DAY-53

-------

EXCEPTION HANDLING

------------------------------------

refer dia:1

Exception is unwanted or unexpected event which occurs in the excution of program, which distrubs the normal flow of the program.

compile time error : The mistake done by the programmer in the code which is caught by the compiler during compilation phase.

run-time error : The faulty inputs given by the user during the execution time leads to runtime error and it is also called as 'EXCEPTION'.

The main objective of exception handling is to achive 'graceful termination of the program'.

refer dia:2

refer dia:3

"Exception handling is the way writing alternative way of code to continue the normal flow of the program. "

EXAMPLE:NORMAL TERMINATION

---------------------------

import java.util.\*;

class Demo

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

System.out.println("enter the value for A");

int a = sc.nextInt();

System.out.println("enter the value for B");

int b = sc.nextInt();

System.out.println("A/B is ="+(a/b));

}

}

OUTPUT:

-------

enter the value for A

10

enter the value for B

2

A/B is =5

NOTE: In the above example user as given valid input hence normal termination of the program can be seen.

EXAMPLE : ABRUPT TERMINATION

-----------------------------

import java.util.\*;

class Demo

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

System.out.println("enter the value for A");

int a = sc.nextInt();

System.out.println("enter the value for B");

int b = sc.nextInt();

System.out.println("A/B is ="+(a/b));

}

}

OUTPUT:

-------

enter the value for A

10

enter the value for B

0

Exception in thread "main" java.lang.ArithmeticException: / by zero

at Demo.main(Demo.java:11)

NOTE: In the above program user as not given right input at the time of execution hence it resulted in runtime error or exception.

In this case JVM acts as DEFAULT EXCEPTION HANDLER and provides the exception details.

NOTE: Exception can be handled with the help of try and catch block.

syntax:

-------

try

{

risky code

}

catch (\*\* e)

{

alternative code to handle the exception

}

EXAMPLE: GRACEFUL TERMINATION by using user difiend exception handler

---------------------------------------------------------------------

import java.util.\*;

class Demo

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

System.out.println("enter the value for A");

int a = sc.nextInt();

System.out.println("enter the value for B");

int b = sc.nextInt();

try

{

System.out.println("A/B is ="+(a/b));

}

catch (Exception e)

{

System.out.println("exception occuurred b value must be non-zero positive number");

}

}

}

// ANY RISKY CODE SHOULD PUT INSIDE THE try and catch block.

OUTPUT:

------

enter the value for A

10

enter the value for B

0

exception occuurred b value must be non-zero positive number

RUN-TIME STACK OR Exception object propogation

-----------------------------------------------

EXAMPLE:

---------

class Demo1

{

public static void wakeup()

{

sleep();

System.out.println("in wakeup method");

}

public static void sleep()

{

System.out.println("in sleep method");

}

public static void main(String[] args)

{

wakeup();

System.out.println("in main method");

}

}

OUTPUT:

-------

in sleep method

in wakeup method

in main method

REFER FIG:

EXAMPLE:

--------

class Demo1

{

public static void wakeup()

{

System.out.println(10/0);

sleep();

}

public static void sleep()

{

System.out.println("in sleep method");

}

public static void main(String[] args)

{

wakeup();

System.out.println("in main method");

}

}

OUTPUT:

-------

Exception in thread "main" java.lang.ArithmeticException: / by zero

at Demo1.wakeup(Demo1.java:5)

at Demo1.main(Demo1.java:16)

REFER FIG:

NOTE : w.r.t above programs we have two choice :

1. use user difiend exception handler in the specific method where the exception as araised or:

2. use the UDEH in main method because anyways at the end the control will come to main method.

EXAMPLE :1

----------

class Demo1

{

public static void wakeup()

{

try

{

System.out.println(10/0);

sleep();

}

catch (Exception e)

{

System.out.println("exception handled in specific method");

}

}

public static void sleep()

{

System.out.println("in sleep method");

}

public static void main(String[] args)

{

wakeup();

System.out.println("in main method");

}

}

EXAMPLE:2

-----------

class Demo1

{

public static void wakeup()

{

System.out.println(10/0);

sleep();

}

public static void sleep()

{

System.out.println("in sleep method");

}

public static void main(String[] args)

{

try

{

wakeup();

System.out.println("in main method");

}

catch (Exception e)

{

System.out.println("exception handled in main");

}

}

}