Exception handling in java (Demo7.java)

Class Test2

{

Psvm(…)

{

Int a,b;

a=10;

b=2;//b=0;

sopln(a/b);//no problem//not defined

}

}

Exception handling

1. Checked
   1. File Not Found.
2. Unchecked
   1. Arithmetic Exception / by 0.
3. Error
   1. Out of memory.
4. Try-catch finally.
5. Throw

try {

    System.out.println(a/b);

}

catch(exception e)

{

    //handling

}

try {

    connection can =new connection;

    System.out.println(a/b);

}

catch(exception e)

{

    //handling

}

finally()

{

    //distruction code

}

Exception handling Any kinds of exception by using building classes

There are many building classes.

Example: -

Arithmetic exception (arithmetic operation), array index out of bound (array), Null pointer exception (reference) (accessing pointer though null pointer), number formant exception, Class not found exception, class cast exception, file not found exception, SQL exception.

Try and catch block: -

Try block: - it is used enclose the code that might all the statement that lightly to range and exception at run time are kept inside try block.

This block will detect exception and thoughts it is catch block.

Catch block: - it is used to handle the exception. It must be after try block only.

It provides suitable message to the user, then user will able to proceed with the application.

Or try block can follow multiple catch blocks that is we can use multiple catch block with the single try.

Syntax: -

try

{

//statement to be checked

}

Catch (exception ref)

{

// handling code

}

Catch (exceptionClass2 ref)

{

//handling code

}

Example: -

Int a=0;

Int b=0;

Try

{

Int c=b/a;

Sopln(“The result is”+c);

}

Catch (ArithmeticException c)

{

Sopln(“Divided by Zero”);

}

Int sum=a+b;

Sopln(“sum is” + sum);

}//psvm

}//class

Finally block: it is execute important code when exception have occur closing an exception and closing file etc.

Finally, block is away executed weather exception occur or not.

Finally, block is used to put cleanup code.

Example: -

Psvm (String []args)

{

Try

{

Int data =25/5;

Sopln(data);

}

Catch (Arithmetic Exception)

{

Sopln(e);

}

Finally

{

Sopln(“finally block always executed”);

}

Sopln(“rest of code..”)’

}

}

**Through keyword: -**

Through keyword is used to throw of an exception explicitly.

We can through either checked or unchecked exception.

Void validate (int age)

{

If(age<18)

Throw Exception e;

//throw new ArithmeticException ( “invalid age”);

else

sopln(“you can vote”);

}

**Throws keyword: -**

Throws keyword is used to declare and exception it gives information of the programmer there may occur exception show it is better for programmer to provide exception handling code so the normal flow can be maintained.

Void validate(int age) throws ArithmeticExecption

{

if(age<18)

throws new ArithmeticException(“Invalid age”);