DAY-43

22 August 2023 15:35

ARRAY INDEX OUT OF BOUNDS EXCEPTION HANDLING

```
package exc;

public class ArrayIndex {

        public static void main(String[] args)
        {
                int[] a1 = new int[3];

                try
                {
                        a1[0]= 1;
                        System.out.println(a1[0]);

                        a1[1]= 2;
                        System.out.println(a1[1]);

                        a1[2]= 3;
                        System.out.println(a1[2]);

                        a1[3]= 4;
                        System.out.println(a1[3]);
                        System.out.println("end of the program");
                }
                catch(ArrayIndexOutOfBoundsException a)
                {
                        System.out.println("entering catch block");
                        System.out.println(a.getMessage());
                }


        }

}
```

CAN ONE TRY BLOCK HAVE MULTIPLE CATCH BLOCK ?

```
Try
{
    s.o.p(10/0);
}

Catch(AIB e)
{
        ------
}

Catch(Arithmetic exception)
{
        ------
}

Catch(String exception)
{
        ------
}
```

- WE CAN HAVE MULTIPLE CATCH() BLOCK FOR SINGLE TRY BLOCK
- WE CAN NOT HAVE MULTIPLE EXCEPTION IN A SINGLE TRY BLOCK
- WE CAN WRITE MULTIPLE TRY BLOCKS INTO A PROGRAM
- WE NEED TO ALWAYS IN TRY-CATCH(), TRY-CATCH() FORMAT ONLY.
- WE CAN NOT WRITE IN TRY-TRY , CATCH() - CATCH() FORMAT
- IF THERE ARE MULTIPLE EXCEPTIONS IN A SINGLE TRY BLOCK THEN WE GET CTE

SPECIALIZED CATCH() BLOCK

```
TRY
{
        S.O.P(10/0)
}

CATCH(ARITHMETICEXCEPTION e)
{
        //SPECIALIZED CATCH BLOCK

}
```

ArithmaticException e = New ArithmaticException();

This is done implicitly by jvm

GENERALIZED CATCH() BLOCK

```
TRY
{
        S.O.P(10/0)
}

CATCH(EXCEPTION e)
{
        //GENERALIZED CATCH BLOCK

}
```

Exception e = New ArithmaticException();

Here implicit upcasting is being done by jvm

Exception e = New ArrayIndexOutOfBoundsException();

Exception e = New NullPointerException();

HOW THE EXECUTION OF SPECIALIZED AND GENERALIZED CATCH BLOCK TAKE PLACE ?

```
TRY
{
        S.O.P(10/0) // or any other exception
}

CATCH(ArithmaticException e)
{
        //SPECIALIZED CATCH BLOCK
```

```
CATCH(ArithmaticException e)
{
        //SPECIALIZED CATCH BLOCK

}


CATCH(Exception e)
{
        //GENERALIZED CATCH BLOCK

}
```

✓

```
TRY
{
        S.O.P(10/0) // or any other exception
}


CATCH(Exception e)
{
        //GENERALIZED CATCH BLOCK

}

CATCH(ArithmaticException e)
{
        //SPECIALIZED CATCH BLOCK

}
```

✗

THIS WILL GIVE **CTE** AS THE JCM WILL NOT BE ABLE TO REACH THE SCB (SPECIALIZED CATCH BLOCK)

- CAN ONE TRY BLOXK HAVE MULTIPLE CATCH() BLOCKS ?

  YES

- WHICH CATCH() BLOCK WILL BE EXECUTED ?

  DEPENDS OF TYPE OF EXECPTION AND TYPE OF EXCEPTION OBEJCT REFERNCE IN CATCH.

- MULTIPLE CATCH() BLOCKS WILL BE EXECUTED ? (TRUE/FALSE)

  FALSE

```
package exc;

public class Test1 {

        public static void main(String[] args)
        {
                try
                {
                        System.out.println(1/0);
        }
        catch(ArithmeticException c1)
        {
            System.out.println("inside specilized catch block");
            System.out.println(c1.getMessage());
        }
        catch(Exception c)
        {
            System.out.println("inside generalized catch block");
            System.out.println(c.getMessage());
        }

    }

}
```

```
package exc;

public class Test1 {

    public static void main(String[] args)
    {
        String a = "java";
        try
        {
            Test1 a1 = null;
            System.out.println(a1);
        }
        catch(Exception c)
        {
            System.out.println("inside generalized catch block");
            System.out.println(c.getMessage());
        }

        catch(NullPointerException c1)
        {
            System.out.println("inside specilized catch block");
            System.out.println(c1.getMessage());
        }

    }

}
```

CAN WE HAVE A SPECIALIZED AND GENERALIZED BLOCK FOR A SINGLE TRY ?

- YES , BUT THE ORDER MUST BE FOLLWED i.e, FIRST SPECILIAZED AND THEN GEENRALIZED.
- IF WE WRITE SPECIALIZED BLICK FIRST , THEN SPECIALIZED BLOCK CAN HANDLE THE EXCEPTION OR ELSE GENERALIZED BLOCK MAY GET THE CHANCE TO HANDLE THE EXCEPTION.
- BUT, IF WRITE GENERALIZED BLOCK FIRST , THEN IT CAN HANDLE ALL TYPE OF EXCEPTION, SO THE SPECIALIZED BLOCK WILL NEVER GET A CHANCE AND IT WILL BECOME UNREACHABLE CODE, SO IT LEADS TO COMPILE TIME ERROR.

```
TRY
{

}
CATCH()
{

}
```
✓

```
TRY
{

}
CATCH()
{

}
CATCH()
{

}
```
✓

```
TRY
{

}
CATCH()
{

}

TRY
{

}
CATCH()
{

}
```
✓

```
CATCH()
{

}
```
✗

```
TRY
{

}
```
✗

```
TRY
{

}
S.O.P()
CATCH()
{

}
```
✗

```
TRY
{

}

TRY
{

}
CATCH()
{

}
CATCH()
{

}
```
✗

```
TRY
{

}
CATCH()
{

}
FINALLY
{

}
```
✓

THROW KEYWORD

```
TRY
{
        S.O.P(1/0)
}

CATCH(AE e)
{

}
```

```
TRY
{
        Throw new ArithmeticException();
}

CATCH(AE e)
{

}
```

- THROW KEYWORD IS USED TO THROW EXCEPTION OBJECT EXPLICITLY
- THROW CAN BE USED TO THROW BOTH CHECKED AND UNCHECKED EXCEPTION OBEJCT

    - Throw new NullPointerException();
    - Throw new IOException();
- CAN BE USED ONLY IN METHOD DEFINTION
- THROW CAN BE USED TO THROW ONLY THOSE EXCEPTION OBJECTS WHICH HAS THE PROPERTY OF THROWABLE TYPE.

```
TRY
{
        System.out.println("hello");
        System.out.println(10/0); // may or may not
        System.out.println("bye");

}

CATCH(AE e)
{

}
```

- IN RUNTIME DECISSION WILL BE TAKEN AS IN EXCEPTION IS OCCURING OR NOT

```
TRY
{
        System.out.println("hello");
        Throw new AE(); //100% IT WILL THROW
        System.out.println("bye");

}

CATCH(AE e)
{

}
```