MeanMedian2 > DisplayMedian(double[])
13      numbers[i] = sc.nextDouble();
14    }
15    sc.close();
16
17    System.out.println();
18    DisplayMean(numbers);
19    DisplayMedian(numbers);
20  }
21
22  static void DisplayMedian(double[] arr){
23    Arrays.sort(arr);
24    if(arr.length % 2 == 0){
25      System.out.println(String.format("Median: %.1f", (arr[(arr.length/2)-1] + arr[arr.length/2])/2));
26    }
27  }
28
29  static void DisplayMean(double[] arr){
30    double ave = 0;
31    for(double n:arr)
32      ave += n;
33
34    System.out.println(String.format("Mean: %.1f", (ave / arr.length)));
35  }
36

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
3: Java Debug Console
Enter a number #3: 45
Enter a number #4: 98
Enter a number #5: 11
Enter a number #6: 69

Mean: 41.8
Median: 35.0
PS C:\Users\Gernel Esguerra\Documents\Java_Projects\Chapter 9>
```

### 3. Chapter 9 – Recording.java



```
File Edit Selection View Go Run Terminal Help
Recording.java - Chapter 9 - Visual Studio Code

EXPLORER
> OPEN EDITORS
CHAPTER 9
  Activity1
    StringSort.java
    StringSort2.java
  Activity2
    MeanMedian.java
    MeanMedian2.java
  Activity3
    Recording.java
    RecordingSort.java
  Activity4
    SalesPerson.java
    SalesPersonSort.java

Activity3 > Recording.java > Recording > SetArtist(String)
1 package Activity3;
2
3 public class Recording {
4     private String Title, Artist;
5     private int TimeInS;
6
7     public void SetTitle(String title){
8         this.Title = title;
9     }
10
11     public void SetArtist(String artist){
12         this.Artist = artist;
13     }
14
15     public void SetTimeInS(int time){
16         this.TimeInS = time;
17     }
18
19     public String GetTitle(){
20         return this.Title;
21     }
22
23     public String GetArtist(){
24         return this.Artist;
25     }
26
27     public int GetTimeInS(){
28         return this.TimeInS;
29     }
30
31 }
32
```

Ln 13, Col 6 Spaces: 4 UTF-8 CRLF Java JavaSE-15 9:31 AM

### 3. Chapter 9 - RecordingSort.java

The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** Shows the project structure with CHAPTER 9 expanded, containing Activity1, Activity2, Activity3, and Activity4. Activity3 contains RecordingSort.java, which is the current file.
- EDITOR:** Displays the code for RecordingSort.java, showing the main method. The code includes a while loop that prompts the user for sorting options (Title, Artist, Length, or Quit) and then sorts the array based on the selected option.
- TERMINAL:** Shows the output of the program, including the sorting options menu and the results of the sorting process.

```
27         rc[i].SetTimeInS(time);
28
29         System.out.println();
30     }
31
32     String c = "T";
33
34     while(!c.toUpperCase().equals("X")){
35         System.out.println("Sorting Options: [T]Song Title, [A]Song Artist, [L]Song Length and [X]Quit");
36         System.out.print("Input Option: ");
37         c = sc.nextLine();
38
39         if(c.substring(0, 1).toUpperCase().equals("T")) { Sort(rc, "title"); Display(rc);}
40         else if(c.substring(0, 1).toUpperCase().equals("A")) { Sort(rc, "artist"); Display(rc); }
41         else if(c.substring(0, 1).toUpperCase().equals("L")) { Sort(rc, "length"); Display(rc);}
42         else{
43             if(!c.toUpperCase().equals("X") || !c.substring(0, 1).toUpperCase().equals("T") || !c.substring(0, 1).toUpperCase().equals("A") || !c.substring(0, 1).toUpperCase().equals("L")){
44                 System.out.println("Invalid Option. Please try again.");
45             }
46         }
47     }
```

3: Java Debug Console

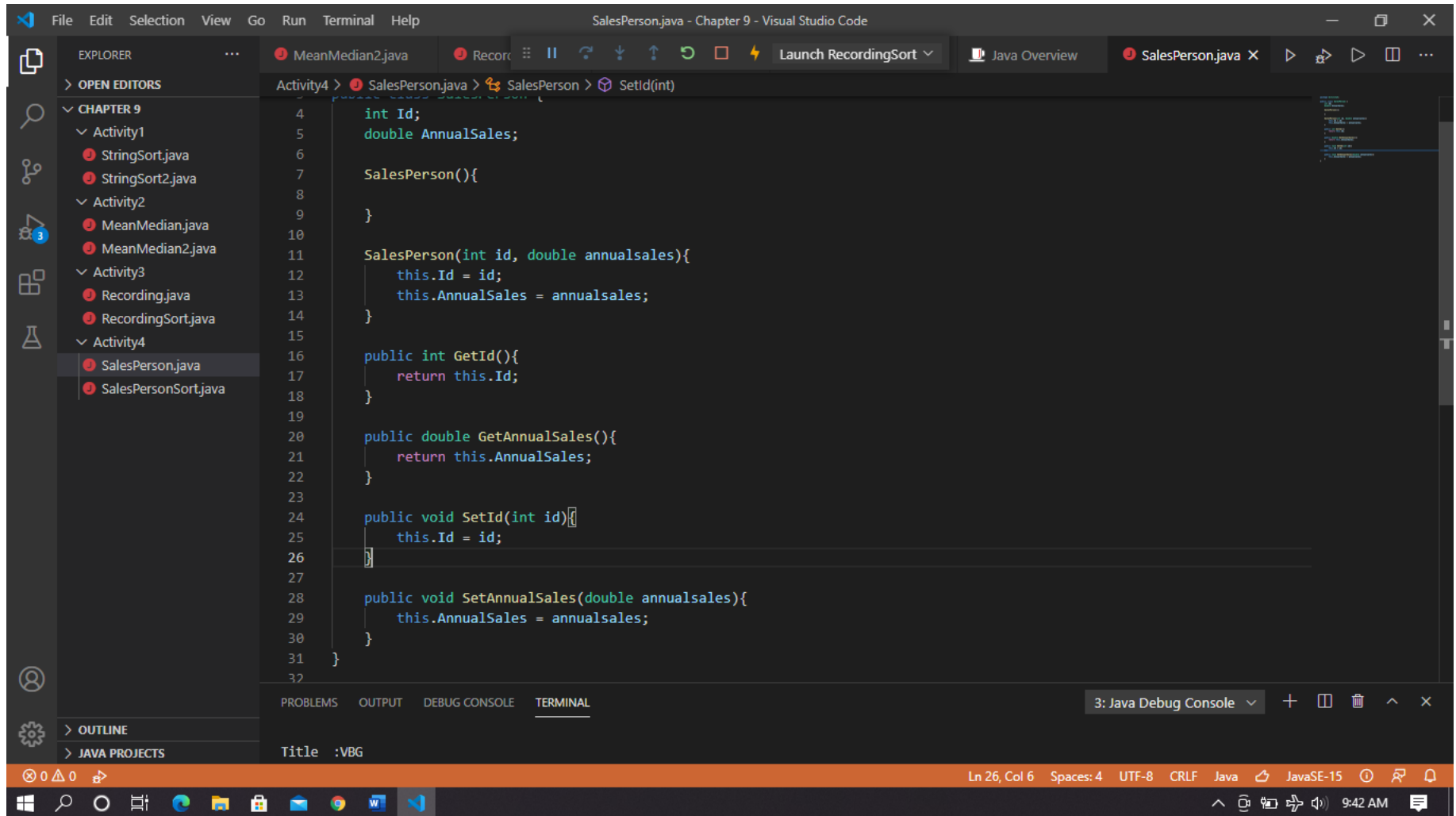
Artist :MMM  
Length :694s

Title :RTR  
Artist :DSA  
Length :212s

Title :VBG  
Artist :AAA  
Length :900s

Sorting Options: [T]Song Title, [A]Song Artist, [L]Song Length and [X]Quit  
Inout Option: []

#### 4. SalesPerson.java



#### 4. Chapter 9 – SalesPersonSort.java

The screenshot shows the Visual Studio Code editor with the file `SalesPersonSort.java` open. The Explorer sidebar on the left shows the project structure, including `CHAPTER 9` and `Activity4`. The main editor displays the following Java code:

```
50         sb.append("SalesPerson Id : " + r.GetId() + "\n");
51         sb.append("Annual Sales   : $" + r.GetAnnualSales() + "\n\n");
52     }
53     System.out.println(sb.toString());
54 }
55
56 static void Sort(SalesPerson[] rc, String sortOption){
57     SalesPerson r;
58     for(int i = 0; i < rc.length; i++){
59         for(int j = 0; j < rc.length; j++){
60             switch(sortOption){
61                 case "id":
62                     if(rc[i].GetId() < rc[j].GetId()){
63                         r = rc[i];
64                         rc[i] = rc[j];
65                         rc[j] = r;
66                     } break;
```

The bottom panel shows the `TERMINAL` output, which includes the input for annual sales and the sorting options menu. The output shows the first three salespersons sorted by ID:

```
Enter SalesPerson Annual Sales: 985.64

Sorting Options: [I]Id, [S]Annual Sales and [X]Quit
Inout Option: I
SalesPerson Id : 875
Annual Sales   : $65.96

SalesPerson Id : 2536
Annual Sales   : $56.6

SalesPerson Id : 4785
Annual Sales   : $265.0

SalesPerson Id : 4875
Annual Sales   : $985.64
```

The status bar at the bottom indicates the current position is `Ln 37, Col 62` with `Spaces: 4`, `UTF-8` encoding, `CRLF` line endings, and the `Java` language mode.