

Name : Siddhant Kotak

Student ID : 202201410

Course : IT 314 Software Engineering

Professor : Saurabh Tiwari

Semester : Autumn 2024

**Lab : Program Inspection, Debugging and Static
Analysis**

Multiply matrices

Given code :

```
//Java program to multiply two matrices
```

```
import java.util.Scanner;
```

```
class MatrixMultiplication
```

```
{
```

```
    public static void main(String args[])
```

```
    {
```

```
        int m, n, p, q, sum = 0, c, d, k;
```

```
        Scanner in = new Scanner(System.in);
```

```
        System.out.println("Enter the number of rows and columns of first matrix");
```

```
        m = in.nextInt();
```

```
        n = in.nextInt();
```

```
        int first[][] = new int[m][n];
```

```
        System.out.println("Enter the elements of first matrix");
```

```
        for ( c = 0 ; c < m ; c++ )
```

```
            for ( d = 0 ; d < n ; d++ )
```

```
                first[c][d] = in.nextInt();
```

```
        System.out.println("Enter the number of rows and columns of second matrix");
```

```
        p = in.nextInt();
```

```
        q = in.nextInt();
```

```
        if ( n != p )
```

```
        System.out.println("Matrices with entered orders can't be multiplied with each other.");
    else
    {
        int second[][] = new int[p][q];
        int multiply[][] = new int[m][q];

        System.out.println("Enter the elements of second matrix");

        for ( c = 0 ; c < p ; c++ )
            for ( d = 0 ; d < q ; d++ )
                second[c][d] = in.nextInt();

        for ( c = 0 ; c < m ; c++ )
        {
            for ( d = 0 ; d < q ; d++ )
            {
                for ( k = 0 ; k < p ; k++ )
                {
                    sum = sum + first[c-1][c-k]*second[k-1][k-d];
                }

                multiply[c][d] = sum;
                sum = 0;
            }
        }

        System.out.println("Product of entered matrices:-");

        for ( c = 0 ; c < m ; c++ )
        {
            for ( d = 0 ; d < q ; d++ )
```

```

        System.out.print(multiply[c][d]+"\\t");

        System.out.print("\\n");
    }
}
}
}

```

Input: Enter the number of rows and columns of first matrix

2 2

Enter the elements of first matrix

1 2 3 4

Enter the number of rows and columns of first matrix

2 2

Enter the elements of first matrix

1 0 1 0

Output: Product of entered matrices:

3 0

7 0

Program Inspection for multiply matrices

1. How many errors are there in the program? Mention the errors you have identified.

- Errors Identified: 4

- Uninitialized/Incorrect Variable Access:

- The program uses `first[c-1][c-k]` and `second[k-1][k-d]` in the multiplication logic, which can lead to `ArrayIndexOutOfBoundsException` when `c` or `k` is 0. Array indices in Java are zero-based, so the valid range for `c` and `k` should start from 0.

- **Logic Error in Matrix Multiplication:**
 - The formula for matrix multiplication is incorrectly implemented. The correct access for multiplication should be `first[c][k]` and `second[k][d]` instead of `first[c-1][c-k]` and `second[k-1][k-d]`. This results in incorrect product values.
- **Potential Logic Error with Initialization of sum:**
 - The sum variable is initialized to 0 at the beginning of the multiplication loop, which is correct; however, it might cause confusion when the logic fails due to the wrong index access.
- **Output Format:**
 - The output is printed without a proper heading for the second matrix input and lacks a clear separation between input sections, which could confuse users.

2. Which category of program inspection would you find more effective?

- **Effective Category:**
 - The **Control-Flow Errors** category was effective in this case. Issues like ensuring loops terminate correctly and checking that array indices are within bounds are crucial for matrix multiplication operations.

3. Which type of error you are not able to identify using the program inspection?

- **Errors Not Identified:**
 - **Logic Errors in Mathematical Computation:** While program inspections can identify syntax errors and potential runtime exceptions, they may not catch logical errors where the implemented algorithm does not perform the intended mathematical operations correctly.

4. Is the program inspection technique worth applicable?

- **Applicability of Program Inspection:**
 - Yes, the program inspection technique is worth applying. It helps identify a range of potential issues before runtime, improving code quality and reliability. Although it may not catch all logical errors, it provides a framework for systematic error checking, which is essential for complex operations like matrix multiplication.

Code Debugging

The screenshot shows an IDE with a Java file named `MatrixMultiplication.java`. The code is a program to multiply two matrices. It prompts the user to enter the elements of the second matrix. The code is as follows:

```
30 {
31     int second[][] = new int[p][q];
32     int multiply[][] = new int[m][q];
33
34     System.out.println("Enter the elements of second matrix");
35
36     for ( c = 0 ; c < p ; c++ )
37         for ( d = 0 ; d < q ; d++ )
38             second[c][d] = in.nextInt();
39
40     for ( c = 0 ; c < m ; c++ )
41     {
42         for ( d = 0 ; d < q ; d++ )
43         {
44             for ( k = 0 ; k < p ; k++ )
45             {
46                 sum = sum + first[c-1][c-k]*second[k-1][k-d];
47             }
48
49             multiply[c][d] = sum;
50             sum = 0;
51         }
52     }
53
54     System.out.println("Product of entered matrices:-");
55 }
```

The console output shows the program terminated with an exception:

```
<terminated> MatrixMultiplication [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (20 Oct 2024, 6:34:58 pm - 6:36:46 pm) [pid: 14152]
1 0
1 0
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index -1 out of bounds for length 2
    at DebuggingCodes/DebugMatrixMultiplication.MatrixMultiplication.main(MatrixMultiplication.java:46)
```

The screenshot shows the package and import statements for the `MatrixMultiplication` class. The code is as follows:

```
1 package DebugMatrixMultiplication;
2 import java.util.Scanner;
3
4 public class MatrixMultiplication {
5
6     public static void main(String[] args) {
7         // TODO Auto-generated method stub
8         int m, n, p, q, sum = 0, c, d, k;
9
10        Scanner in = new Scanner(System.in);
11        System.out.println("Enter the number of rows and columns of first matrix");
12        m = in.nextInt();
13        n = in.nextInt();
14
15        int first[][] = new int[m][n];
16
17        System.out.println("Enter the elements of first matrix");
18
19        for ( c = 0 ; c < m ; c++ )
20            for ( d = 0 ; d < n ; d++ )
21                first[c][d] = in.nextInt();
22
23        System.out.println("Enter the number of rows and columns of second matrix");
24        p = in.nextInt();
25        q = in.nextInt();
26    }
```

```
25     q = in.nextInt();
26
27     if ( n != p )
28         System.out.println("Matrices with entered orders can't be multiplied");
29     else
30     {
31         int second[][] = new int[p][q];
32         int multiply[][] = new int[m][q];
33
34         System.out.println("Enter the elements of second matrix");
35
36         for ( c = 0 ; c < p ; c++ )
37             for ( d = 0 ; d < q ; d++ )
38                 second[c][d] = in.nextInt();
39
40         for ( c = 0 ; c < m ; c++ )
41         {
42             for ( d = 0 ; d < q ; d++ )
43             {
44                 for ( k = 0 ; k < p ; k++ )
45                 {
46                     sum = sum + first[c][k]*second[k][d];
47                 }
48                 multiply[c][d] = sum;
49                 sum = 0;
50             }
51         }
```

```
49         multiply[c][d] = sum;
50         sum = 0;
51     }
52 }
53
54 System.out.println("Product of entered matrices:-");
55
56 for ( c = 0 ; c < m ; c++ )
57 {
58     for ( d = 0 ; d < q ; d++ )
59         System.out.print(multiply[c][d]+" ");
60     System.out.print("\n");
61 }
62 }
63 }
64 }
65 }
66 }
67 }
68 }
```

Console × Problems Debug Shell

terminated> MatrixMultiplication [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (20 Oct 2024, 6:40:12 pm – 6:40:26 pm) [p

```
1 0
1 0
Product of entered matrices:-
3 0
7 0
```

Errors Identified

1. Incorrect Array Indexing:

- **Original Line:** `sum = sum + first[c-1][c-k] * second[k-1][k-d];`
- **Correction:** Change the indices to use `first[c][k]` and `second[k][d]`. This ensures you access the correct elements of the matrices without causing an `ArrayIndexOutOfBoundsException`.

Breakpoints Needed

You can set breakpoints at the following lines for effective debugging:

- **Line 44:** To check how matrix multiplication is performed and ensure that the correct indices are being accessed.

Steps to Fix the Errors

1. **Change the first index access** from `first[c-1][c-k]` to `first[c][k]`.
2. **Change the second index access** from `second[k-1][k-d]` to `second[k][d]`.
3. **Ensure the inner loop iterates over $k < n$** instead of $k < p$, since you want to multiply corresponding elements.

Fixed Code

// Java program to multiply two matrices

```
import java.util.Scanner;
```

```
class MatrixMultiplication {
```

```
    public static void main(String args[]) {
```

```
        int m, n, p, q, sum = 0, c, d, k;
```

```
        Scanner in = new Scanner(System.in);
```

```
        System.out.println("Enter the number of rows and columns of first matrix");
```

```
        m = in.nextInt();
```

```
        n = in.nextInt();
```

```
        int first[][] = new int[m][n];
```

```
        System.out.println("Enter the elements of first matrix");
```

```
        for (c = 0; c < m; c++)
```

```
            for (d = 0; d < n; d++)
```

```
                first[c][d] = in.nextInt();
```

```
        System.out.println("Enter the number of rows and columns of second matrix");
```

```
        p = in.nextInt();
```

```
        q = in.nextInt();
```



```
if (n != p) {  
    System.out.println("Matrices with entered orders can't be multiplied with each other.");  
} else {  
    int second[][] = new int[p][q];  
    int multiply[][] = new int[m][q];  
  
    System.out.println("Enter the elements of second matrix");  
    for (c = 0; c < p; c++)  
        for (d = 0; d < q; d++)  
            second[c][d] = in.nextInt();  
  
    for (c = 0; c < m; c++) {  
        for (d = 0; d < q; d++) {  
            for (k = 0; k < n; k++) { // Fixed index handling  
                sum += first[c][k] * second[k][d]; // Fixed matrix access  
            }  
            multiply[c][d] = sum;  
            sum = 0;  
        }  
    }  
  
    System.out.println("Product of entered matrices:");  
    for (c = 0; c < m; c++) {  
        for (d = 0; d < q; d++)  
            System.out.print(multiply[c][d] + "\t");  
        System.out.print("\n");  
    }  
}  
  
in.close(); // Close the scanner to avoid resource leaks
```

```
}  
}
```

Input and Output

Input:

Enter the number of rows and columns of first matrix

2 2

Enter the elements of first matrix

1 2

3 4

Enter the number of rows and columns of second matrix

2 2

Enter the elements of second matrix

5 6

7 8

Output:

Product of entered matrices:

19 22

43 50