**19CSE204**

**OBJECT ORIENTED PARADIGM**

**S. Abhishek**

**AM. EN. U4CSE19147**



**import java.util.Scanner**

To import the Java Scanner class Library from the Java Library.

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

Create the Instance of the Scanner class and invoking it in the object and using it to extract all other input methods to get the input.

**int size=in.nextInt();**

Using the object to get the “Integer” type inputs.

**int[] myList=new int[size];**

Creating the Array named “myList” by using the keyword “new” and dynamically allotting memory.

It assigns the reference of the newly created array to the variable “myList”.

**for (int i=0; i<size; i++)**

**{**

**myList[i] = in.nextInt();**

**}**

The array elements are accessed through the index. Array indices are 0-based; that is, they start from 0 to length of the Array-1.

In this the user is prompted to enter the input value for the allotted size and store it in the “myList” Array.

**For (int i=0; i < myList.length; i++) {**

**System.out.print(myList[i]);}**

This prints the elements after getting input from the User.

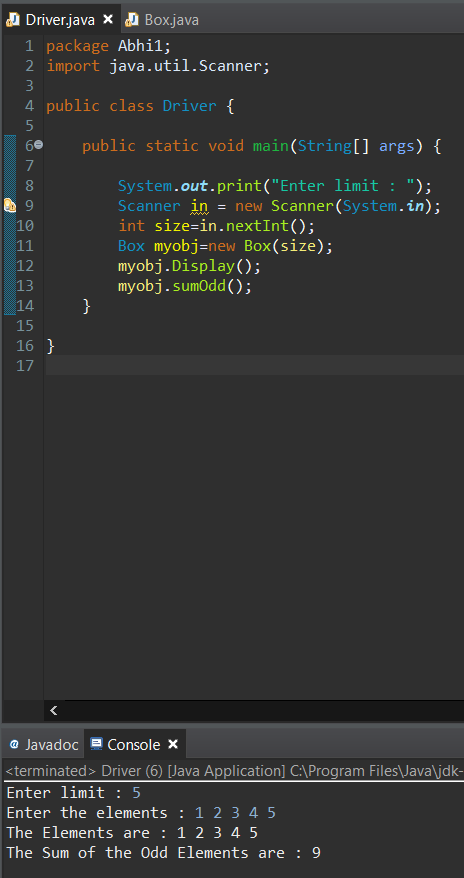
**For (int num:myList) {**

**sum = sum+num;}**

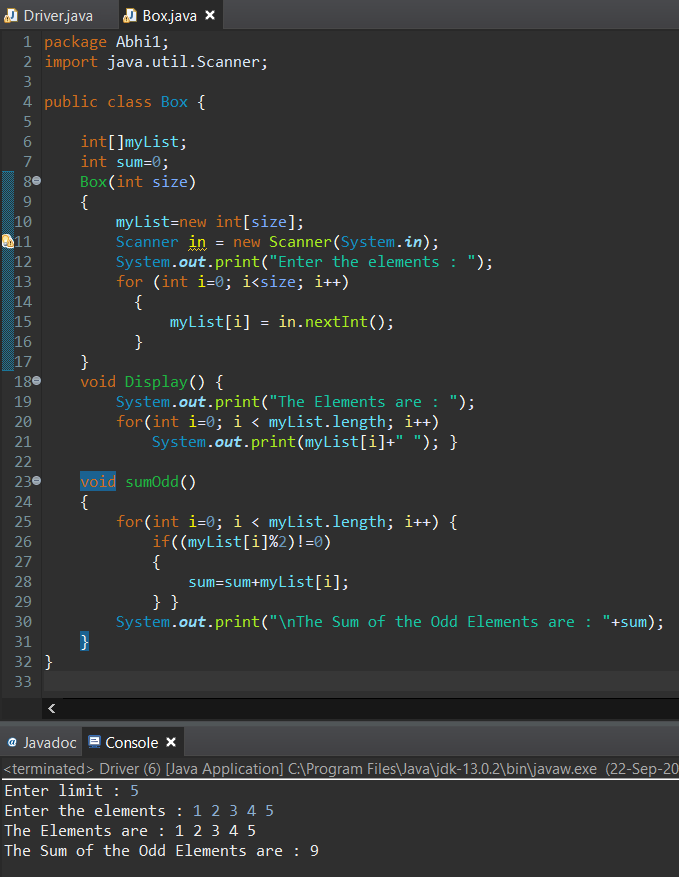
}

This sums up all the elements that are stored in the Array.

**DRIVER CLASS:**



**BOX CLASS:**



**BOX CLASS**

**int[]myList;**

Creating the Array “myList”.

**Box (int size)**

**{**

**myList=new int[size];**

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

Parameterized Constructor which is used to allot the memory to the Array.

**for (int i=0; i<size; i++)**

**{**

**myList[i] = in.nextInt();**

**}**

The array elements are accessed through the index. Array indices are 0-based; that is, they start from 0 to length of the Array-1.

In this the user is prompted to enter the input value for the allotted size and store it in the “myList” Array.

**}**

**void Display()**

**{**

**for(int i=0; i < myList.length; i++)**

**System.*out*.print(myList[i]+"** ");

This method is used to print the elements after storing it in the Array by getting input from the User.

**}**

**void sumOdd(){**

**for(int i=0; i < myList.length; i++) {**

**if((myList[i]%2)!=0)**

**{**

**sum=sum+myList[i];**

**}**

**}**

**}**

This sums up all the odd elements that are stored in the Array.

“If” condition is used to check whether its odd or not using the denominator.

**DRIVER CLASS**

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

Create the Instance of the Scanner class and invoking it in the object and using it to extract all other input methods to get the input.

**int size=in.nextInt();**

Using the object to get the “Integer” type inputs.

**Box myobj=new Box(size);**

Creating the Object named “myobj” and calling the Parameterized constructor named “Box” to invoke the instances of the Class “Box” into the object “myobj”.

**myobj.Display();**

Calling the member function in the Class “Box()” using the object “myobj” and Dot operator.

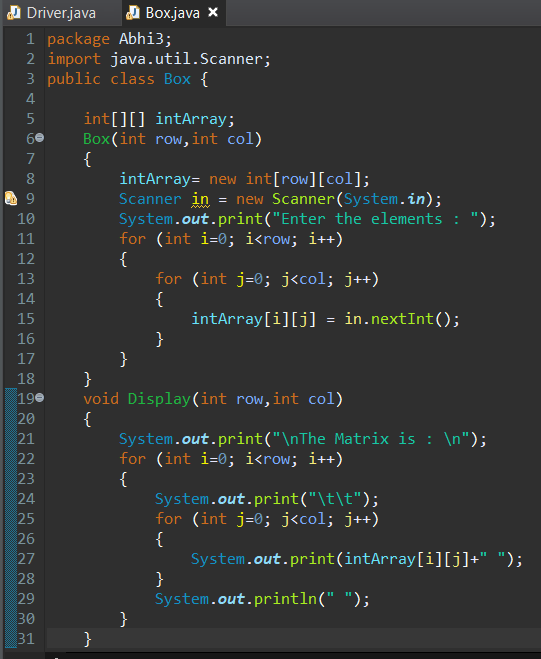
This method is used to Display the elements of the Array.

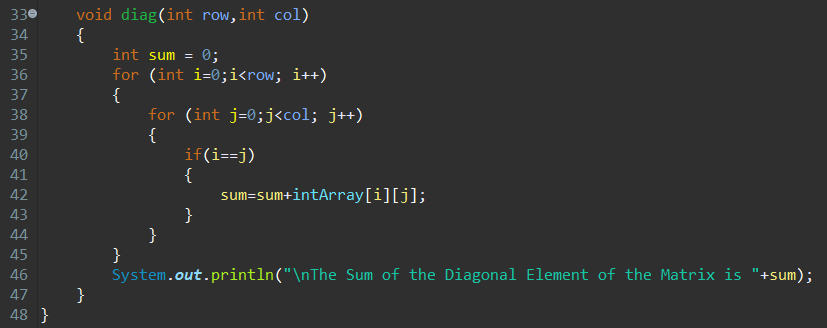
**myobj.sumOdd();**

Calling the member function in the Class “Box()” using the object “myobj” and Dot operator.

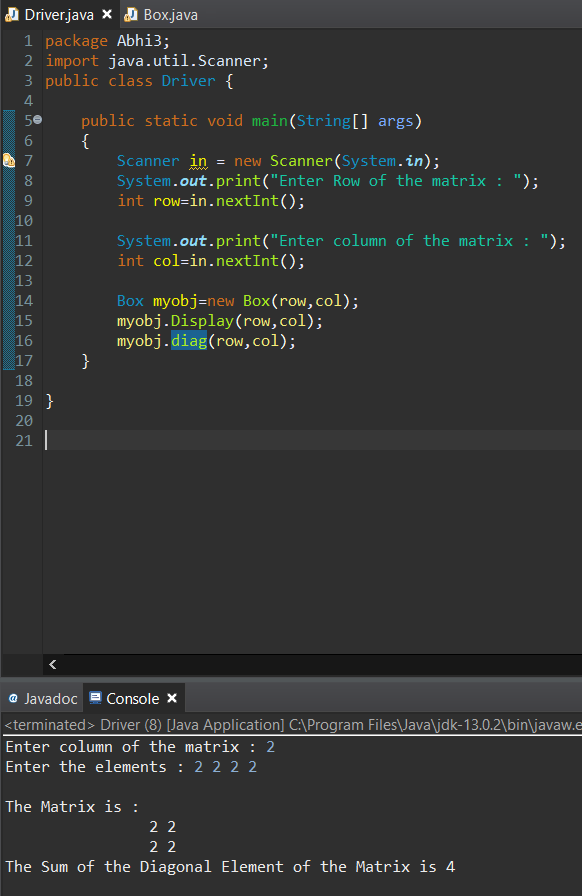
This method is used to calculate the sum of the Odd elements of the Array.

**BOX CLASS:**

****

****

**DRIVER CLASS:**

****

**int[][] intArray;**

This is used to create the Array of name “intArray”.

**Box(int row,int col)**

Box constructor to create the Matrix of Size N Rows and M Columns.

**{**

**intArray= new int[row][col];**

Declaring size for the 2D Array.

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

Create the Instance of the Scanner class and invoking it in the object and using it to extract all other input methods to get the input.

**for (int i=0; i<row; i++)**

**{**

**for (int j=0; j<col; j++)**

**{**

**intArray[i][j] = in.nextInt();**

**}**

**}**

**}**

The array elements are accessed through the index. Array indices are 0-based; that is, they start from 0 to length of the Array-1.

In this the user is prompted to enter the input value for the allotted size and store it in the “intArray” Matrix or 2D Array.

**void Display(int row,int col)**

This Member function is used to display the elements stored in the 2D Matrix.

**{**

**System.out.print("\nThe Matrix is : \n");**

**for (int i=0; i<row; i++)**

**{**

**System.out.print("\t\t");**

**for (int j=0; j<col; j++)**

**{**

**System.out.print(intArray[i][j]+" ");**

This statement is used to print the elements of the 2D Array.

**}**

**System.out.println(" ");**

**}**

**}**

Here “i” is used to represent the i’th 1D Array and the “j” is used to represent the j’th element of the i’th 1D array.

**void diag(int row,int col)**

This Member function is used to find the sum of the Anti diagonal Elements.

**{**

**int sum = 0;**

**int j=col-1;**

**for (int i=0;i<row; i++) {**

**sum=sum+intArray[i][j];**

Here the antidiagonal element is accessed using the two counters. “I” is for Rows and “j” is for Columns.

**j--;**

“j” starts from the last column and in each iteration the value decreases by 1 and finally ends at the 0’th Position.

**}**

**System.*out*.println("\nThe Sum of the Diagonal Element of the Matrix is "+sum);**

This statement is used to print the sum of the Anti-Diagonal elements that is calculated.

**}**

**DRIVER CLASS**

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

Create the Instance of the Scanner class and invoking it in the object and using it to extract all other input methods to get the input.

Inputs the Rows of the Matrix.

**System.*out*.print("Enter Row of the matrix : ");**

**int row=in.nextInt();**

Inputs the Columns of the Matrix.

**System.*out*.print("Enter column of the matrix : ");**

**int col=in.nextInt();**

**Box myobj=new Box(row,col);**

Creating the Object named “myobj” and calling the Parameterized constructor named “Box” to invoke the instances of the Class “Box” into the object “myobj”.

**myobj.Display(row,col);**

Calling the member function in the Class “Box()” using the object “myobj” and Dot operator.

It prints the elements of the 2D Array.

**myobj.diag(row,col);**

This calls the “diag” member function which is used to find the sum of the Anti-diagonal Elements.