Exception Handling

Exception handling doesn't mean fixing an exception, We need to provide an alternative solution for the free flow of program.

(An error in a program is called bug. Removing errors from program is called debugging.)



Error:

if our program is having an error. The program will terminated.

We cant save our program

There are basically three types of errors in the Java program:

Compile time errors: (Occurs due to wrong syntax)

Errors which occur due to syntax or format is called compile time errors. These errors are detected by java compiler at compilation time. Desk checking is solution for compile-time errors.

Runtime errors: (due to ineffiency of the syntax)

These are the errors that represent computer inefficiency. Insufficient memory to store data or inability of the microprocessor to execute some statement is examples to runtime errors. Runtime errors are detected by JVM at runtime.

Logical errors:(due to bad logic in the program)

These are the errors that occur due to bad logic in the program. These errors are rectified by comparing the outputs of the program manually.

Exception: An abnormal event in a program is called Exception.

/if your program is having an exception ,the program will be terminated.

But we can save our program by handling the exception.

All Exceptions occur at runtime only but some are detected at compile time and some are detected at runtime.

Exceptions that are checked at compile time by the java compiler are called “Checked exceptions”. eg: ClassNotFoundException, NoSuchMethodException, NoSuchFieldException etc.

Exceptions that are checked at run time by the JVM are called “Unchecked exceptions”.

eg: ArrayIndexOutOfBoundsException, ArithmeticException, NumberFormatException etc.



Common Scenarios of Java Exceptions

ArithmeticException occurs : If we divide any number by zero, there occurs an ArithmeticException.

Ex: int a=50/0;//ArithmeticException

NullPointerException occurs : If we have null value in any variable, obtaining the length of that variable occurs an NullPointerException.

Ex:String s=null;

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

ArrayIndexOutOfBoundsException occurs If you are inserting any value in the wrong index, it would result ArrayIndexOutOfBoundsException

Ex: int a[]=new int[5];

a[10]=50; //ArrayIndexOutOfBoundsException

List of important built-in exceptions Exception

ArithmeticException Thrown when an exceptional condition has occurred in an arithmetic operation

ArrayIndexOutOfBoundsException Thrown to indicate that an array has been accessed with an illegal index

ClassNotFoundException Thrown when we try to access a class whose definition is not found

FileNotFoundException Raised when a file is not accessible or does not open

IOException Thrown when an input-output operation failed or interrupted

NoSuchFieldException Thrown when a class does not contain the field(or variable) specified

NullpointerException Raised when referring to the members of a null object List of important built-in exceptions Exception Class IMeaning

NumberFormatException Raised when a method could not convert a string in to a numeric format

RuntimeException This represents any exceptions which occurs during runtime

StringIndexOutOfBoundsException Thrown by String class methods to indicate that an index is either negative or greater than the size of the string Except

Exception Handling An exception can be handled by the programmer where as an error cannot be handled by the programmer.

Types of exceptions:

Predefined exception: the sun micro system developed the predefined exception classes and supplied as the part of  jdk.by using this we can handle the exceptions

* Ex: class not found Exception, Arithmetic Exception

**User defined exceptions** :Based  on the user requirement create its own exception classes is called "user defined exception classes"

* Ex: Atm Exception, Bank Exception
* User defined Exceptions: the Exception are created by user based on their requirement

➢Exception occurs only either inside the block or a method.

➢When exception has raised, that block or method creates an exception object which contains the complete information of that exception including.

We are using try-catch-finally need maintain the order

* try-catch
* try-finally
* try-catch-finally
* try(resourses)

syntax:

try

{

//we need to write suspicious code

}

Catch(exceptionname )

{

//catch the exception in try block

}

Finally

{

//it will execute the irrespective of the exection

}

Important points

* A try block never exit a without a catch or finally
* We can write multiple catch blocks for a single try block to handle multiple exceptions
* We cant have multiple finally blocks for a single try blocks
* Always try to write minimum code inside the try block
* Without writing the catch block exception will not be handled

Exception Keywords

‘try’ keyword:

The "try" keyword is used to specify a block where we should place an exception code. It means we can't use try block alone. The try block must be followed by either catch or finally.

‘catch’ keyword:

The "catch" block is used to handle the exception. It must be preceded by try block which means we can't use catch block alone. It can be followed by finally block later.

‘throw’ keyword

It is used to throw userdefined exceptions . we can throw either checked exceptions or unchecked exceptions

‘throws’ keyword

It is used to escape from exception handling .we can throws only checked exceptions

‘Final’ keyword:

The final keyword in java is used to restrict the user.

‘final’ keyword is used in three ways:

It is used to declare constants as:

✓ Final double Pi=3.14159; //PI is constant

It is used to prevent inheritance as:

✓ Final class A // sub class to A cant be created

It is used to stop method Overriding as:

✓ Final sum() // sum() method can’t be overridden

The final can be:

➢Variable

➢Method

➢Class

‘final’ Variable: If you make any variable as final, you cannot change the value of final variable(It will be constant).

‘final’ Method: If you make any method as final, you cannot override it. Because they are not available to sub classes. So only overloading is possible.

‘final’ Class: If you make any class as final, you cannot extend it. It means sub classes can’t be created to final class