Batch: B1 Roll No.: 1711072

Experiment / assignment / tutorial No. 02

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

TITLE :Multi-dimensional Arrays

AIM: Write a program which stores information about n players in a two dimensional array. The array should contain number of rows equal to number of players. Each row will have number of columns equal to number of matches played by that player which may vary from player to player. The program should display player number (index +1), runs scored in all matches and its batting average as output. (It is expected to assign columns to each row dynamically after getting value from user.

Expected OUTCOME of Experiment:

CO2: Solve problems using Java basic constructs (like if else statement, control structures, and data types, array, string, vectors, packages, collection class).

Books/ Journals/ Websites referred:

1. Ralph Bravaco , Shai Simoson , "Java Programing From the Group Up" Tata McGraw-Hill.

2.Grady Booch, Object Oriented Analysis and Design .

Pre Lab/ Prior Concepts:

Arrays:

Multi-Dimensional Array:

10 12 43 11 22

20 45 56 1 33

30 67 32 14 44

40 12 87 14 55

50 86 66 13 66

60 53 44 12 11

A multi-dimensional array is one that can hold all the values above. You set them up like this:

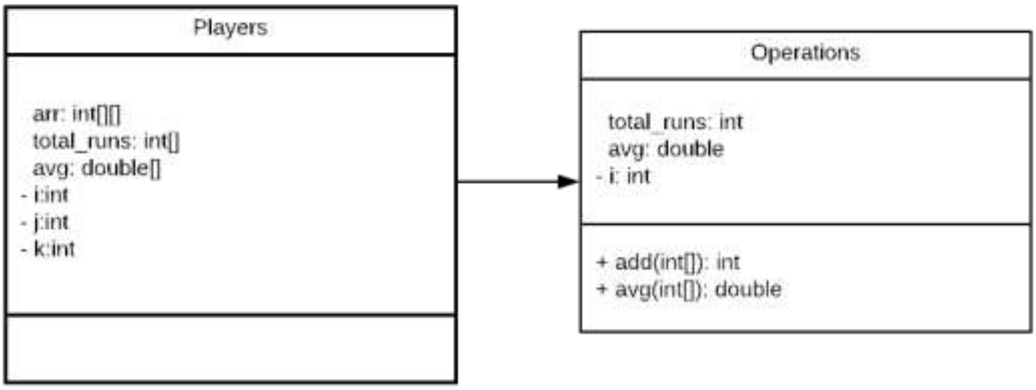
```
int[ ][ ] numbers = new int[6][5];
```

The first set of square brackets is for the rows and the second set of square brackets is for the columns. In the above line of code, we're telling Java to set up an array with 6 rows and 5 columns.

```
aryNumbers[0][0] = 10;
aryNumbers[0][1] = 12;
aryNumbers[0][2] = 43;
aryNumbers[0][3] = 11;
aryNumbers[0][4] = 22;
```

So the first row is row 0. The columns then go from 0 to 4, which is 5 items.

Class Diagram:



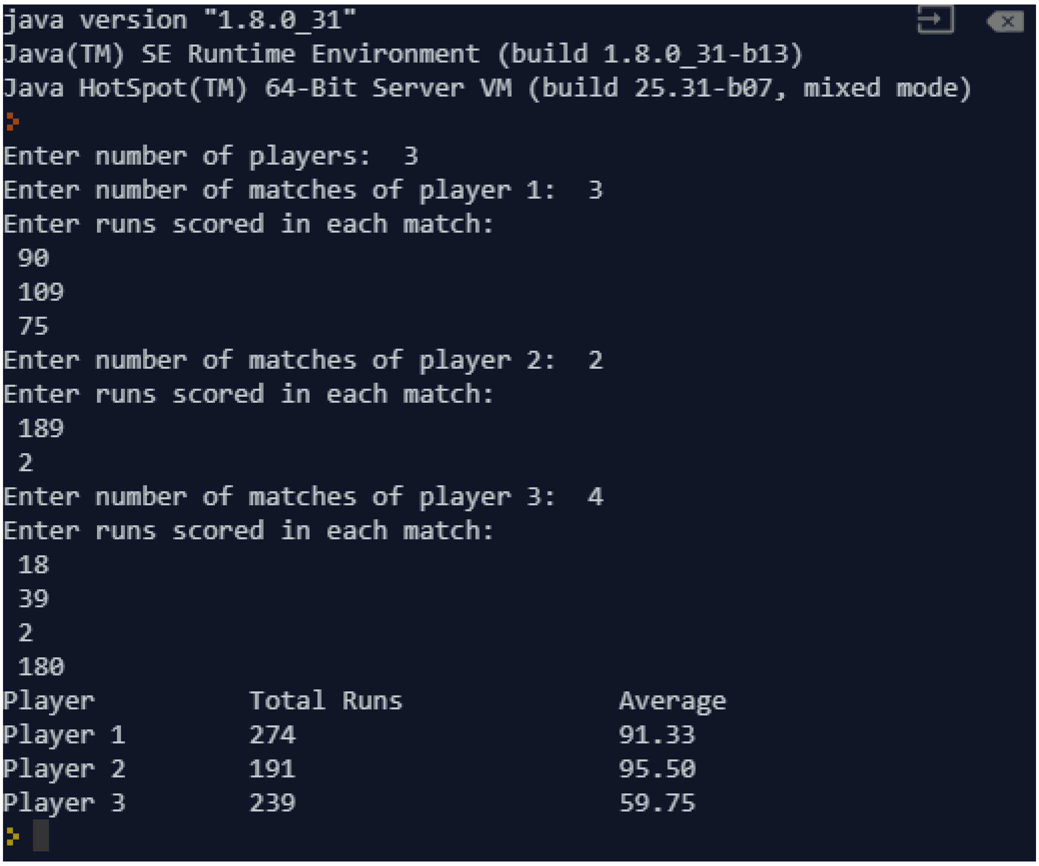
Implementation details:

```
import java.util.*;
import java.text.DecimalFormat;
class Operations{
  public int add(int arr[]){
    int total_runs=0;
    for(int i=0;i<arr.length;i++){
        total_runs+=arr[i];
    }
    return total_runs;
  }
   public double avg(int arr[]){
    double avg=0.0;
    int total_runs=0;
    for(int i=0;i<arr.length;i++){
        total_runs+=arr[i];
      }
      return (float)total_runs/(float)arr.length;
  }
}
public class Players {
  public static void main(String[] args) {
    System.out.print("Enter number of players: ");
    Scanner sc=new Scanner(System.in);
    DecimalFormat df=new DecimalFormat("#.00");
    int n=sc.nextInt();
    Operations op=new Operations();
    int[][] arr= new int[n][];
    int[] total_runs=new int[n];
    double[] avg=new double[n];
    for(int i=0;i<n;i++){
    System.out.print("Enter number of matches of player "+(i+1)+": ");
    int j=sc.nextInt();
    arr[i]=new int[j];
    System.out.println("Enter runs scored in each match: ");
    for(int k=0;k<arr[i].length;k++){
        arr[i][k]=sc.nextInt();
       }
    }
  System.out.println("Player\t\tTotal Runs\t\tAverage");
    for(int i=0;i<n;i++){
  total_runs[i]=op.add(arr[i]);
  avg[i]=op.avg(arr[i]);
```

```
        System.out.println("Player 
"+(i+1)+"\t"+total_runs[i]+"\t\t\t"+df.format(avg[i]));
    }
  }
}
```

For cross verification, the code can be found at:
https://repl.it/@ARGHYADEEPDAS/ArghyadeepDasOOPMExperiments

Output Screen:



```
java version "1.8.0_31"
Java(TM) SE Runtime Environment (build 1.8.0_31-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.31-b07, mixed mode)

Enter number of players:  3
Enter number of matches of player 1:  3
Enter runs scored in each match:
 90
 109
 75
Enter number of matches of player 2:  2
Enter runs scored in each match:
 189
 2
Enter number of matches of player 3:  4
Enter runs scored in each match:
 18
 39
 2
 180
Player          Total Runs              Average
Player 1        274                     91.33
Player 2        191                     95.50
Player 3        239                     59.75
```

Conclusion: The concept of jagged array was implemented, arrays were passed as parameters and arrays were returned. The program ran successfully and the aim of the experiment was met.

Post Lab Descriptive Questions (Add questions from examination point view)

Q.1 Which of the following statements are valid array declaration?

(A) int number();
(B) float average[];
(C) double[] marks;
(D) counter int[];

(i) (D)

(ii) (A) & (C)

(iii) (A)

(iv) (B) & (C)

Option (iv) contains the choices which are valid declarations.

Q.2 Consider the following code:

int number[] = new int[5];
After execution of this statement, which of the following are true?
(A) number[0] is undefined
(B) number[5] is undefined
(C) number[4] is null
(D) number[2] is 0
(E) number.length() is 5

(i) (C) & (E)

(ii) (A) & (E)

(iii) (E)

(iv) (B), (D) & (E)

Option (iv) is correct. An array which is declared is initialized as 0 for int type. 5th element is number[4] but not number[5]. Length of array is 5.

Q.3 What will be the content of array variable table after executing the following code?

```
for(int i = 0; i < 3; i + +)
for(int j = 0; j < 3; j + +)
if(j = i) table[i][j] = 1;
else table[i][j] = 0;
```

A).
0 0 0
0 0 0
0 0 0

B).
0 0 1
0 1 0
1 0 0

C).
1 0 0
1 1 0
1 1 1

D)
1 0 0
0 1 0
0 0 1

Option D gives the correct array pattern since the given logic allocates non-zero value to diagonal elements only (diagonal elements are those which have row number=column number in a square matrix).

Date: 28/07/2018 **Signature of faculty in-charge**