

Lab Report 4

Submitted to:

Shakib Mahmud Dipto
Faculty

Submitted by:

Sumaiya Akter

ID: 201014071

Department of CSE
Summer'24

Course code: CSE 2104

Course Title: Object Oriented Programming Lab

Section: 01

University of Liberal Arts Bangladesh

July 07, 2024

Practice Problem 01

Code Explanation : involves implementing three sorting algorithms: Bubble Sort, Selection Sort, and Merge Sort. Bubble Sort repeatedly swaps adjacent elements if they are in the wrong order. Selection Sort selects the smallest element from an unsorted array and swaps it with the first element. Merge Sort divides the array into halves, recursively sorts them, and then merges the sorted halves. Each algorithm is demonstrated with sample code.

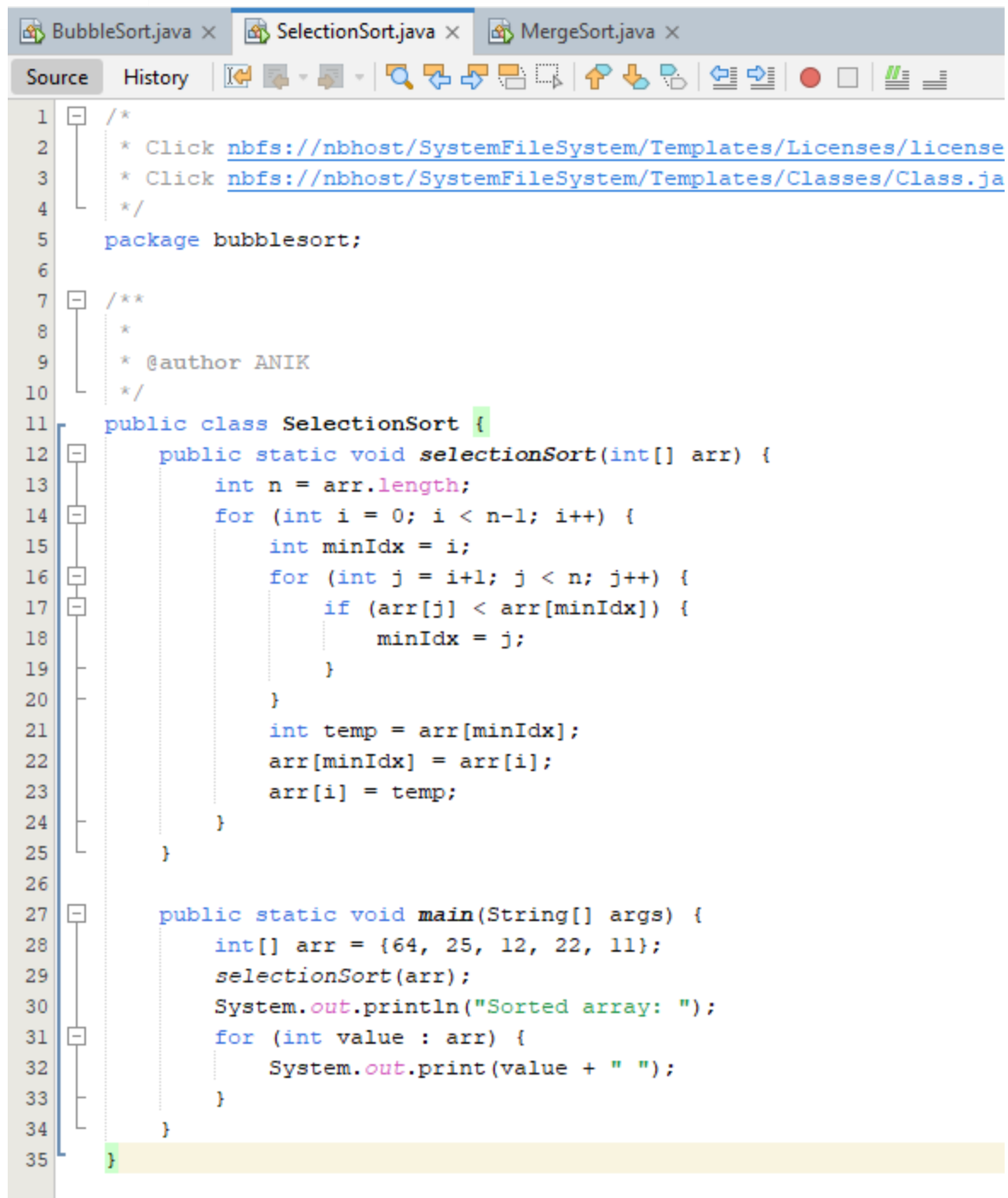
Code Screenshot :

Input:

```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
3   */
4
5   package bubblesort;
6
7   /**
8    *
9    * @author ANIK
10   */
11  public class BubbleSort {
12      public static void bubbleSort(int[] arr) {
13          int n = arr.length;
14          for (int i = 0; i < n-1; i++) {
15              for (int j = 0; j < n-i-1; j++) {
16                  if (arr[j] > arr[j+1]) {
17                      // swap arr[j] and arr[j+1]
18                      int temp = arr[j];
19                      arr[j] = arr[j+1];
20                      arr[j+1] = temp;
21                  }
22              }
23          }
24      }
25
26      public static void main(String[] args) {
27          int[] arr = {64, 25, 12, 22, 11};
28          bubbleSort(arr);
29          System.out.println("Sorted array: ");
30          for (int value : arr) {
31              System.out.print(value + " ");
32          }
33      }
34  }

```



The screenshot shows an IDE with three tabs: BubbleSort.java, SelectionSort.java (active), and MergeSort.java. The SelectionSort.java file contains the following Java code:

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java
4   */
5   package bubblesort;
6
7   /**
8    *
9    * @author ANIK
10   */
11  public class SelectionSort {
12      public static void selectionSort(int[] arr) {
13          int n = arr.length;
14          for (int i = 0; i < n-1; i++) {
15              int minIdx = i;
16              for (int j = i+1; j < n; j++) {
17                  if (arr[j] < arr[minIdx]) {
18                      minIdx = j;
19                  }
20              }
21              int temp = arr[minIdx];
22              arr[minIdx] = arr[i];
23              arr[i] = temp;
24          }
25      }
26
27      public static void main(String[] args) {
28          int[] arr = {64, 25, 12, 22, 11};
29          selectionSort(arr);
30          System.out.println("Sorted array: ");
31          for (int value : arr) {
32              System.out.print(value + " ");
33          }
34      }
35  }
```

Department of Computer Science & Engineering

UNIVERSITY OF LIBERAL ARTS BANGLADESH

```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit
4   */
5   package bubblesort;
6
7   /**
8    *
9    * @author ANIK
10   */
11  public class MergeSort {
12      public static void mergeSort(int[] arr, int left, int right) {
13          if (left < right) {
14              int mid = (left + right) / 2;
15
16              mergeSort(arr, left, mid);
17              mergeSort(arr, mid + 1, right);
18
19              merge(arr, left, mid, right);
20          }
21      }
22
23      public static void merge(int[] arr, int left, int mid, int right) {
24          int n1 = mid - left + 1;
25          int n2 = right - mid;
26
27          int[] leftArray = new int[n1];
28          int[] rightArray = new int[n2];
29
30          for (int i = 0; i < n1; ++i)
31              leftArray[i] = arr[left + i];
32          for (int j = 0; j < n2; ++j)
33              rightArray[j] = arr[mid + 1 + j];
34
35          int i = 0, j = 0;
36          int k = left;
37          while (i < n1 && j < n2) {
38              if (leftArray[i] <= rightArray[j]) {
39                  arr[k] = leftArray[i];
40                  i++;
41              } else {
42                  arr[k] = rightArray[j];
43                  j++;
44              }
45              k++;
46          }
47      }

```

```
48 while (i < n1) {  
49     arr[k] = leftArray[i];  
50     i++;  
51     k++;  
52 }  
53  
54 while (j < n2) {  
55     arr[k] = rightArray[j];  
56     j++;  
57     k++;  
58 }  
59 }  
60  
61 public static void main(String[] args) {  
62     int[] arr = {64, 25, 12, 22, 11};  
63     mergeSort(arr, 0, arr.length - 1);  
64     System.out.println("Sorted array: ");  
65     for (int value : arr) {  
66         System.out.print(value + " ");  
67     }  
68 }  
69 }
```

Output:
BubbleSort

```
Output - Run (BubbleSort) ×
cd F:\sumaiya the V.I.P\sem 14\OOP\lab\submit\lab report 4\BubbleSort; "JAVA_HOME=C:\Pr
Scanning for projects...

-----< com.mycompany:BubbleSort >-----
Building BubbleSort 1.0-SNAPSHOT
  from pom.xml
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ BubbleSort ---
- skip non existing resourceDirectory F:\sumaiya the V.I.P\sem 14\OOP\lab\submit\lab report

--- compiler:3.11.0:compile (default-compile) @ BubbleSort ---
- Nothing to compile - all classes are up to date

--- exec:3.1.0:exec (default-cli) @ BubbleSort ---
Sorted array:
11 12 22 25 64

BUILD SUCCESS

Total time: 0.543 s
Finished at: 2024-07-14T11:22:01+06:00
```

SelectionSort

```
Output - Run (BubbleSort) x
cd F:\sumaiya the V.I.P\sem 14\OOP\lab\submit\lab report 4\BubbleSort; "JAVA_HOME"
Scanning for projects...

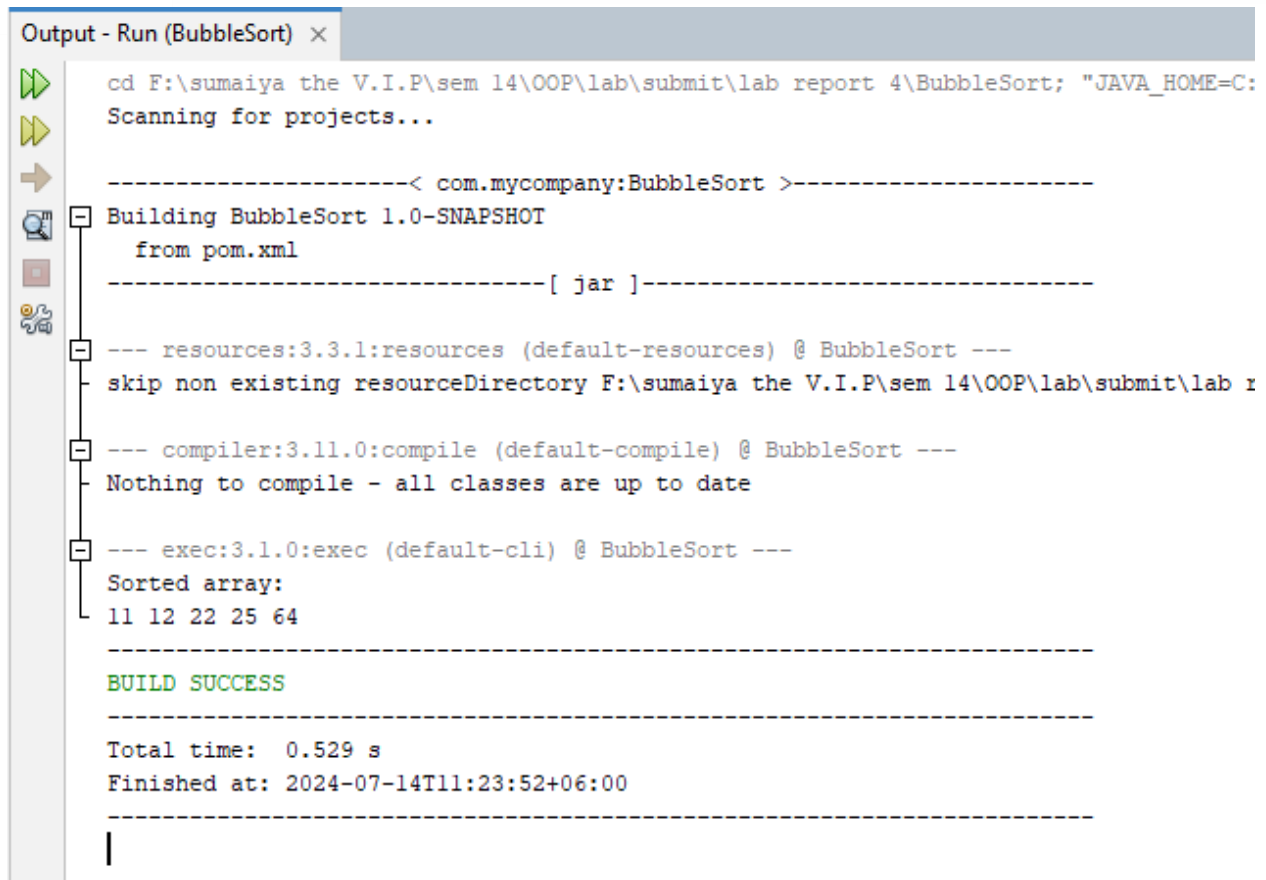
-----< com.mycompany:BubbleSort >-----
Building BubbleSort 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ BubbleSort ---
skip non existing resourceDirectory F:\sumaiya the V.I.P\sem 14\OOP\lab\submit\lab report 4\BubbleSort
--- compiler:3.11.0:compile (default-compile) @ BubbleSort ---
Nothing to compile - all classes are up to date
--- exec:3.1.0:exec (default-cli) @ BubbleSort ---
Sorted array:
11 12 22 25 64

BUILD SUCCESS

Total time: 0.541 s
Finished at: 2024-07-14T11:22:49+06:00
```

MergeSort



```
Output - Run (BubbleSort) x
cd F:\sumaiya the V.I.P\sem 14\OOP\lab\submit\lab report 4\BubbleSort; "JAVA_HOME=C:
Scanning for projects...

-----< com.mycompany:BubbleSort >-----
Building BubbleSort 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ BubbleSort ---
skip non existing resourceDirectory F:\sumaiya the V.I.P\sem 14\OOP\lab\submit\lab r

--- compiler:3.11.0:compile (default-compile) @ BubbleSort ---
Nothing to compile - all classes are up to date

--- exec:3.1.0:exec (default-cli) @ BubbleSort ---
Sorted array:
11 12 22 25 64

BUILD SUCCESS

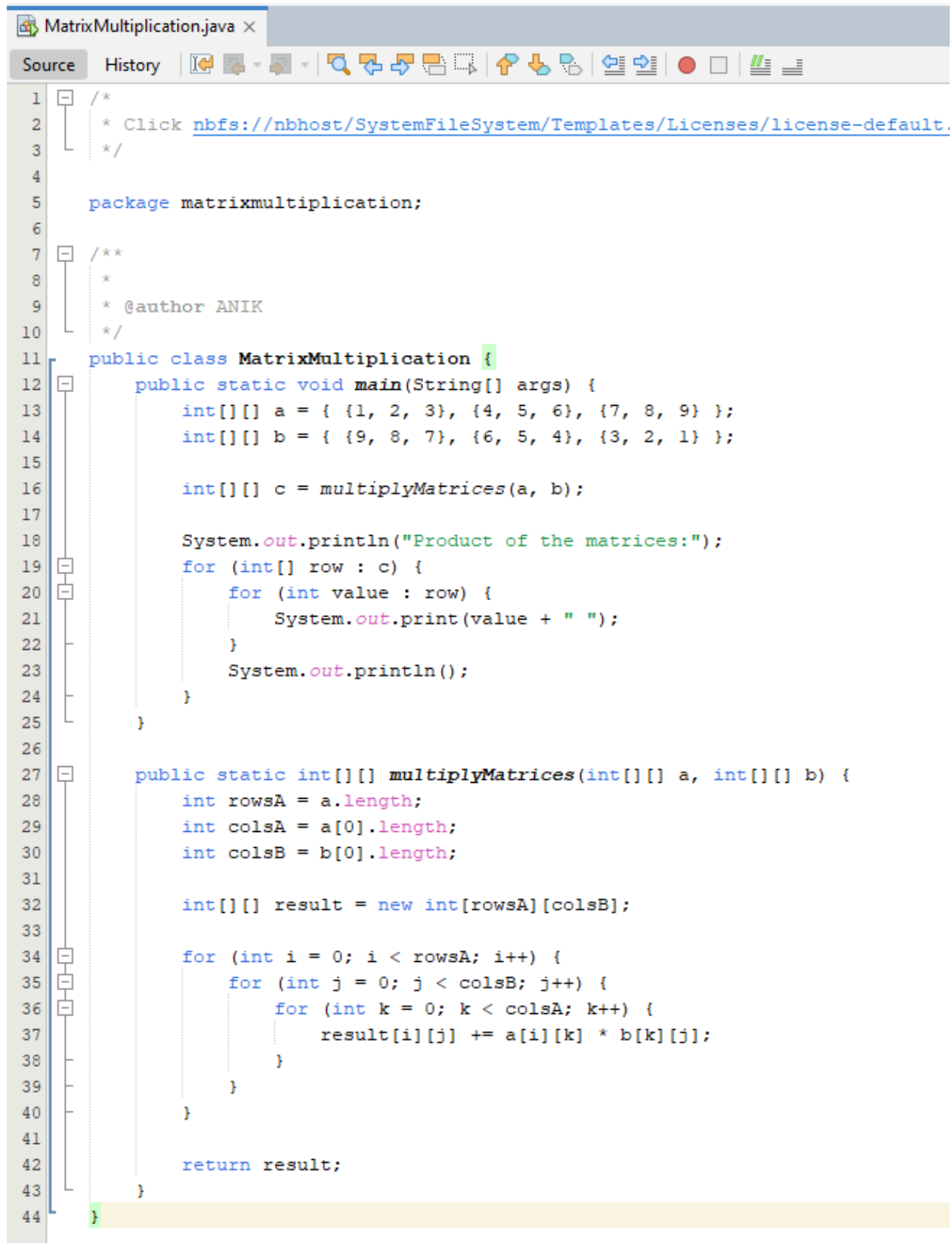
Total time: 0.529 s
Finished at: 2024-07-14T11:23:52+06:00
```

Practice Problem 02

Code Explanation: involves multiplying two matrices. The program defines two matrices and multiplies them by iterating over their rows and columns. For each element in the result matrix, it calculates the sum of the product of corresponding elements from the two input matrices. The resulting product matrix is then displayed. This demonstrates matrix multiplication using nested loops in Java.

Code Screenshot:

Input:



```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.
3   */
4
5   package matrixmultiplication;
6
7   /**
8    *
9    * @author ANIK
10   */
11  public class MatrixMultiplication {
12      public static void main(String[] args) {
13          int[][] a = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9} };
14          int[][] b = { {9, 8, 7}, {6, 5, 4}, {3, 2, 1} };
15
16          int[][] c = multiplyMatrices(a, b);
17
18          System.out.println("Product of the matrices:");
19          for (int[] row : c) {
20              for (int value : row) {
21                  System.out.print(value + " ");
22              }
23              System.out.println();
24          }
25      }
26
27      public static int[][] multiplyMatrices(int[][] a, int[][] b) {
28          int rowsA = a.length;
29          int colsA = a[0].length;
30          int colsB = b[0].length;
31
32          int[][] result = new int[rowsA][colsB];
33
34          for (int i = 0; i < rowsA; i++) {
35              for (int j = 0; j < colsB; j++) {
36                  for (int k = 0; k < colsA; k++) {
37                      result[i][j] += a[i][k] * b[k][j];
38                  }
39              }
40          }
41
42          return result;
43      }
44  }
```

Output:

```

Output - Run (MatrixMultiplication) x
cd F:\sumaiya the V.I.P\sem 14\OOP\lab\submit\lab report 4\practice problem 02\Matri
Scanning for projects...

-----< com.mycompany:MatrixMultiplication >-----
Building MatrixMultiplication 1.0-SNAPSHOT
  from pom.xml
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ MatrixMultiplication ---
skip non existing resourceDirectory F:\sumaiya the V.I.P\sem 14\OOP\lab\submit\lab r

--- compiler:3.11.0:compile (default-compile) @ MatrixMultiplication ---
Nothing to compile - all classes are up to date

--- exec:3.1.0:exec (default-cli) @ MatrixMultiplication ---
Product of the matrices:
30 24 18
84 69 54
138 114 90

BUILD SUCCESS

Total time: 0.547 s
Finished at: 2024-07-14T12:12:53+06:00
|

```

Practice Problem 03

Code Explanation: involves managing a collection of books using both ArrayList and LinkedList. Each book has a title, author, year, and genre. The program demonstrates adding books to the collections, displaying their details, and removing a book from each collection. The ArrayList and LinkedList classes provide dynamic data structures with methods for adding, accessing, and removing elements, illustrating their usage in managing book data.

Code Screenshot:

Input:

```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to ch
3   */
4
5  package bookcollection;
6  import java.util.ArrayList;
7  import java.util.LinkedList;
8  /**
9   *
10   * @author ANIK
11   */
12  public class BookCollection {
13      public static void main(String[] args) {
14          // Using ArrayList
15          ArrayList<Book> bookList = new ArrayList<>();
16          bookList.add(new Book("Book1", "Author1", 2001, "Genrel"));
17          bookList.add(new Book("Book2", "Author2", 2002, "Genre2"));
18          bookList.add(new Book("Book3", "Author3", 2003, "Genre3"));
19
20          System.out.println("Books in ArrayList:");
21          for (Book book : bookList) {
22              book.displayDetails();
23          }
24
25          bookList.remove(1);
26          System.out.println("After removal in ArrayList:");
27          for (Book book : bookList) {
28              book.displayDetails();
29          }
30
31          // Using LinkedList
32          LinkedList<Book> bookLinkedList = new LinkedList<>();
33          bookLinkedList.add(new Book("Book1", "Author1", 2001, "Genrel"));
34          bookLinkedList.add(new Book("Book2", "Author2", 2002, "Genre2"));
35          bookLinkedList.add(new Book("Book3", "Author3", 2003, "Genre3"));
36
37          System.out.println("Books in LinkedList:");
38          for (Book book : bookLinkedList) {
39              book.displayDetails();
40          }
41
42          bookLinkedList.remove(1);
43          System.out.println("After removal in LinkedList:");
44          for (Book book : bookLinkedList) {
45              book.displayDetails();
46          }
47      }
48  }

```

Department of Computer Science & Engineering

UNIVERSITY OF LIBERAL ARTS BANGLADESH

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package bookcollection;
6
7  /**
8   *
9   * @author ANIK
10  */
11  public class Book{
12      String title;
13      String author;
14      int year;
15      String genre;
16
17      Book(String title, String author, int year, String genre) {
18          this.title = title;
19          this.author = author;
20          this.year = year;
21          this.genre = genre;
22      }
23
24      void displayDetails() {
25          System.out.println("Title: " + title + ", Author: " + author + ", Year: " + year + ", Genre: " + genre);
26      }
27  }
28  }
```

Output:

```

Output - Run (BookCollection) x
cd F:\sumaiya the V.I.P\sem 14\OOP\lab\submit\lab report 4\practice problem 03\BookCollection; '
Scanning for projects...

-----< com.mycompany:BookCollection >-----
Building BookCollection 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ BookCollection ---
skip non existing resourceDirectory F:\sumaiya the V.I.P\sem 14\OOP\lab\submit\lab report 4\pra

--- compiler:3.11.0:compile (default-compile) @ BookCollection ---
Nothing to compile - all classes are up to date

--- exec:3.1.0:exec (default-cli) @ BookCollection ---
Books in ArrayList:
Title: Book1, Author: Author1, Year: 2001, Genre: Genrel
Title: Book2, Author: Author2, Year: 2002, Genre: Genre2
Title: Book3, Author: Author3, Year: 2003, Genre: Genre3
After removal in ArrayList:
Title: Book1, Author: Author1, Year: 2001, Genre: Genrel
Title: Book3, Author: Author3, Year: 2003, Genre: Genre3
Books in LinkedList:
Title: Book1, Author: Author1, Year: 2001, Genre: Genrel
Title: Book2, Author: Author2, Year: 2002, Genre: Genre2
Title: Book3, Author: Author3, Year: 2003, Genre: Genre3
After removal in LinkedList:
Title: Book1, Author: Author1, Year: 2001, Genre: Genrel
Title: Book3, Author: Author3, Year: 2003, Genre: Genre3

BUILD SUCCESS

Total time: 0.530 s
Finished at: 2024-07-14T12:23:45+06:00
|

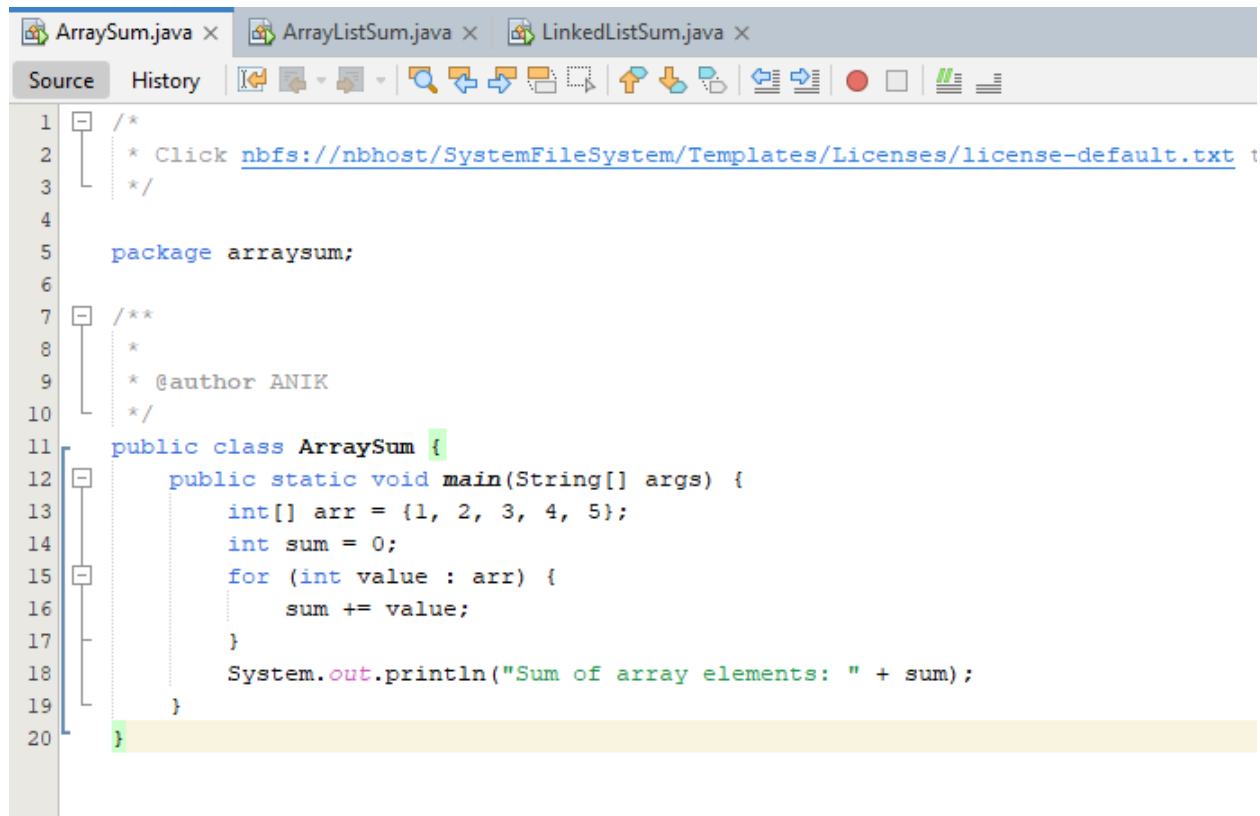
```

Practice problem 04

Code Explanation: involves implementing a common task (e.g., summing elements) using arrays, ArrayList, and LinkedList. Each method demonstrates how to iterate through the data structure and accumulate the sum of its elements. Arrays provide a fixed-size structure, while ArrayList and LinkedList offer dynamic sizing and different performance characteristics. This comparison highlights the versatility and specific use cases of each data structure in Java.

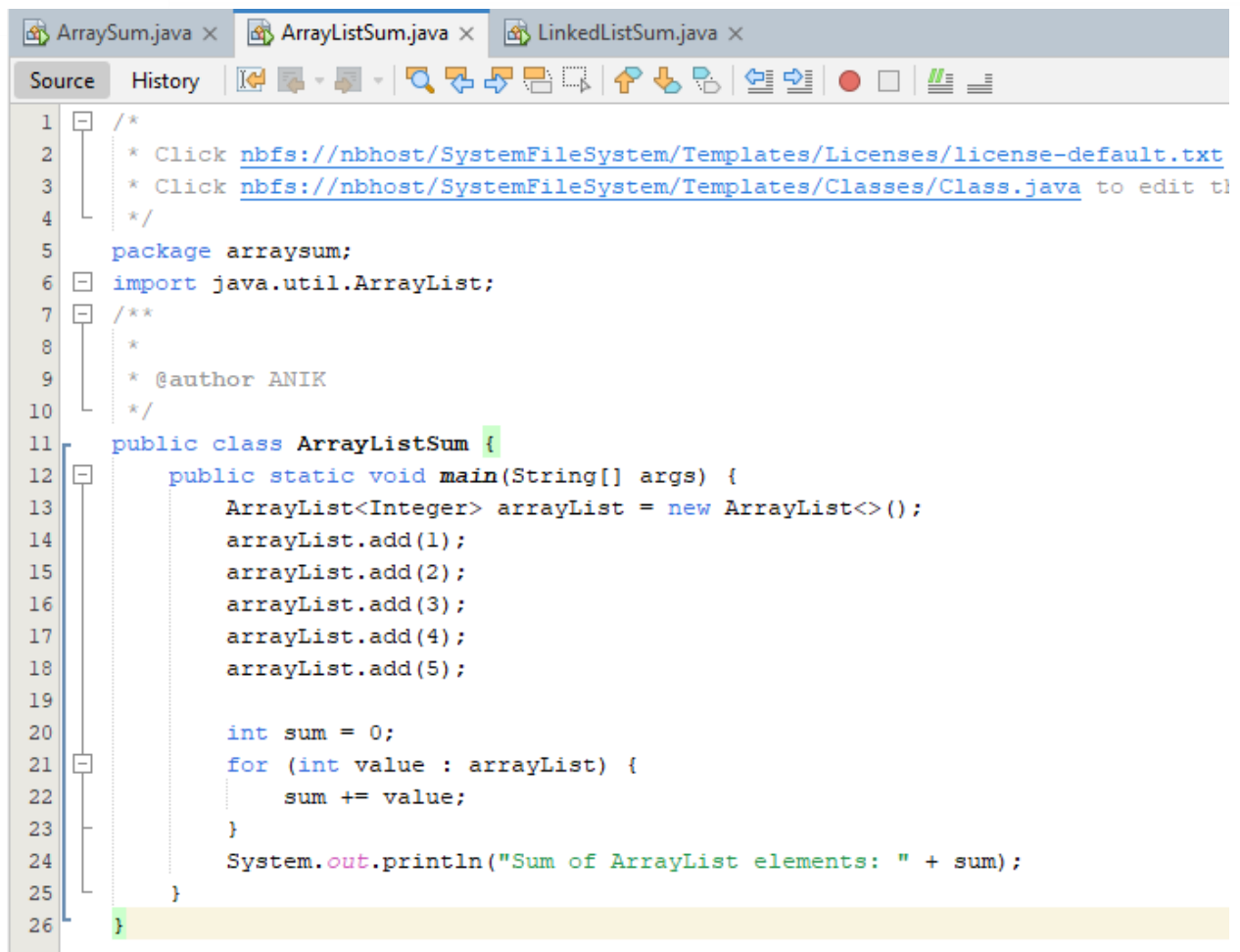
Code Screenshot:

Input:

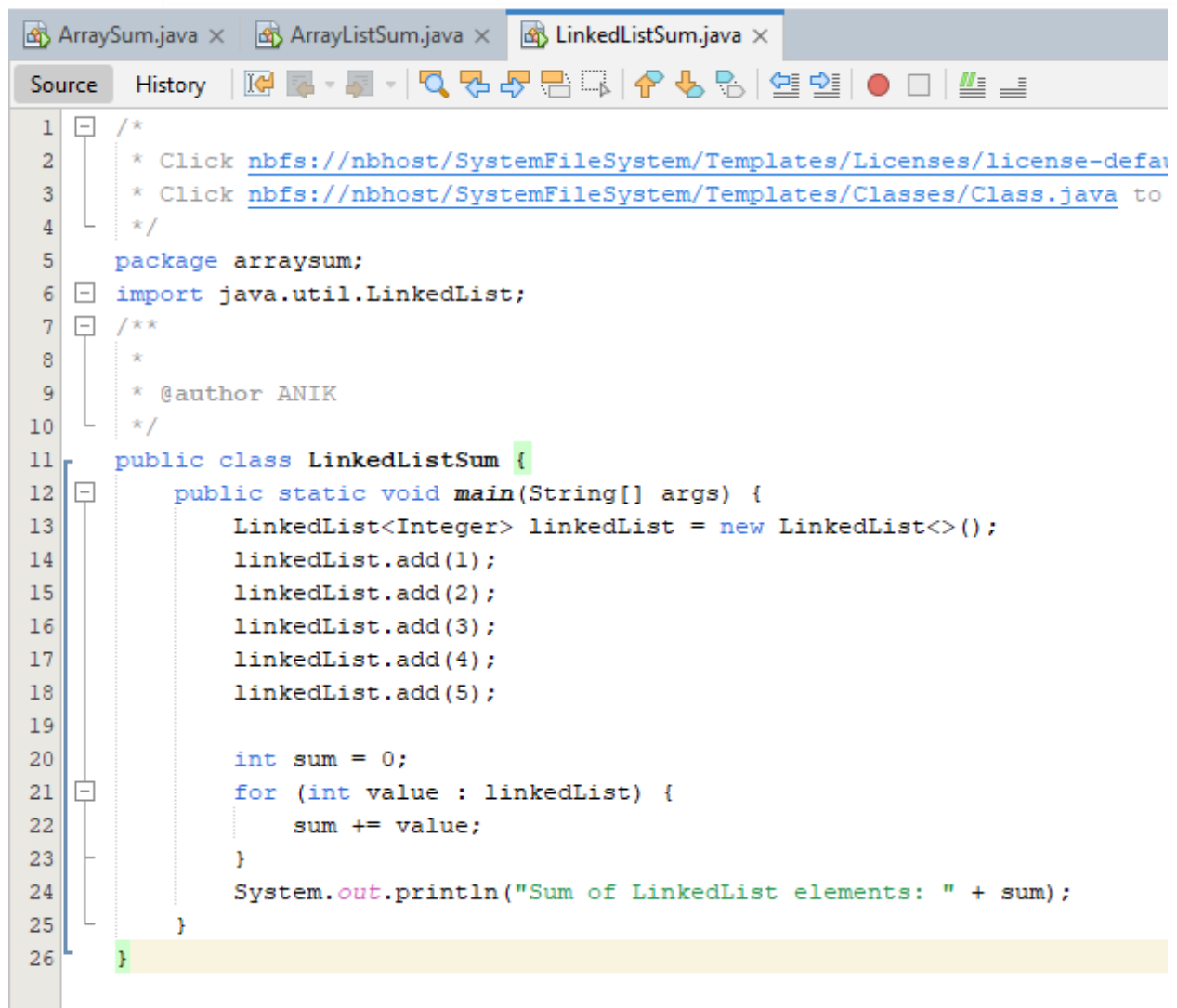


```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
3   *
4   */
5
6   package arraysum;
7
8   /**
9    *
10   * @author ANIK
11   */
12   public class ArraySum {
13       public static void main(String[] args) {
14           int[] arr = {1, 2, 3, 4, 5};
15           int sum = 0;
16           for (int value : arr) {
17               sum += value;
18           }
19           System.out.println("Sum of array elements: " + sum);
20       }
```

Department of Computer Science & Engineering
UNIVERSITY OF LIBERAL ARTS
BANGLADESH



```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit t
4   */
5  package arraysum;
6  import java.util.ArrayList;
7  /**
8   *
9   * @author ANIK
10  */
11 public class ArrayListSum {
12     public static void main(String[] args) {
13         ArrayList<Integer> arrayList = new ArrayList<>();
14         arrayList.add(1);
15         arrayList.add(2);
16         arrayList.add(3);
17         arrayList.add(4);
18         arrayList.add(5);
19
20         int sum = 0;
21         for (int value : arrayList) {
22             sum += value;
23         }
24         System.out.println("Sum of ArrayList elements: " + sum);
25     }
26 }
```

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-defa
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
4   */
5   package arraysum;
6   import java.util.LinkedList;
7   /**
8    *
9    * @author ANIK
10   */
11  public class LinkedListSum {
12      public static void main(String[] args) {
13          LinkedList<Integer> linkedList = new LinkedList<>();
14          linkedList.add(1);
15          linkedList.add(2);
16          linkedList.add(3);
17          linkedList.add(4);
18          linkedList.add(5);
19
20          int sum = 0;
21          for (int value : linkedList) {
22              sum += value;
23          }
24          System.out.println("Sum of LinkedList elements: " + sum);
25      }
26  }
```

Output:

Output - Run (ArraySum) ×

```
cd F:\sumaiya the V.I.P\sem 14\OOP\lab\submit\lab report 4\practice problem 04\Array
Scanning for projects...

-----< com.mycompany:ArraySum >-----
Building ArraySum 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ ArraySum ---
skip non existing resourceDirectory F:\sumaiya the V.I.P\sem 14\OOP\lab\submit\lab r

--- compiler:3.11.0:compile (default-compile) @ ArraySum ---
Changes detected - recompiling the module! :source
Compiling 3 source files with javac [debug target 21] to target\classes

--- exec:3.1.0:exec (default-cli) @ ArraySum ---
Sum of array elements: 15

BUILD SUCCESS

Total time: 0.863 s
Finished at: 2024-07-14T12:30:46+06:00
```