Tutorial of Multi-dimensional array and static method

> Based on the tutorial of "2020F-Java-A" designed by teaching group in SUSTech
>
> Modified (mainly change to markdown file) by ZHU Yueming in 2021. March. 22th
>
> Add before exercise by ZHU Yueming in 2021. Oct. 18th
>
> Update by ZHU Yueming in 2024. Oct. 10th.

Experimental Objective

1. Learn how to use two dimensional arrays.
2. Understand static method and can invoke defined method.

Before Exercise

Run following code and create two dimesional array, then try following three print method.

```
        int[][] arr = new int[3][];
        arr[0] = new int[]{1, 2, 3};
        arr[1] = arr[0];
        arr[2] = new int[]{3, 4, 5, 6};

        System.out.println(Arrays.toString(arr[1]));
        System.out.println(Arrays.toString(arr));
        System.out.println(Arrays.deepToString(arr));
```

Try following three foreach method, and thinking whether the value can change and whether not.

```
        for (int e : arr[1]) {
            System.out.println(e);
            e = 0;
            System.out.println(e);
        }

        for (int[] a : arr) {
            System.out.println(a);
            a = new int[2];
        }
        System.out.println(Arrays.deepToString(arr));

        for (int[] a : arr) {
            for (int i = 0; i < a.length; i++) {
                a[i]++;
            }
        }
        System.out.println(Arrays.deepToString(arr));
```

Exercises

Exercise 1:

Write a program to get students’ grades from their courses and then print the scores and average scores in a grade table.

1. Prompt the user to enter the number of students (less than 10) and the number of courses (less than 10).
2. Prompt the user to enter the course scores for each student. The scores from different courses are entered on separate lines. On each line, there are scores for a course for each student.
3. Print a grade table. The first row shows the course names and the first column shows the student names. The last row shows the average scores of each course and the last column shows the average scores of each students.

Sample input and output:

```
Please enter the number of subjects: 3
Please enter the number of students: 4
32 44 52 32
89 92 80 94
11 22 32 23
       Course1  Course2  Course3   Average
Student1   32      89      11         44.00
Student2   44      92      22         52.67
Student3   52      80      32         54.67
Student4   32      94      23         49.67
Average   40.00   88.75   22.00
```

Exercise 2:

Write a program to calculate the product of two matrices, the formula is:

$$(AB)_{ij} = \sum_{k=1}^{p} a_{ik}b_{kj} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{ip}b_{pj}$$

如下所示：

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \end{bmatrix}$$

$$B = \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \\ b_{3,1} & b_{3,2} \end{bmatrix}$$

$$C = AB = \begin{bmatrix} a_{1,1}b_{1,1} + a_{1,2}b_{2,1} + a_{1,3}b_{3,1}, & a_{1,1}b_{1,2} + a_{1,2}b_{2,2} + a_{1,3}b_{3,2} \\ \\ a_{2,1}b_{1,1} + a_{2,2}b_{2,1} + a_{2,3}b_{3,1}, & a_{2,1}b_{1,2} + a_{2,2}b_{2,2} + a_{2,3}b_{3,2} \end{bmatrix}$$

Sample input and output is:

```
Enter the number of row and column of matrix A:
3 4
Enter the elements of the matrix:
1 2 3 4
1 1 1 1
2 2 2 2
Enter the number of row and column of matrix B:
4 2
Enter the elements of the matrix:
1 2
3 4
5 6
7 8
Product of Matrix is:
[50, 60]
[16, 20]
[32, 40]
```

Exercise 3:

Continue to the exercise 2, write a program to calculate the product of n matrices.

1. Read the number of matrices from user.
2. Read the elements of all the matrices from user. Before the elements of each matrix, the user should input the rows and columns of that matrix.
3. Print the result.

In this exercise, you should complete the code below:

```java
import java.util.Scanner;

public class Exercise3 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Please enter the number of matrix:");
        int n = in.nextInt();
        int[][] a = inputMatrix(in);
        while (n-- > 1) {
            int[][] b = inputMatrix(in);
            //todo: complete the while loop
        }
        in.close();
    }
    private static int[][] inputMatrix(Scanner in) {
        int[][] a ;
        //todo: complete the input matrix
        return a;
    }
    private static int[][] productMatrixs(int[][] a, int[][] b) {
        int[][] c = new int[a.length][b[0].length];
        //todo: complete the process of product
        return c;
    }

}
```

Sample input and output:

```
Please enter the number of matrix:3
Enter the number of row and column of matrix:
3 5
Enter the elements of the matrix:
6 -7 3 -5 1
0 4 8 2 3
3 -2 1 -7 2
Enter the number of row and column of matrix:
5 1
Enter the elements of the matrix:
0
9
-3
4
1
Product of Matrix is:
[-91]
[23]
[-47]
```

```
Enter the number of row and column of matrix:
1 3
Enter the elements of the matrix:
-1 3 9
Product of Matrix is:
[91, -273, -819]
[-23, 69, 207]
[47, -141, -423]
```

Exercise 4:

Complete the method `check9Numbers()` below:

```
public static boolean check9Numbers(int[] array)
```

If the length of array is not 9, return false.

If the elements in array is smaller then 1 or larger then 9, return false.

If the elements of array are 1-9 and each number only appears once, then return true otherwise return false.

Test case:

```
public static void main(String[] args) {
        System.out.println(check9Numbers(new int[]{1, 3, 2, 4, 9, 8, 7, 6,
5}));
        System.out.println(check9Numbers(new int[]{1, 1, 2, 4, 9, 8, 7, 6,
5}));
        System.out.println(check9Numbers(new int[]{2, 1, 2, 4, 9, 10, -1, 6,
5}));
        System.out.println(check9Numbers(new int[]{1, 2, 1, 2, 4, 9, 7, 4, 6,
5, 3}));
}
 public static boolean check9Numbers(int[] array) {
       //todo: complete the method
 }
```

Result:

```
true
false
false
false
```

Exercise 5:

Continue to the Exercise 4, complete following requirements:

Sudoku is a famous mathematical game in which players fill numbers 1–9 in a 9×9 square. The square satisfies that every row and every column contain 1–9 only once. Specially, the square is divided into 9 subsquares, and every subsquares also contains 1–9 only once. Write a program to judge whether a 1–9 square is a Sudoku square.

1. Get a 9×9 square from console.
2. If it is a Sudoku square,print Yes.
3. If it is not a Sudoku square,print No.

Sample input and output:

```
2 9 3 7 1 5 4 8 6
8 6 1 2 4 9 5 3 7
7 4 5 8 6 3 1 9 2
6 7 8 9 2 1 3 4 5
1 3 9 5 7 4 2 6 8
4 5 2 6 3 8 7 1 9
9 2 4 3 8 7 6 5 1
3 8 6 1 5 2 9 7 4
5 1 7 4 9 6 8 2 3
Yes
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
No
1 2 3 1 2 3 1 2 3
4 5 6 4 5 6 4 5 6
7 8 9 7 8 9 7 8 9
1 2 3 1 2 3 1 2 3
4 5 6 4 5 6 4 5 6
7 8 9 7 8 9 7 8 9
1 2 3 1 2 3 1 2 3
4 5 6 4 5 6 4 5 6
7 8 9 7 8 9 7 8 9
No
2 9 3 7 1 5 4 8 6
8 6 1 2 4 9 5 3 7
7 4 5 8 6 1 3 9 2
6 7 8 9 2 3 1 4 5
```

```
1 3 9 5 7 4 2 6 8
4 5 2 6 3 8 7 1 9
9 2 4 3 8 7 6 5 1
3 8 6 1 5 2 9 7 4
5 1 7 4 9 6 8 2 3
No
```