DAY-56

-------

Examples for unchecked exceptions:

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

note: all the sub-classes under RUNTIME EXCEPTIONS are unchecked exceptions

1. Arithmetic exception:

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

EXAMPLE:

--------

// Arithmetic exception

class Demo

{

public static void main(String[] args)

{

int i = 10;

int j = 0;

float f = i/j;

System.out.println(f);

}

}

OUTPUT:

--------

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

at Demo.main(Demo.java:8)

2. NullPointer Exception

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

EXAMPLE:

--------

// NullPointer exception

class Demo

{

public static void main(String[] args)

{

String s = null;

System.out.println(s.length());

}

}

OUTPUT:

--------

Exception in thread "main" java.lang.NullPointerException

at Demo.main(Demo.java:7)

3. ArrayIndexOutOfBounds exceptions:

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

EXAMPLE:

--------

// ArrayIndexOutOfBoundsException

class Demo

{

public static void main(String[] args)

{

int a[] = {1,2,3,4,5};

System.out.println(a[5]);

}

}

OUTPUT:

--------

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5

at Demo.main(Demo.java:7)

4.StringIndexOutOfBoundsException:

EXAMPLE:

--------

// StringIndexOutOfBoundsException:

class Demo

{

public static void main(String[] args)

{

String s = "sagar";

System.out.println(s.charAt(5));

}

}

OUTPUT:

--------

Exception in thread "main" java.lang.StringIndexOutOfBoundsException: String index out of range: 5

at java.lang.String.charAt(String.java:658)

at Demo.main(Demo.java:7)

5. ClassCastException:

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

EXAMPLE:

---------

// ClassCastException:

class A

{

}

class B extends A

{

}

class Demo

{

public static void main(String[] args)

{

A a = new A();

B b = new B();

//B b = (B)a;

A a1 = b;

}

}

OUTPUT:

-------

Exception in thread "main" java.lang.ClassCastException: A cannot be cast to B

at Demo.main(Demo.java:14)

NOTE: In java programs we are able to keep sub-class(child class) object reference value in the super-class(parent class) reference variable,but

parent class object reference value can not be kept in child class variable.

If we try to do so we will get classCastException.

6.ClassNotFoundException:

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

EXAMPLE:

--------

// ClassNotFoundException:

class A

{

{

System.out.println("class loading...!");

}

}

class Demo

{

public static void main(String[] args)

{

Class c = Class.forName("XYZ");

}

}

OUTPUT:

-------

emo.java:12: error: unreported exception ClassNotFoundException; must be caught or declared to be thrown

Class c = Class.forName("XYZ");

^

EXAMPLES FOR ERRORS IN JAVA:

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

1. OutOfMemory Error:

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

EXAMPLE:

---------

// OutOfMemoryError:

class Demo

{

public static void main(String[] args)

{

boolean ar[] = new boolean [2147483647];

}

}

OUTPUT:

--------

Exception in thread "main" java.lang.OutOfMemoryError: Requested array size exceeds VM limit

at Demo.main(Demo.java:6)

2. stackOverflow Error:

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

refer dia:1

EXAMPLE:

--------

// StackOverflowError:

{

public static void main(String[] args)

{

fun1();

}

static void fun1()

{

System.out.println("inside fun1");

fun1();

}

}

OUTPUT:

--------

Exception in thread "main" java.lang.StackOverflowError

EXAMPLE FOR CHECKED EXCEPTION:

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

EXAMPLE:

--------

import java.io.\*;

class Demo

{

public static void main(String[] args)

{

try

{

FileReader r = new FileReader("abc.txt");

}

catch (FileNotFoundException e)

{

System.out.println("file not found");

}

}

}

OUTPUT:

-------

file not found

'throw' keyword:

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

-->'Explicitly creating exception object'.

case-1

-------

class Demo

{

public static void main(String[] args)

{

System.out.println(10/0);

}

}

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

at Demo.main(Demo.java:5)

note: exception handled by jvm(default)

case-2

-------

class Demo

{

public static void main(String[] args)

{

throw new ArithmeticException("/ by zero");

}

}

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

at Demo.main(Demo.java:5)

note: exception handled by jvm(manually)

As a programmer we can create a exception explicitly and handover to the jvm manually.To perform this activity we need thorw keyword.

EXAMPLE:

---------

import java.util.\*;

class Demo

{

public static void main(String[]args)

{

Scanner sc = new Scanner(System.in);

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

int a = sc.nextInt();

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

int b = sc.nextInt();

System.out.println(divide(a,b));

}

public static int divide (int a ,int b)

{

if (b<0)

{

throw new ArithmeticException ("/ by zero");

}

else if(b>a)

{

throw new ArithmeticException ("b>a");

}

return a/b;

}

}

OUTPUT:

-------

enter the value of A

10

enter the value of B

2

5

enter the value of A

10

enter the value of B

0

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

at Demo.divide(Demo.java:23)

at Demo.main(Demo.java:11)

enter the value of A

10

enter the value of B

20

Exception in thread "main" java.lang.ArithmeticException: b>a

at Demo.divide(Demo.java:21)

at Demo.main(Demo.java:11)

NOTE: throw keyword is best suitable for customized exception.

case-1

-------

class Demo1

{

static ArithmeticException e = new ArithmeticException();

public static void main(String[] args)

{

throw e;

}

}

OUTPUT:

-------

Exception in thread "main" java.lang.ArithmeticException

at Demo1.<clinit>(Demo1.java:3)

case-2

-------

class Demo1

{

static ArithmeticException e ;

public static void main(String[] args)

{

throw e;

}

}

OUTPUT:

--------

Exception in thread "main" java.lang.NullPointerException

at Demo1.main(Demo1.java:6)

NOTE: First we must create the exception object using new keyword and then we can throw that exception. Without creating exception object if we

try to throw the exception we will get nullpointerexception because there is no exception object in the memory.

case-3

------

class Demo1

{

public static void main(String[] args)

{

System.out.println(10/0);

System.out.println("hello world");

}

}

OUTPUT:

-------

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

at Demo1.main(Demo1.java:5)

case-4

-------

class Demo1

{

public static void main(String[] args)

{

throw new ArithmeticException("/by zero");

System.out.println("hello world");

}

}

OUTPUT:

-------

Demo1.java:6: error: unreachable statement

System.out.println("hello world");

^

NOTE: The line below the exception is not excecuted because jvm stops the flow of execution immeaditly after finding exception object.

In case of throw keyword the lines below throw statements are unreachable statments.

case-5

-------

class Demo1

{

public static void main(String[] args)

{

throw new Demo1();

}

}

OUTPUT:

-------

Demo1.java:5: error: incompatible types: Demo1 cannot be converted to Throwable

throw new Demo1();

^

case-6

-------

class Demo1 extends RuntimeException

{

public static void main(String[] args)

{

throw new Demo1();

}

}

OUTPUT:

-------

Exception in thread "main" Demo1

at Demo1.main(Demo1.java:5)

NOTE: All the java objects are not throwable.Only throwable objects can be thrown using throw keyword.