

Scopes in java

Local and Global

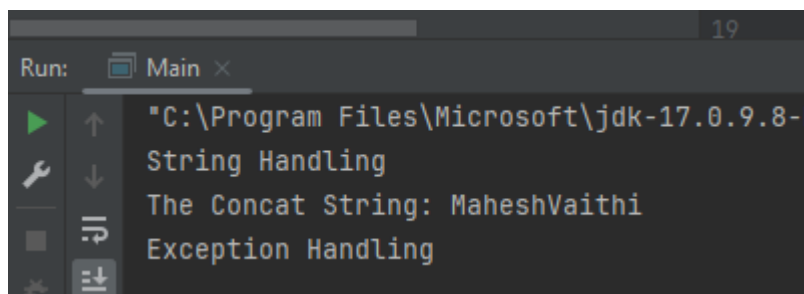
Access Modifiers in Java:

Public , Private and Protected

String Handling in Java:

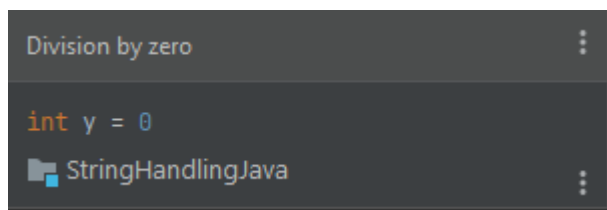
Refer an oracle string operations for inbuilt function

```
System.out.println("String Handling");  
String a = "Mahesh";  
String b = "Vaithi";  
  
String c=a+b; //concat string  
  
System.out.println("The Concat String: "+c);
```



Exception Handling

```
System.out.println("Exception Handling");  
  
int x = 10;  
int y = 0;  
  
int res = x/y;  
System.out.println("Result Of :"+res);
```



Divide by Zero exception

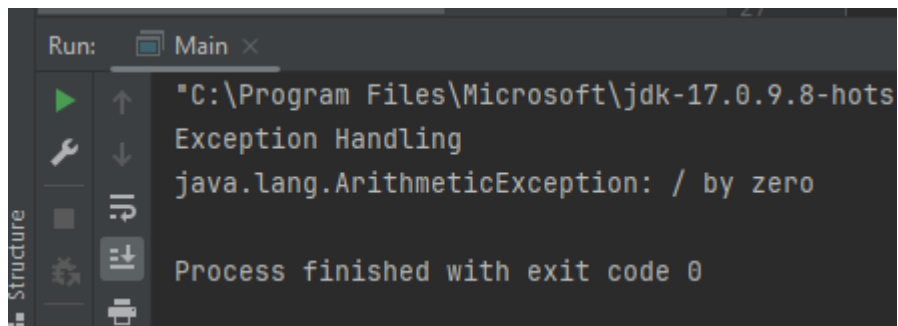
By Arithmetic exception:

```
System.out.println("Exception Handling");

int x = 10;
int y = 0;

try {

    int res = x / y;
    System.out.println("Result Of :"+res);
}
catch(ArithmeticException e) {
    System.out.println(e);
}
```



```
// Multiple Exception

int n1=200;
int n2=0;

try{

    int s = n1/n2;
    System.out.println("Result of: "+s);
}
catch (Exception e){

    System.out.println(e);
}
finally {
    System.out.println(n1);
}
```

```
Run: Main x
"C:\Program Files\Microsoft\jdk-17.0.9.8-hotspot\bin\java.exe"
Exception Handling
java.lang.ArithmeticException: / by zero
java.lang.ArithmeticException: / by zero
200
Process finished with exit code 0
```

Parent exception should be under the child exception:

Exception - child

Arithmetic Exception - parent

Enums in Java

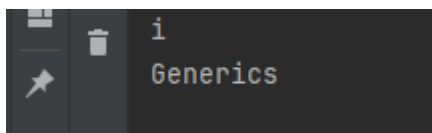
```
// Enum in java

Learning learning = Learning.GENERICS;

switch (learning) {
    case COREJAVA -> System.out.println("Core JAVA");
    case COLLECTIONS -> System.out.println("Collections");
    case GENERICS -> System.out.println("Generics");
    case JSP -> System.out.println("JSP");
    case MULTITHREADING -> System.out.println("MultiThreading");
}
```

```
package org.example;

public enum Learning {
    COREJAVA, COLLECTIONS, GENERICS, JSP, MULTITHREADING
}
```



All Array Ops:

```
package org.example;
import java.util.*;
public class Main {
    public static void main(String[] args) {
```

```

// arrays in java
// to print an array via user input

Scanner sc = new Scanner(System.in);
System.out.println("Enter the Size of Array: ");
int size = sc.nextInt();
int[] arr = new int[size];

// to get user input for an array using for loop
System.out.println("Enter the elements in the array: ");
for(int i=0;i<size;i++){
    arr[i]= sc.nextInt();
}
// to print the array

System.out.println("The elements in the Array are: ");

for(int j=0;j<arr.length;j++){
    System.out.println(arr[j]);
}

// to print for each loop to print individual array

System.out.println("Individual Array elements Using For each: ");

for(int s : arr){
    System.out.println(s);
}

Main m = new Main();
m.bsort(arr);

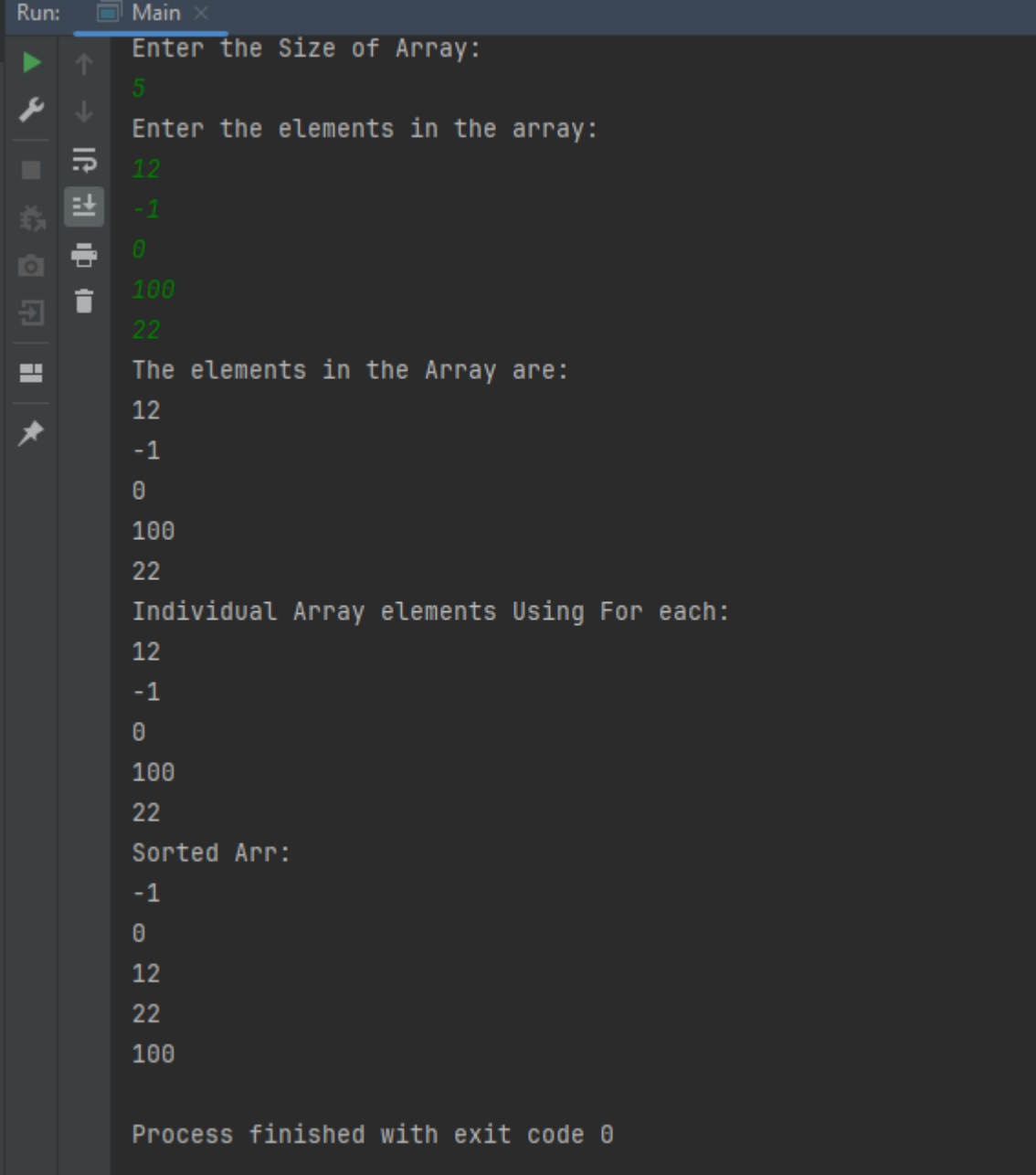
}

public void bsort(int[] arr){
    for(int i=0;i<arr.length;i++){
        for(int j=i+1;j< arr.length;j++){
            if(arr[i]>arr[j]){
                //swap the arr
                int temp = 0;
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
    System.out.println("Sorted Arr:");

    for(int a: arr){
        System.out.println(a);
    }
}

```

```
}  
}
```



The screenshot shows a Java IDE console window titled "Run: Main". The console displays the following text:

```
Enter the Size of Array:  
5  
Enter the elements in the array:  
12  
-1  
0  
100  
22  
The elements in the Array are:  
12  
-1  
0  
100  
22  
Individual Array elements Using For each:  
12  
-1  
0  
100  
22  
Sorted Arr:  
-1  
0  
12  
22  
100  
  
Process finished with exit code 0
```

The console window includes a toolbar on the left with icons for running, debugging, and other IDE functions.

Array Lists in Java

```
package org.example;  
import java.util.*;  
public class Main {  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);
```

```
// array list implementation

System.out.println("Array List Implementation: ");

System.out.println("Enter the size of list: ");

int size = sc.nextInt();

ArrayList<String> list = new java.util.ArrayList<java.lang.String>();

System.out.println("Enter the Strings into List:");
for(int i=0;i<size;i++){
    list.add(sc.next());
}
// to view the elements in the list
System.out.println("Strings in the List are: "+list);

//for each to view the list
System.out.println("Iterate through list: ");
for(String s : list){
    System.out.println(s);
}

}

}
```

```
Run: Main x
"C:\Program Files\Microsoft\jdk-17.0.9.8-hotspot\bin\java.exe" "-ja
Array List Implementation:
Enter the size of list:
5
Enter the Strings into List:
mahesh
pradeep
jai
vijay
afsal
Strings in the List are: [mahesh, pradeep, jai, vijay, afsal]
Iterate through list:
mahesh
pradeep
jai
vijay
afsal

Process finished with exit code 0
|
```

Stack Linked List are refer code doc :

Some of the Stack and Linked list methods are refer docs for inbuilt method

Hash Collections and its types refer code file:

Queue in java

```
package org.example;
import java.util.*;
import java.util.concurrent.LinkedBlockingDeque;
import java.util.concurrent.LinkedBlockingQueue;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Queue q = new LinkedBlockingQueue();

        System.out.println("Enter the size: ");
        int size = sc.nextInt();
```

```

        //enqueue operation
        System.out.println("Add the elements in the Queue");
        for(int i=0;i<size;i++){
            q.offer(sc.nextInt());
        }

        //display queue
        System.out.println("Queue Elements: "+q);

        // dequeue Operation:

        System.out.println("Element Dequeued: "+q.poll());

    }
}

```

```

Run: Main x
"C:\Program Files\Microsoft\jdk-17.0.9
Enter the size:
5
Add the elements in the Queue
12
34
56
8
9
Queue Elements: [12, 34, 56, 8, 9]
Element Dequeued: 12

Process finished with exit code 0

```

Maps in Java:

TreeMap and HashMap

Refer code doc:

Multithreading in Java: