HEXAWARE

Java - Exception

Session Objective

- Scenario for Exception
- Error and Exception
- Checked and Unchecked Exception
- Handling of Exception
- Keywords in Exception Handling
- New Features of Exception Handling in Java7 and Java8

Scenario for Exception

- Exception is an abnormal condition.
- In java, exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime
- Exception Handling is a mechanism to handle runtime errors such as ClassNotFound, IO, SQL, Remote etc.
- The core advantage of exception handling is to maintain the normal flow of the application. Exception normally disrupts the normal flow of the application that is why we use exception handling. Let's take a scenario:

Scenario for Exception cont.…

When Executing an Application

```
statement 1;
statement 2;
statement 3;
statement 4;
statement 5;//exception occurs
statement 6;
statement 7;
statement 8;
statement 9;
statement 10;
```

Suppose there is 10 statements in your program and there occurs an exception at statement 5, rest of the code will not be executed i.e. statement 6 to 10 will not run. If we perform exception handling, rest of the exception will be executed. That is why we use exception handling in java.

Error and Exception

- Error: Any departure from the expected behavior of the system or program, which stops the working of the system is an error.
- Exception: Any error or problem which one can handle and continue to work normally.
- In Java a compile time error is normally called an "error," while a runtime error is called an "exception.“
- Errors don't have subclasses while exception has two subclasses, they are compile time exception or checked exception and runtime or unchecked

Checked and Unchecked Exception

Checked Exception

Exceptions that are checked at compile-time are called checked Exceptions.

Checked exceptions are exceptions that the designers of Java feel that your programs absolutely must provide for an exception.

Whenever you code a statement that could throw a checked exception, your program must handle it .

Checked exceptions in Java extend the java.lang.Exception class.

Checked and Unchecked Exception cont..

HEXAWARE

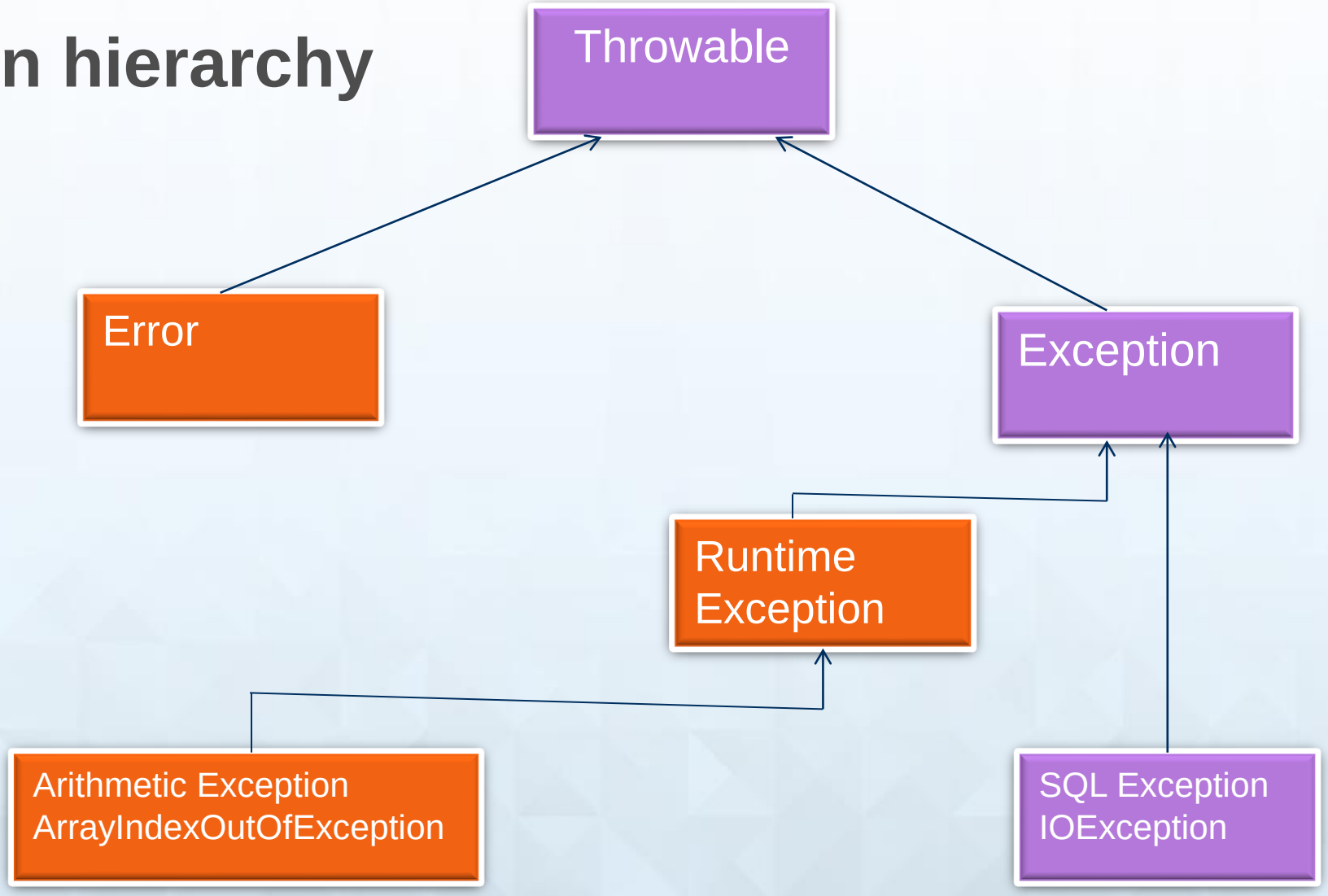
Unchecked Exception :

- The exceptions that are not checked at compile time.
- Unchecked Exception is the subclass for java.lang.RuntimeException and java.lang.Error .
- Exceptions should not normally occur during the normal execution of application
- It will fire only because of some abnormal situation and wrong input.

Example:

- ✓ ArrayIndexOutOfBoundsException
- ✓ NullPointerException
- ✓ ArithmeticException

Exception hierarchy



Key Words Used to Handle Exception

- Try
- Catch
- Finally
- Throw
- Throws

Key Words Used to Handle Exception

- Try –catch-finally

  Use to handle an exception

- Throws

  The throws keyword is used to declare an exception. It gives an information to the programmer that there may occur an exception so it is better for the programmer to provide the exception handling code so that normal flow can be maintained.

  Exception Handling is mainly used to handle the checked exceptions. If there occurs any unchecked exception such as NullPointerException, it is programmers fault that he is not performing check up before the code being used

Key Words Used to Handle Exception

- When to use Try–catch
- When to use Throw
- When to use Throws

Demos

Try - catch

Throws

Multi Catch

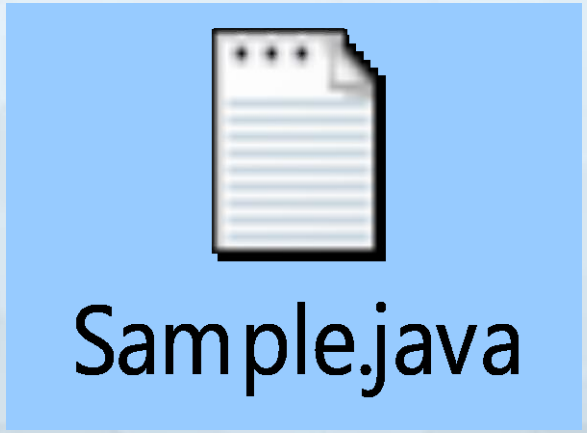
- Java SE 7 provides a new multi-catchclause.
- The new multi-catch clause reduces the amount of code you must write
- Another benefit of the multi-catch clause is that it makes it less likely that you will attempt to catch a generic exception.
- The type alternatives that are separated with vertical bars cannot have an inheritance relationship. You may not list both a FileNotFoundException and an IOException in a multi-catch clause.

Syntax:

```
try
{ // execute code that may throw 1 of the 3 exceptions below.
}
 catch(SQLException | IOException e )
{
 logger.log(e);
}
```


Throwing an Exception

- You can throw an exception that has already been caught. Note that there is both a throws clause and a throw statement.
- You can create custom exception classes by extending Exception or one of its subclasses.
- Custom exceptions are never thrown by the standard Java class libraries. To take advantage of a custom exception class, you must throw it yourself.



LET US THINK

Let Us Think

1) A null pointer Exception must be caught by using a try-catch statement
   a)True
   b)False
2) Which of the following types are all checked exceptions ?
   a) Error
   b)Throwable
   c) Runtime exception
3)Which key word is used to declare an exception
   a)Try
   b) Throws
   c)Finally
4)Which key word is used in user defined exception
   a) Try
   b) Finally
   c) Throws

Exception - Assignment

Exception
Assignment

HEXAWARE

Innovative Services

Passionate Employees

Delighted Customers

Thank you

www.hexaware.com