Exception handling

1. exception handling can be identify the only runtime. Not a compile time

2. its is also called runtime errors

for example we access an array using index that is out of bounds

```
package com.thbs.mainpack;

public class Main {

    public static void main(String[] args) {

        int result = 0;
        //ArrayIndexOutOfBoundsException

        int n1 = Integer.parseInt(args[0]); //commond line arguments,.NumberFormatException,ArithematicException
        int n2 = Integer.parseInt(args[1]);
        result = n1/n2; //--- arthematic operation runtime error
        System.out.println(result);
        System.out.println("bye!!!!");


    }
}
```

output

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 1
        at com.thbs.mainpack.Main.main(Main.java:11)

Process finished with exit code 1
```

```
package com.thbs.mainpack;

public class Main {

    public static void main(String[] args) {

        int result = 0;
        //ArrayIndexOutOfBoundsException
        try { // Anticipate block


            int n1 = Integer.parseInt(args[0]); //commond line arguments,.NumberFormatException,ArithematicException
            int n2 = Integer.parseInt(args[1]);
            result = n1 / n2; //--- arthematic operation runtime error
```

```
                    // ArithematicException created and thrown
        }catch (ArithmeticException e)
        {

            System.out.println("Denominator is zero "+e.getMessage());
        }
        catch(NumberFormatException e)
        {
            System.out.println("Invalid inputs...."+e.getMessage());
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("Denominator is zero    "+e.getMessage());
        }
        finally { //whether or not exception encountered finally block will be executed
            System.out.println(result);
        }

        System.out.println("hello good morning!!!!");

    }
}
```

```
Invalid inputs....For input string: "good"
0
hello good morning!!!!

Process finished with exit code 0
```

---

modify the run congifuration

```
5
hello good morning!!!!

Process finished with exit code 0
```

---

main.java

```
package com.thbs.mainpack;

import com.thbs.exceptionpack.ReadInput;

import java.io.IOException;

public class Main {
```

```
    public static void main(String[] args) {
        try {
            ReadInput.readInput2();
        } catch (ArithmeticException e) {
            System.out.println("Denominator is zeo    " + e.getMessage());
        } catch (NumberFormatException e) {
            System.out.println("Invalid inputs...    " + e.getMessage());
        } catch (IOException e) {
            e.printStackTrace();
        }

    }
}
```

ReadInput.java

```
package com.thbs.exceptionpack;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
public class ReadInput {

    // try with resources ------> try with auto close option
    public static void readInput1()
    {
        try(InputStreamReader ir = new InputStreamReader(System.in);
            BufferedReader  br = new BufferedReader(ir);) {
            System.out.println("Read two input values : ");
            String str1 = br.readLine();
            String str2 = br.readLine();
            int result = Integer.parseInt(str1) / Integer.parseInt(str2);
            System.out.println(result);

        }
        catch (ArithmeticException e) {
            System.out.println("Denominator is zeo    " + e.getMessage());
        } catch (NumberFormatException e) {
            System.out.println("Invalid inputs...    " + e.getMessage());
        }
        catch (IOException e) {
            e.printStackTrace();
        }
    }

    public static void readInput2() throws IOException ,ArithmeticException,NumberFormatException{
        InputStreamReader ir = new InputStreamReader(System.in);
        BufferedReader  br = new BufferedReader(ir);
        System.out.println("Read two input values : ");
        String str1 = br.readLine();
```

```
        String str2 = br.readLine();
        int result = Integer.parseInt(str1) / Integer.parseInt(str2);
        System.out.println(result);
    }


    // for a checked exception
    public static void readInput() {
        InputStreamReader ir = null;
        BufferedReader br = null;
        String str1, str2;
        // InputStreamReader is pipelined with BufferedReader
        try {
            ir = new InputStreamReader(System.in);  // will read one char at a time
            br = new BufferedReader(ir);  // br will be
            System.out.println("Read two input values : ");
            str1 = br.readLine();
            str2 = br.readLine();
            int result = Integer.parseInt(str1) / Integer.parseInt(str2);
            System.out.println(result);
        } catch (ArithmeticException e) {
            System.out.println("Denominator is zeo    " + e.getMessage());
        } catch (NumberFormatException e) {
            System.out.println("Invalid inputs...    " + e.getMessage());
        } catch (IOException e) {
            System.out.println("Resource error...." + e.getMessage());
            // e.printStackTrace();
        } finally {
            try {
                ir.close();
                br.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }

    }

}
```

output

Read two input values :

10

0

Denominator is zeo   / by zero

Process finished with exit code 0

Read two input values :

10

2

5

Process finished with exit code 0

```
package com.thbs.mainpack;

import com.thbs.exceptionpack.ReadInput;

import java.io.IOException;

public class Main {

    public static void main(String[] args) {
        System.out.println(ReadInput.readInput1());
        /*try {
            ReadInput.readInput2();
        } catch (ArithmeticException e) {
            System.out.println("Denominator is zeo    " + e.getMessage());
        } catch (NumberFormatException e) {
            System.out.println("Invalid inputs...    " + e.getMessage());
        } catch (IOException e) {
            e.printStackTrace();
        }
*/
    }
}
```

Readinput.java

```
package com.thbs.exceptionpack;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
public class ReadInput {

    // try with resources ------> try with auto close option
    public static String readInput1()
    {
        try(InputStreamReader ir = new InputStreamReader(System.in);
```

```
        BufferedReader  br = new BufferedReader(ir);) {
        System.out.println("Read two input values : ");
        String str1 = br.readLine();
        String str2 = br.readLine();
        int result = Integer.parseInt(str1) / Integer.parseInt(str2);
        return ("result "+result);

    }
    catch (ArithmeticException e) {
        return ("Denominator is zeo    " + e.getMessage());
    } catch (NumberFormatException e) {
        return ("Invalid inputs...    " + e.getMessage());
    }
    catch (IOException e) {
        return ("problem occured in the console....."+e.getMessage());
    }
}

public static void readInput2() throws IOException ,ArithmeticException,NumberFormatException{
    InputStreamReader ir = new InputStreamReader(System.in);
    BufferedReader  br = new BufferedReader(ir);
    System.out.println("Read two input values : ");
    String str1 = br.readLine();
    String str2 = br.readLine();
    int result = Integer.parseInt(str1) / Integer.parseInt(str2);
    System.out.println(result);
}


// for a checked exception
public static void readInput() {
    InputStreamReader ir = null;
    BufferedReader br = null;
    String str1, str2;
    // InputStreamReader is pipelined with BufferedReader
    try {
        ir = new InputStreamReader(System.in);  // will read one char at a time
        br = new BufferedReader(ir);  // br will be
        System.out.println("Read two input values : ");
        str1 = br.readLine();
        str2 = br.readLine();
        int result = Integer.parseInt(str1) / Integer.parseInt(str2);
        System.out.println(result);
    } catch (ArithmeticException e) {
        System.out.println("Denominator is zeo    " + e.getMessage());
    } catch (NumberFormatException e) {
        System.out.println("Invalid inputs...    " + e.getMessage());
    } catch (IOException e) {
        System.out.println("Resource error...." + e.getMessage());
        // e.printStackTrace();
    } finally {
```

```
        try {
            ir.close();
            br.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }}}
```

output

```
Read two input values :

12

0

Denominator is zeo    / by zero
```

```
package com.thbs.mainpack;

import com.thbs.emppack.Employee;
import com.thbs.exceptionpack.InsufficentLeaveException;

public class Main {

    public static void main(String[] args) {
        Employee employee1 = new Employee(100,15);
        System.out.println(employee1);
        try {
            System.out.println(employee1.applyLeave(10));
        } catch (InsufficentLeaveException e) {
            e.printStackTrace();
        }
        System.out.println(employee1);
        try {
            System.out.println(employee1.applyLeave(10));
        } catch (InsufficentLeaveException e) {
            //e.printStackTrace();
            System.out.println(e.getMessage());
        }
        System.out.println(employee1);


    }
}
```

```java
package com.thbs.exceptionpack;
// this is custom exception is registred as checked exception
//public class InsfficentLeaveException extends Runtime
public class InsufficentLeaveException extends Exception {

   private String msg;

   public InsufficentLeaveException(String msg)
   {
      this.msg=msg;
   }

   @Override
   public String getMessage() {
      return msg;
   }
}
```

Employee{empId=100, balanceLeave=15}

Leave sanctioned.....happy time!!!

Employee{empId=100, balanceLeave=5}

Insufficent Leave....

Employee{empId=100, balanceLeave=5}

Process finished with exit code 0

Some more examples in Exception handling

```java
package com.kannu;

public class Main {

    public static void main(String[] args) {
        // throwable ----->Exception--->1.checked Exception 2.unchecked
Exception
        //1.checked Exception
                //-->IOException,SQLException
        //2.unchecked Exception ----->RuntimeException----->ArrayBoundException
etc

        int i,j,k=0;
        int a[] = new int[4];
        i=8;
        j=0;
        try {
            k=i/j;//Uncheked Exception
            for(int c=0; c<=4;c++)
```

```
            {
                int value = 0;
                System.out.println(value);
            }
        }
        catch (Exception e)
        {
            System.out.println(e);
            System.out.println("can not divide by zero"+e );
        }
       // catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("Maximum number of values is 4");
        }
        System.out.println(k);
    }
}
```

output

java.lang.ArithmeticException: / by zero
can not divide by zerojava.lang.ArithmeticException: / by zero
Maximum number of values is 4
0
Process finished with exit code 0

```
package com.kannu;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.nio.Buffer;

public class Main {

    public static void main(String[] args) {
        // throwable ----->Exception--->1.checked Exception 2.unchecked Excep-
tion
        //1.checked Exception
        //-->IOException,SQLException
        //2.unchecked Exception ----->RuntimeException----->ArrayBoundException
etc

        int i, j, k = 0;
        BufferedReader br = new BufferedReader(new InputStreamReader(Sys-
tem.in));

        //int a[] = new int[4];
        i = 8;
        //j=0;
        try {
            j = Integer.parseInt(br.readLine());
            k = i / j;

        } catch (IOException e) {
            System.out.println("some IO error");
        } catch (ArithmeticException e) {
            System.out.println("can not dive by zero " + e);

        }
        System.out.println(k);
```

```
        }
}
```