Day-6 JA-111 Batch

Topics

- **Array in java, 2-D array**
- **Array as parameter & return type**
- **Command line argument**

Array in Java

Say Masai School has started a new course and in the first batch they have 5 scholars only (As this batch is just to check for proficiency of trainer and efficacy of course content). The first week evaluation is done and we need to write program to find average marks of each scholar. Look at the following code-

```
package com.masai.sprint2;
import java.util.Scanner;

public class SumAverageExample {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int a, b, c, d, e;
    System.out.print("Enter marks of five scholars ");
    a = sc.nextInt();
    b = sc.nextInt();
    c = sc.nextInt();
    d = sc.nextInt();
    e = sc.nextInt();

    int sum = a + b + c + d + e;
    float average = (float)sum/5;

    System.out.println("Sum = " + sum);
    System.out.println("Average= " + average);

    sc.close();
  }
}
```

Output

```
Enter marks of five scholars 10 20 30 40 50
Sum= 150
Average= 30
```

Say the course is successful and content is also as per the industry demand so we have 100 students in the next batch, Now have want to extend this program for 1000 variables doing so using the same approach is very difficult.

Solution: *Use array*

- An array is a collection of similar typed variable that are referred to by a common name.
- Each item in array is called element and each element is accessed by its numeric index.
- Numeric index begins with zero and ends to one less than length of array.
- In our discussion we are will discuss about 1-D, 2-D, Asymmetric array & Array of objects

One Dimensional Array

Creating an array is two steps process:

Step-1: Declaration: General form of declaring an array is

```
type var-name[];
      or
type[] var-name;
```

Here type defines the data type of array. i.e. type of value that every element of array is going to hold. The size of array is not part of declaration and var-name is any valid name of array.

e.g. `int month[];`

Above discussion so far is just declaration of an array still we have to allocate memory to it.

Step-2: Constructing or allocating memory: Memory is allocated to an array using new operator. General form of constructing an array is

```
var-name = new type[size];
```

Here var-name is name of the array. new is an operator that provide memory to variable. Type is simply data type of array and size indicate no of elements that it can hold.

e.g. `month = new int[10];`

step-1 and **step-2** can be combined as

```
type[] var-name = new type[size];
```

e.g. `int[] month = new int[12];`

Initializing an Array

General form of initializing an array is

```
type[] var-name = {initializing values};
```

When initialization is used, no need to use new operator because memory is allocated implicitly for total number of elements specified in initialization list.

e.g. `int[] month = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};`

Miscellaneous Points

1. Java performs bound checking that is, if we try to access any element that is outside range of array then run time error will be generated that will halts program immediately (How to avoid halting of program will discuss in exception handling)

   ```
   System.out.println("month 13 have " + month[12] + " days");
   Output:
   Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 12
   ```

2. java implements array as an object. To find size of an array, java provides inbuilt property which is, length that returns total number of elements in array.

   ```
   System.out.println("Array month have "  + month.length + " elements");
   Output:
   Array month have 12 elements
   ```

The for-each loop

Apart from the conventional while, for & do-while loop we have one more type of loop that is for-each loop which is very much useful for iterating over the sequences and array.

General form of for loop is

```
for(type var:arr){
  //body of loop
}
```

Using foreach loop we don't required to create and maintain a variable that deals with index of array.

An Example:

```
package com.masai.sprint2;
import java.util.Scanner;

class SumAverageExample {
  public static void main(String args[]){
    Scanner sc = new Scanner(System.in);
    int marks[] = new int[5];

    System.out.print("Enter marks of five scholars ");
    for(int i = 0; i < marks.length; i++)
      marks[i] = sc.nextInt();

    int sum = 0;
    for(int currentMarks : marks)
      sum = sum + currentMarks;

    float average = (float)sum/5;
    System.out.println("Sum = " + sum);
    System.out.println("Average= " + average);

    sc.close();
  }
}
```

Output

```
Enter marks of five scholars 10 20 30 40 50
Sum= 150
Average= 30
```

In the code segment given above

```
for(int currentMarks : marks)
  sum = sum + currentMarks;
```

is equivalent to

```
for(int i = 0; i < arr.length; i++){
  int currentMarks = arr[i];
  sum = sum + currentMarks;
}
```

An Analogy

Variable

1-D array

2-D array

2-D Array

This is also called matrix or business table.

Similar to one dimensional, creating a 2-D array is also two-step process:
Step-1 Declaration: General form of declaring a 2-D array is

```
type var-name[][];
      or
type[][] var-name;
```

Here type defines the data type of array. i.e. type of value that every element of array hold. Square bracket indicates that variable holds an array. size of array is not part of declaration. and var-name is any valid name of array.
e.g. `int matrix[][];`
Above discussion so far is just declaration of an array still we have to allocate memory to it.

Step-2 Constructing or allocating memory: Memory is allocated to an array using new operator. General form of constructing an array is

```
var-name = new type[row_size][col_size];
```

Here var-name is name of the array. new is an operator that provide memory to variable. type is simply data type of array and row_size indicate no of 1-D array and col_size indicates no of elements in a 1-D array.
e.g. `matrix = new int[3][4];`

step 1 and step 2 can be combined as

```
type[] var-name = new type[row_size][col_size];
```

e.g. `int[][] matrix = new int[3][4];`

- To initialize a 2-D array, Simply put each 1-D array's initializing values in the their own curly braces separated by comma, and put entire set of curly braces in one pair of curly braces.
- If length property is used with the 2-D array then it will give total number of 1-D array and if length property is used with the 1-D array then it will give total number of elements in the 1-D array.

An Example:

```
package com.masai.sprint2;
class TwoDArrayIni{
  public static void main(String args[]){
    int matrix[][] = {
```

```
      {31, 28, 31, 30},
      {31, 30, 31, 31},
      {30, 31, 30, 31}
    };

    for(int brr[]: matrix)
      for(int a: brr)
        System.out.print(a + " ");
      System.out.println();

    System.out.println(matrix.length + " " + matrix[0].length);
  }
}
```

Output of the program above is

```
31 28 31 30
31 30 31 31
30 31 30 31
3 4
```

Asymmetric Array

Java provides facility of creating array with uneven rows, means we can create an array that have different number of elements in each row. To do that, just specify first dimension of array and no. of element for each row can be specified separately later on.

An Example:

```
package com.masai.sprint2;
class UnevenArray{
  public static void main(String args[]){
    int matrix[][] = new int[3][];
    matrix[0] = new int[2];
    matrix[1] = new int[3];
    matrix[2] = new int[4];

    for(int row = 0; row < matrix.length; row++)
      for(int col = 0; col < matrix[row].length; col++)
        matrix[row][col] =   row + col;

    for(int row = 0; row < matrix.length; row++)
      for(int col = 0; col < matrix[row].length; col++)
        System.out.print(matrix[row][col] + " ");
      System.out.println("");
  }
}
```

Output of the program given above is

```
0 1
1 2 3
2 3 4 5
```

In the program given above matrix is a 2-D array. its first 1-D array have 2 elements, second 1-D array have 3 and last 1-D array have 4 elements. We have just specified number of rows at the time of

declaration that is merged with memory allocation. but number of element for each row are defined in subsequent lines.

using length property of array with 2-D array returns number of 1-D arrays that it have and using it with 1-D array returns number of elements that it have. hence matrix.length returns 3 while matrix[0].length returns 2.

Array of object

As we already know that *a variable for a data type is what an object for the class is* so like we have created array of primitive type, we can create array of objects also. Creating an array of object is a two-step process.

Step-1: Create an array of reference, general form of creating an array is

```
class-name obj-name[] = new class-name[SIZE];
```

In that step we are just creating array of reference so far we have not allocated memory for objects.

Step-2: Allocate memory for each object separately

```
obj-name[0] = new class-name();
obj-name[1] = new class-name();
.
.
obj-name[SIZE - 1] = new class-name();
```

The array in java has length property that contains the total elements in the array.

An Example

```
package com.masai.sprint2;

public class Rectangle {
  double length;
  double breadth;

  Rectangle(double length, double breadth){
    System.out.println("inside constructor");
    this.length = length;
    this.breadth = breadth;
  }

  double getArea(){
    return (length * breadth);
  }
}

package com.masai.sprint2;
import java.util.Scanner;

public class RectanlgeArrayDemo {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    Rectangle rectArr[]= new Rectangle[3];
```

```
    for(int i = 0; i < rectArr.length; i++) {
      System.out.print("Enter length and breadth ");
      double l = sc.nextDouble();
      double b = sc.nextDouble();
      rectArr[i] = new Rectangle(l, b);
    }

    for(Rectangle rectTemp: rectArr) {
      System.out.println("The area of rectangle is " + rectTemp.getArea());
    }
    sc.close();
  }
}
```

Output

```
Enter length and breadth 10 20
inside constructor
Enter length and breadth 5 15
inside constructor
Enter length and breadth 4 14
inside constructor
The area of rectangle is 200.0
The area of rectangle is 75.0
The area of rectangle is 56.0
```

An array can be pass be passed to a method and it can be returned from a method.

```
package com.masai.sprint2;

public class ArrayDemoExample {
  static int[] getEvenOddCount(int arr[]) {
    //create an array with two elements
    //The value at index 0 is for counting of even elements
    //The value at index 1 is for counting of odd elements
    int brr[] = {0, 0};

    for(int a: arr)
      if(a % 2 == 0)
        brr[0]++;
      else
        brr[1]++;

    //return array
    return brr;
  }

  public static void main(String[] args) {
    //Create an array
    int arr[] = {1, 5, 9, 4, 12, 14, 3};
    int brr[] = getEvenOddCount(arr);
    System.out.println("Total even elements " + brr[0]);
    System.out.println("Total odd elements " + brr[1]);
  }
}
```

Output

```
Total even elements 3
Total odd elements 4
```

Command line argument

- Command line arguments are used to pass information to a java program.
- Command line arguments are stored in String array that is part of main method. To access them simply refer to index of String array.

An Example:

```
package com.masai.sprint2;

class CommandLineDemo {
  public static void main(String[] args) {
    for(int i = 0; i < args.length; i++){
      System.out.println((i+1)+ " command line argument is-" + args[i]);
    }
  }
}
```

Do right click → Run As → Run Configuration → open arguments tab → type “all is well” (without quotations) on the in the program arguments Text box → Run

Output of code given above is:

1 command line argument is-all
2 command line argument is-is
3 command line argument is-well

To pass multiword string as a single argument simply enclose them in single/double quotes
Do right click → Run As → Run Configuration → open arguments tab → type “all is well” (with quotations) on the in the program arguments Text box → Run

Output of code in that case is

1 command line argument is-all is well

You Activity

Write a program to input an array of N X N size (i.e. it is square matrix) then find the difference of sum of diagonal elements.

Sample Input
Enter Size of matrix: 4
Enter elements of matrix:
1 2 3 4
5 6 3 1
7 1 2 3
5 8 4 6

Sample Output
The difference of sum of diagonal is 2

Explanation
The addition of elements of first diagonal is 1 + 6 + 2 + 6 = 15
The addition of elements of second diagonal is 5 + 1 + 3 + 4 = 13
The difference of 15 – 13 is 2