Ex.No.3

Programming with Arrays

02.09.2020

Mugilan E.S.

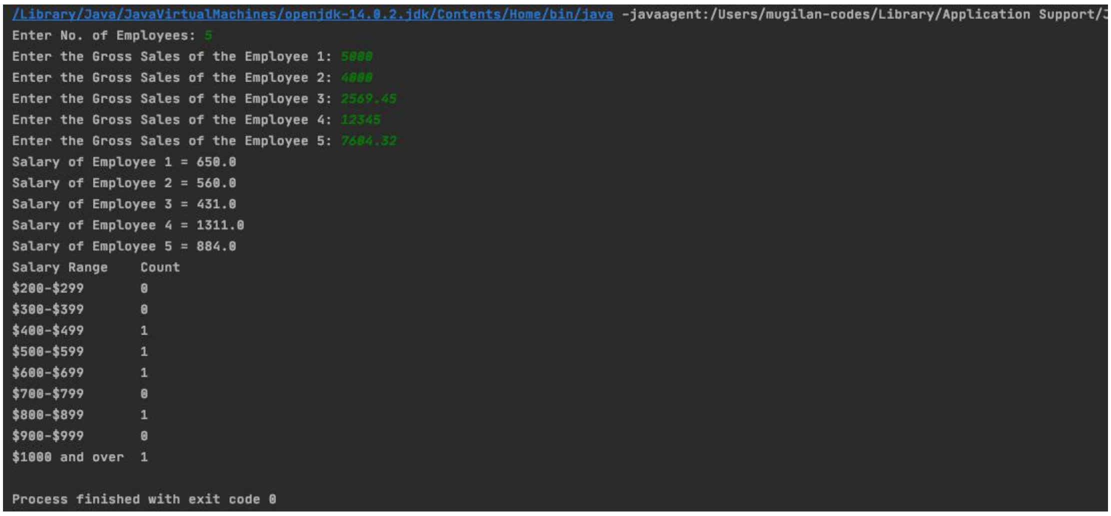
2019202033

1. Sales Commission

Sales.java

```
package lab.three;
import java.util.Scanner;
public class Sales {
    private static int getCommission(double sales) {
        return (int) (200 + (0.09 * sales));
    }
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter No. of Employees: ");
        int noOfEmployees = in.nextInt();
        double[] sales = new double[noOfEmployees];
        double[] salary = new double[noOfEmployees];
        for (int i = 0; i < sales.length; i++) {
            System.out.print("Enter the Gross Sales of the Employee " + (i + 1) + ": ");
            sales[i] = in.nextDouble();
            salary[i] = getCommission(sales[i]);
        }
        int[] salaryRangeCount = new int[9];
        String[] ranges = {
                "$200-$299",
                "$300-$399",
                "$400-$499",
                "$500-$599",
                "$600-$699",
                "$700-$799",
                "$800-$899",
                "$900-$999",
                "$1000 and over"
        };
        for (int i = 0; i < salary.length; i++) {
            System.out.println("Salary of Employee " + (i + 1) + " = " + salary[i]);
            if (salary[i] < 300) {
                salaryRangeCount[0]++;
            } else if (salary[i] > 300 && salary[i] < 400) {
                salaryRangeCount[1]++;
            } else if (salary[i] > 400 && salary[i] < 500) {
                salaryRangeCount[2]++;
            } else if (salary[i] > 500 && salary[i] < 600) {
                salaryRangeCount[3]++;
            } else if (salary[i] > 600 && salary[i] < 700) {
                salaryRangeCount[4]++;
            } else if (salary[i] > 700 && salary[i] < 800) {
                salaryRangeCount[5]++;
            } else if (salary[i] > 800 && salary[i] < 900) {
                salaryRangeCount[6]++;
            } else if (salary[i] > 900 && salary[i] < 1000) {
                salaryRangeCount[7]++;
            } else {
                salaryRangeCount[8]++;
            }
        }
        System.out.printf("%-15s %-5s %n", "Salary Range", "Count");
        for (int i = 0; i < ranges.length; i++) {
            System.out.printf("%-15s %-5d %n", ranges[i], salaryRangeCount[i]);
        }
    }
}
```

Output:



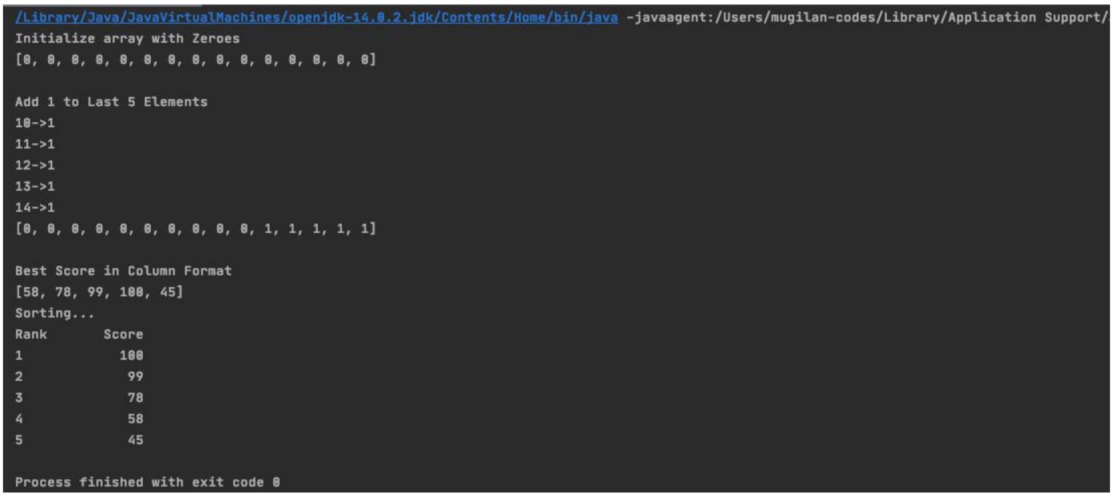
```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/J
Enter No. of Employees: 5
Enter the Gross Sales of the Employee 1: 5000
Enter the Gross Sales of the Employee 2: 4000
Enter the Gross Sales of the Employee 3: 2569.45
Enter the Gross Sales of the Employee 4: 12345
Enter the Gross Sales of the Employee 5: 7684.32
Salary of Employee 1 = 650.0
Salary of Employee 2 = 560.0
Salary of Employee 3 = 431.0
Salary of Employee 4 = 1311.0
Salary of Employee 5 = 884.0
Salary Range    Count
$200-$299       0
$300-$399       0
$400-$499       1
$500-$599       1
$600-$699       1
$700-$799       0
$800-$899       1
$900-$999       0
$1000 and over  1

Process finished with exit code 0
```

2. One-Dimensional Array Operations

OneDimensionalOperations.java

```
package lab.three;
import java.util.Arrays;
public class OneDimensionalOperations {
    public static void main(String[] args) {
        int[] arr = new int[15];
        System.out.println("Initialize array with Zeroes");
        Arrays.fill(arr, 0);
        System.out.println(Arrays.toString(arr));
        System.out.println("\nAdd 1 to Last 5 Elements");
        for(int i = 10; i < arr.length; i++){
            arr[i] += 1;
            System.out.println(i + "->" + arr[i]);
        }
        System.out.println(Arrays.toString(arr));
        System.out.println("\nBest Score in Column Format");
        int[] bestScores = new int[] {58, 78, 99, 100, 45};
        System.out.println(Arrays.toString(bestScores));
        System.out.println("Sorting...");
        for (int i = 0; i < bestScores.length; i++) {
            for (int j = i + 1; j < bestScores.length; j++) {
                int tmp = 0;
                if (bestScores[i] < bestScores[j]) {
                    tmp = bestScores[i];
                    bestScores[i] = bestScores[j];
                    bestScores[j] = tmp;
                }
            }
        }
        System.out.printf("%-5s %10s %n", "Rank", "Score");
        for(int i=0; i< bestScores.length;i++){
            System.out.printf("%-5d %10d %n", (i+1), bestScores[i]);
        }
    }
}
```

Output:



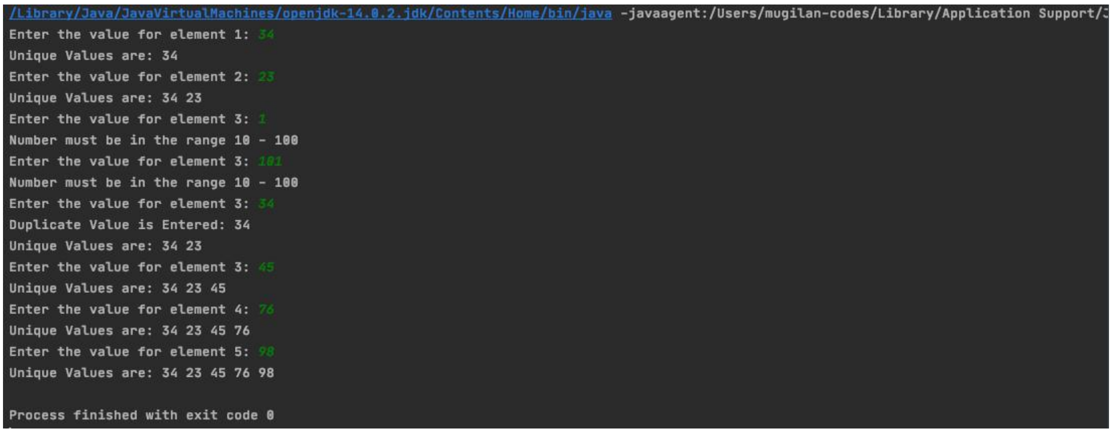
```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/
Initialize array with Zeroes
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Add 1 to Last 5 Elements
10->1
11->1
12->1
13->1
14->1
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1]

Best Score in Column Format
[58, 78, 99, 100, 45]
Sorting...
Rank       Score
1            100
2             99
3             78
4             58
5             45

Process finished with exit code 0
```

3. Duplicate Elimination

RemoveDuplicateArray.java

```java
package lab.three;
import java.util.Scanner;
public class RemoveDuplicateArray {
    private static boolean isPresent(int[] array, int n) {
        for (int j : array) {
            if (j == n) {
                return true;
            }
        }
        return false;
    }
    public static void main(String[] args) {
        int[] arr = new int[5];
        Scanner in = new Scanner(System.in);
        int i = 0;
        do {
            System.out.print("Enter the value for element " + (i + 1) + ": ");
            int n = in.nextInt();
            if (n < 10 || n > 100) {
                System.out.println("Number must be in the range 10 - 100");
                continue;
            }
            if (!isPresent(arr, n)) {
                arr[i] = n;
                i++;
            } else {
                System.out.println("Duplicate Value is Entered: " + n);
            }
            System.out.print("Unique Values are: ");
            for (int j : arr) {
                if (j == 0) {
                    continue;
                }
                System.out.print(j + " ");
            }
            System.out.println();
        } while (i < 5);
    }
}
```

Output:



```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/J
Enter the value for element 1: 34
Unique Values are: 34
Enter the value for element 2: 23
Unique Values are: 34 23
Enter the value for element 3: 1
Number must be in the range 10 - 100
Enter the value for element 3: 101
Number must be in the range 10 - 100
Enter the value for element 3: 34
Duplicate Value is Entered: 34
Unique Values are: 34 23
Enter the value for element 3: 45
Unique Values are: 34 23 45
Enter the value for element 4: 76
Unique Values are: 34 23 45 76
Enter the value for element 5: 98
Unique Values are: 34 23 45 76 98

Process finished with exit code 0
```

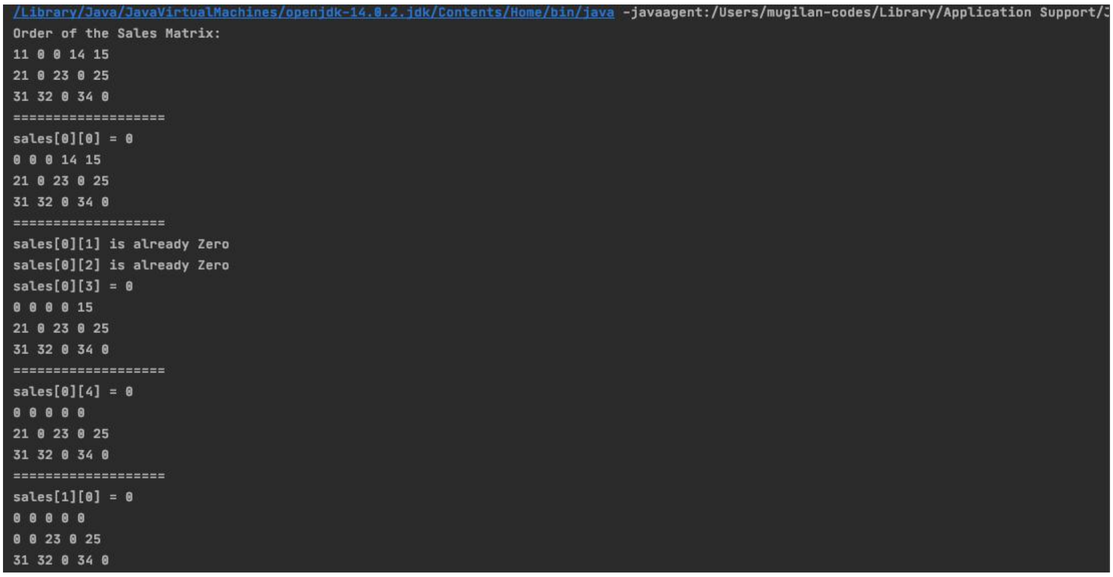
4. Order of Matrix

OrderOfMatrix.java

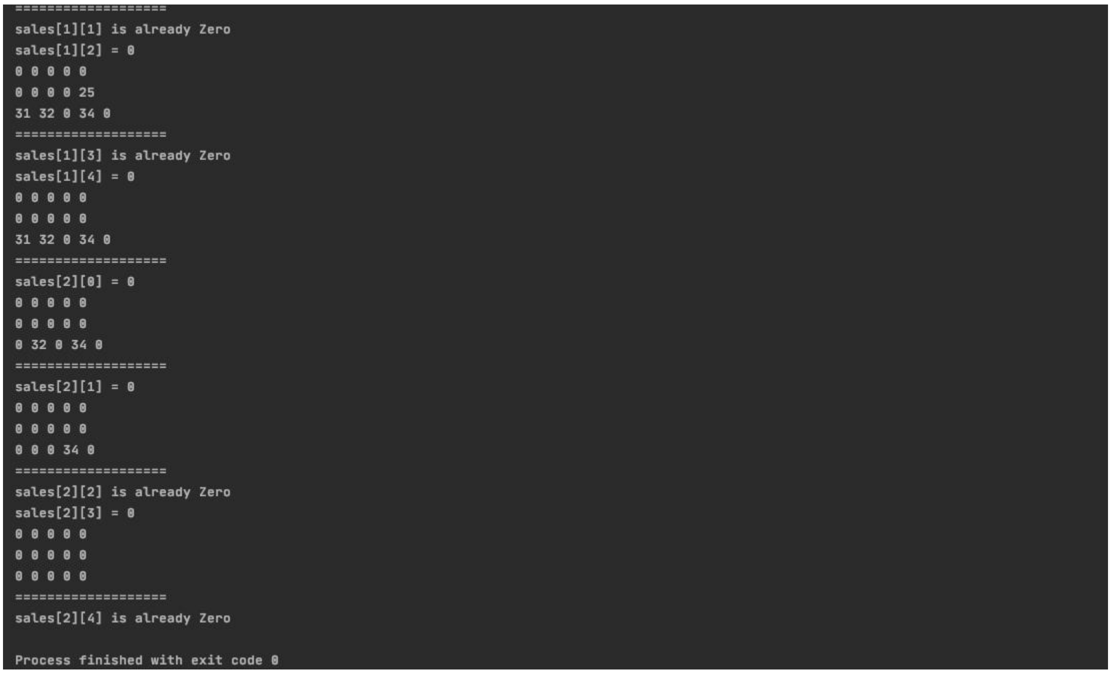
```java
package lab.three;
public class OrderOfMatrix {
    public static void displayArray(int[][] array) {
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 5; j++) {
                System.out.print(array[i][j] + " ");
            }
            System.out.println("");
        }
        System.out.println("=====================");
    }
    public static void main(String[] args) {
        int[][] sales = {
                {11, 0, 0, 14, 15},
                {21, 0, 23, 0, 25},
                {31, 32, 0, 34, 0}
        };
        System.out.println("Order of the Sales Matrix:");
        displayArray(sales);
        for (int row = 0; row < sales.length; row++) {
            for (int col = 0; col < sales[row].length; col++) {
                if (sales[row][col] == 0) {
                    System.out.println("sales[" + row + "][" + col + "] is already Zero");
                    continue;
                }
                sales[row][col] = 0;
                System.out.println("sales[" + row + "][" + col + "] = 0");
                displayArray(sales);
```

```
            }
        }
    }
}
```

Output:



```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/J
Order of the Sales Matrix:
11 0 0 14 15
21 0 23 0 25
31 32 0 34 0
===================
sales[0][0] = 0
0 0 0 14 15
21 0 23 0 25
31 32 0 34 0
===================
sales[0][1] is already Zero
sales[0][2] is already Zero
sales[0][3] = 0
0 0 0 0 15
21 0 23 0 25
31 32 0 34 0
===================
sales[0][4] = 0
0 0 0 0 0
21 0 23 0 25
31 32 0 34 0
===================
sales[1][0] = 0
0 0 0 0 0
0 0 23 0 25
31 32 0 34 0
```



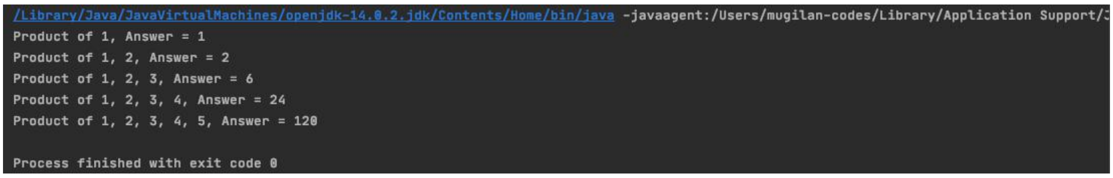
```
===================
sales[1][1] is already Zero
sales[1][2] = 0
0 0 0 0 0
0 0 0 0 25
31 32 0 34 0
===================
sales[1][3] is already Zero
sales[1][4] = 0
0 0 0 0 0
0 0 0 0 0
31 32 0 34 0
===================
sales[2][0] = 0
0 0 0 0 0
0 0 0 0 0
0 32 0 34 0
===================
sales[2][1] = 0
0 0 0 0 0
0 0 0 0 0
0 0 0 34 0
===================
sales[2][2] is already Zero
sales[2][3] = 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
===================
sales[2][4] is already Zero

Process finished with exit code 0
```

5. Variable-Length Argument List

VariableLength.java

```java
package lab.three;
public class VariableLength {
    private static void product(int... args) {
        int prod = 1;
        System.out.print("Product of ");
        for (int i : args) {
            System.out.print(i + ", ");
            prod *= i;
        }
        System.out.println("Answer = " + prod);
    }
    public static void main(String[] args) {
        product(1);
        product(1, 2);
        product(1, 2, 3);
        product(1, 2, 3, 4);
        product(1, 2, 3, 4, 5);
    }
}
```

Output:



```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/
Product of 1, Answer = 1
Product of 1, 2, Answer = 2
Product of 1, 2, 3, Answer = 6
Product of 1, 2, 3, 4, Answer = 24
Product of 1, 2, 3, 4, 5, Answer = 120

Process finished with exit code 0
```

6. Using the Enhanced for statement

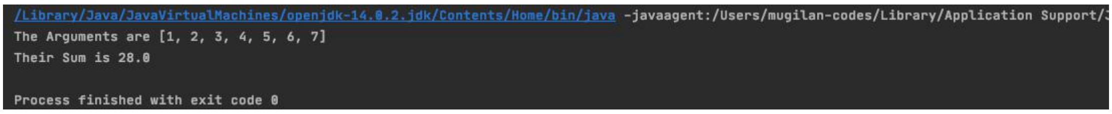
CommandLineArgs.java

```java
package lab.three;
import java.util.Arrays;
public class CommandLineArgs {
    public static void main(String[] args) {
        double answer = 0.0;
        for(String str: args) {
            answer += Double.parseDouble(str);
        }
        System.out.println("The Arguments are "+ Arrays.toString(args));
        System.out.println("Their Sum is " + answer);
    }
}
```

Output:

> javac CommandLineArgs.java

> java CommandLineArgs 1 2 3 4 5 6 7



```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/J
The Arguments are [1, 2, 3, 4, 5, 6, 7]
Their Sum is 28.0

Process finished with exit code 0
```

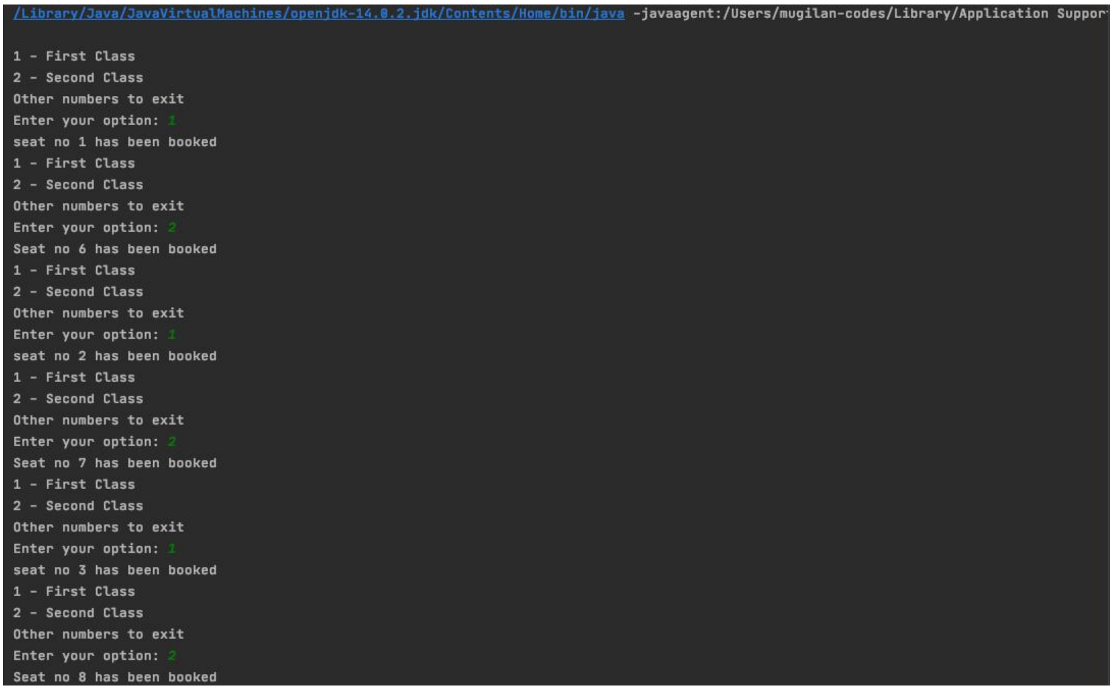
7. Airline Reservation System

Airlines.java

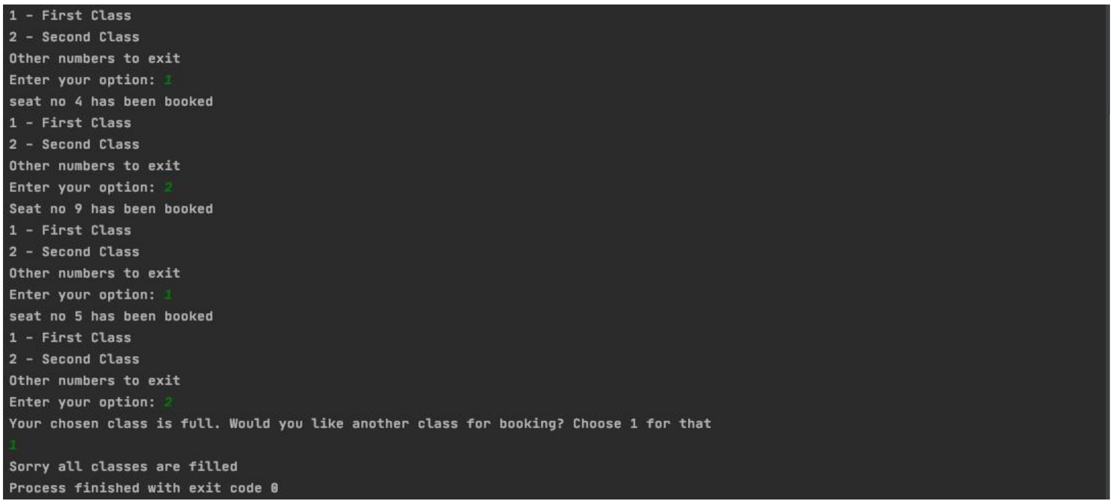
```
package lab.three;
import java.util.Scanner;
class Book {
    Scanner sc = new Scanner(System.in);
    boolean secondClass(boolean[] a) {
        for (int i = 6; i < 10; i++) {
            if (!a[i]) {
                a[i] = true;
                System.out.print("Seat no " + i + " has been booked");
                return true;
            }
        }
        return false;
    }
    boolean firstClass(boolean[] a) {
        for (int i = 1; i <= 5; i++) {
            if (!a[i]) {
                a[i] = true;
                System.out.print("seat no " + i + " has been booked");
                return true;
            }
        }
        return false;
    }
    boolean check(int n, boolean[] a) {
        if (n >= 1 && n <= 2) {
            if (n == 2) {
                boolean bookSecondClass = secondClass(a);
                if (bookSecondClass) {
                    return true;
                } else {
                    System.out.println("Your chosen class is full. Would you " +
                            "like another class for booking? Choose 1 for " +
                            "that");
                    int option = sc.nextInt();
                    if (option == 1) {
                        boolean bookFirstClass = firstClass(a);
                        if (bookFirstClass) {
                            return true;
                        } else {
                            System.out.print("Sorry all classes are filled");
                            return false;
                        }
                    } else {
                        System.out.println("Next flight arrives in 3 hours");
                    }
                }
            } else {
                boolean bookFirstClass = firstClass(a);
                if (bookFirstClass) {
                    return true;
                } else {
                    System.out.println("Your chosen class is full. Would you " +
                            "like another class for booking? Choose 2 for " +
                            "that");
                    int option = sc.nextInt();
                    if (option == 2) {
                        boolean bookSecondClass = secondClass(a);
                        if (bookSecondClass) {
                            return true;
                        } else {
                            System.out.print("Sorry all classes are filled");
                            return false;
                        }
                    } else {
                        System.out.println("Next flight arrives in 3 hours");
                        return false;
                    }
                }
            }
        }
        System.out.println("you chose wrong option");
        return false;
    }
}
```

```
public class Airlines {
    public static void main(String[] args) {
        boolean[] a = new boolean[10];
        Scanner sc = new Scanner(System.in);
        Book book = new Book();
        while (true) {
            System.out.println("\n1 - First Class\n2 - Second Class\nOther " +
                    "numbers to exit");
            System.out.print("Enter your option: ");
            int option = sc.nextInt();
            boolean verify = book.check(option, a);
            if (!verify) {
                break;
            }
        }
    }
}
```

Output:



```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Suppor

1 - First Class
2 - Second Class
Other numbers to exit
Enter your option: 1
seat no 1 has been booked
1 - First Class
2 - Second Class
Other numbers to exit
Enter your option: 2
Seat no 6 has been booked
1 - First Class
2 - Second Class
Other numbers to exit
Enter your option: 1
seat no 2 has been booked
1 - First Class
2 - Second Class
Other numbers to exit
Enter your option: 2
Seat no 7 has been booked
1 - First Class
2 - Second Class
Other numbers to exit
Enter your option: 1
seat no 3 has been booked
1 - First Class
2 - Second Class
Other numbers to exit
Enter your option: 2
Seat no 8 has been booked
```



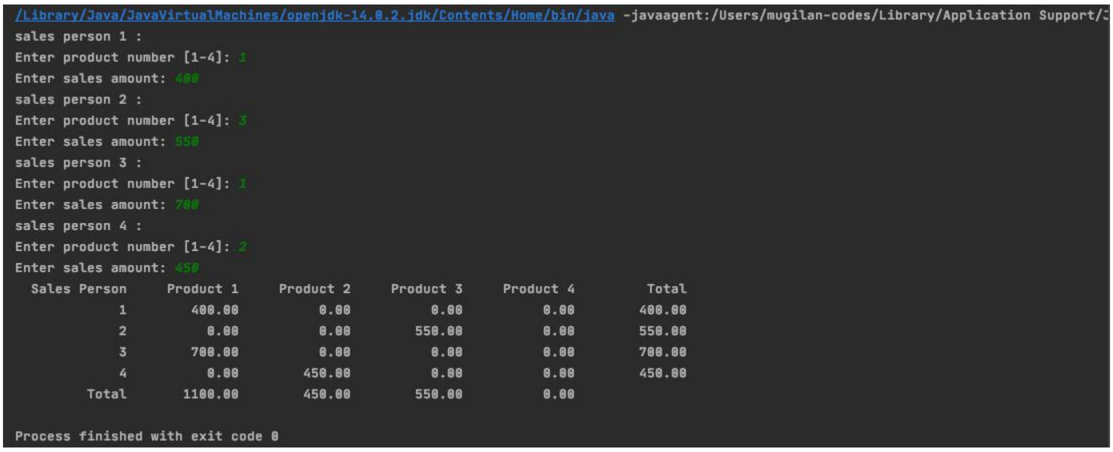
```
1 - First Class
2 - Second Class
Other numbers to exit
Enter your option: 1
seat no 4 has been booked
1 - First Class
2 - Second Class
Other numbers to exit
Enter your option: 2
Seat no 9 has been booked
1 - First Class
2 - Second Class
Other numbers to exit
Enter your option: 1
seat no 5 has been booked
1 - First Class
2 - Second Class
Other numbers to exit
Enter your option: 2
Your chosen class is full. Would you like another class for booking? Choose 1 for that
1
Sorry all classes are filled
Process finished with exit code 0
```

8. Total Sales

TotalSales.java

```
package lab.three;
import java.util.Scanner;
class SalesPerson {
    public void Sales() {
        Scanner input = new Scanner(System.in);
        double[][] sales = new double[5][4];
        for (int x = 0; x < 4; x++) {
            System.out.printf("sales person %d :\n", x + 1);
            int salesPerson = x + 1;
            System.out.print("Enter product number [1-4]: ");
            int p = input.nextInt();
            System.out.print("Enter sales amount: ");
            double amt = input.nextDouble();
            if (salesPerson >= 1 && salesPerson < 5 && p >= 1 && p < 5 && amt >= 0) {
                sales[salesPerson - 1][p - 1] += amt;
            }
            if (p > 4)
                System.out.print("Invalid input!\n");
        }
        double[] personTotal = new double[4];
        int col1;
        for (col1 = 0; col1 < 4; col1++)
            personTotal[col1] = 0;
        System.out.printf("%14s%14s%14s%14s%14s%14s\n", "Sales Person",
                "Product 1",
                "Product 2", "Product 3", "Product 4", "Total");
        int row;
        for (row = 0; row < 4; row++) {
            double productTotal = 0.0;
            System.out.printf("%14d", (row + 1));
            for (int col = 0; col < 4; col++) {
                System.out.printf("%14.2f", sales[row][col]);
                productTotal += sales[row][col];
                personTotal[col] += sales[row][col];
            }
            System.out.printf("%14.2f\n", productTotal);
        }
        System.out.printf("%14s", "Total");
        for (int col = 0; col < 4; col++)
            System.out.printf("%14.2f", personTotal[col]);
        System.out.println();
    }
}
public class TotalSales {
    public static void main(String[] args) {
        SalesPerson s = new SalesPerson();
        s.Sales();
    }
}
```

Output:



```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/
sales person 1 :
Enter product number [1-4]: 1
Enter sales amount: 400
sales person 2 :
Enter product number [1-4]: 3
Enter sales amount: 550
sales person 3 :
Enter product number [1-4]: 1
Enter sales amount: 700
sales person 4 :
Enter product number [1-4]: 2
Enter sales amount: 450
  Sales Person     Product 1     Product 2     Product 3     Product 4         Total
             1        400.00          0.00          0.00          0.00        400.00
             2          0.00          0.00        550.00          0.00        550.00
             3        700.00          0.00          0.00          0.00        700.00
             4          0.00        450.00          0.00          0.00        450.00
         Total       1100.00        450.00        550.00          0.00

Process finished with exit code 0
```

Source Code:

https://github.com/Mugilan-Codes/java-lab-exercises