Last updated: Oct 13, 2021

Selenium Java Training - Session 10 - Java (Part 8) - Interfaces and Exception Handling

Java (Part 8) - Interfaces and Exception Handling

Interfaces

The purpose of an interface is to just to declare all the functionalities required before actually implementing them.

- Interfaces looks similar to Classes and are extensions of abstract classes
- Create an interface say 'Bank' in Eclipse IDE and create variables & methods inside it as shown here
- Variables in the interfaces are of static and final type
- In abstract classes, we can have both methods (i.e. implemented and non-implemented), where as in interfaces, we cannot implement any methods.
- Classes use **implements** keyword to implement any interface - Demonstrate here
- Classes implementing an interface can have their own specific methods apart from methods which are acquired from an interface - Demonstrate here
- Objects cannot be created for an interface - Demonstrate
- Object can be created for the Classes which are implementing the interfaces, for accessing interface defined methods and class specific methods - Demonstrate
- Follow the below steps to provide the access the interface specific methods and not to access the class specific methods
  - Create an object for the Class which is implementing the interface
  - Assign the object of the class to the interface reference variable
  - Using the interface reference variables, we can now access only the methods which are declared in the interface - Demonstrate here

Exception Handling

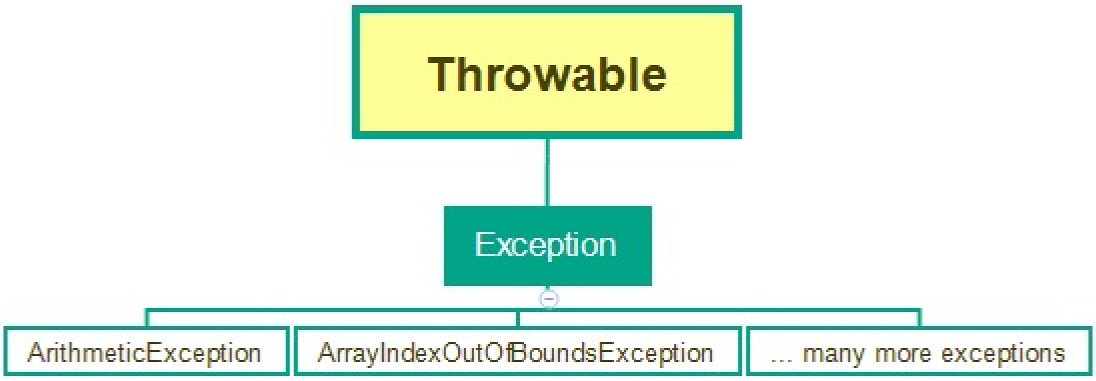
Exception is nothing but an error which is occurred during runtime i.e. during program execution

- If an exception has occurred during program execution at any step, the steps which are after the exception wont be executed - Demonstrate here

try catch blocks

- We can handle the exceptions using the **try catch** blocks
    - Handling the exceptions is known as Exception Handling
    - Syntax: View here
    - Explain the flow of try catch block - view here
    - Demonstrate a program having code to handle the exception using try catch blocks - Demonstrate here
    - In the above Syntax image, 'Exception' is the Class name and 'e' is the object reference which can catch the exception (i.e. object) thrown from try block

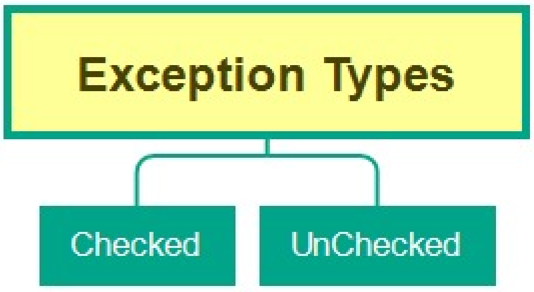
Exceptions Hierarchy



- Demonstrate ArithmeticException and handle it using 'ArithmeticException' class in catch block - Demonstrate here
- Demonstrate ArrayIndexOutOfBoundsException and handle it using 'ArrayIndexOutOfBoundsException' class in catch block - Demonstrate here
- Exception class is the parent class of all the Exception Classes like ArithmeticException and ArrayIndexOutOfBoundsException classes and can handle them
- Throwable class is the grant parent class of all the Exception Classes like ArithmeticException and ArrayIndexOutOfBoundsException classes and can handle them

Exception Types

- Exceptions can be categorized as below:



- Unchecked exceptions are the exceptions that are not checked by compiler and will occur only during execution - Demonstrate AirthmeticException

- Checked Exceptions are the exceptions that are checked by the compiler - Demonstrate FileNotFoundException
- Handling Checked Exceptions using try .. catch block
- Ignoring Checked Exceptions using throws keyword

By,
Arun Motoori