CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

INSTITUTE : UIE

DEPARTMENT : CSE

Bachelor of Engineering (Computer Science & Engineering)

PROJECT BASED LEARNING IN JAVA

(20CST-319/20ITT-319)

TOPIC OF PRESENTATION:

Introduction to Exceptions. Difference between error and exception

Lecture Objectives

In this lecture, we will discuss:

Introduction to Exceptions.
Difference between error and exception.

JAVA - EXCEPTIONS

- An exception (or exceptional event) is a problem that arises during the execution of a program.
- When an **Exception** occurs the normal flow of the program is disrupted and the program/Application terminates abnormally, which is not recommended, therefore, these exceptions are to be handled.
- An exception can occur for many different reasons. Following are some scenarios where an exception occurs.
    - A user has entered an invalid data.
    - A file that needs to be opened cannot be found.
    - A network connection has been lost in the middle of communications or the JVM has run out of memory.
    - Some of these exceptions are caused by user error, others by programmer error, and others by physical resources that have failed in some manner.

Exceptions Methods

Following is the list of important methods available in the Throwable class.

Sr.No.	Method & Description
1	**public String getMessage()** Returns a detailed message about the exception that has occurred. This message is initialized in the Throwable constructor.
2	**public Throwable getCause()** Returns the cause of the exception as represented by a Throwable object.
3	**public String toString()** Returns the name of the class concatenated with the result of getMessage().
4	**public void printStackTrace()** Prints the result of toString() along with the stack trace to System.err, the error output stream.
5	**public StackTraceElement [] getStackTrace()** Returns an array containing each element on the stack trace. The element at index 0 represents the top of the call stack, and the last element in the array represents the method at the bottom of the call stack.
6	**public Throwable fillInStackTrace()** Fills the stack trace of this Throwable object with the current stack trace, adding to any previous information in the stack trace.

Error v/s Exception

- **Error:** An Error indicates serious problem that a reasonable application should not try to catch.
- **Exception:** Exception indicates conditions that a reasonable application might try to catch.
- **Exception Hierarchy :**
  - All exception and errors types are subclasses of class **Throwable**, which is base class of hierarchy.
  - One branch is headed by **Exception**. This class is used for exceptional conditions that user programs should catch.
  - NullPointerException is an example of such an exception.
  - There is an important subclass of Exception, called RuntimeException.Exceptions of this type are automatically defined for the programs that you write and include things such as division by zero and invalid array indexing.

Exception and Error

Exceptions and errors both are subclasses of Throwable class. The error indicates a problem that mainly occurs due to the lack of system resources and our application should not catch these types of problems. Some of the examples of errors are system crash error and out of memory error. Errors mostly occur at runtime that's they belong to an unchecked type.

Exceptions are the problems which can occur at runtime and compile time. It mainly occurs in the code written by the developers. Exceptions are divided into two categories such as checked exceptions and unchecked exceptions.

Sr. No.	Key	Error	Exception
1	Type	Classified as an unchecked type	Classified as checked and unchecked
2	Package	It belongs to java.lang.error	It belongs to java.lang.Exception
3	Recoverable/ Irrecoverable	It is irrecoverable	It is recoverable
4		It can't be occur at compile time	It can occur at run time compile time both
5	Example	OutOfMemoryError ,IOError	NullPointerException , SqlException

Summary of Differences	
ERRORS	**EXCEPTIONS**
Recovering from Error is not possible.	We can recover from exceptions by either using try-catch block or throwing exceptions back to caller.
All errors in java are unchecked type.	Exceptions include both checked as well as unchecked type.
Errors are mostly caused by the environment in which program is running.	Program itself is responsible for causing exceptions.
Errors occur at runtime and not known to the compiler.	All exceptions occurs at runtime but checked exceptions are known to compiler while unchecked are not.
They are defined in java.lang.Error package.	They are defined in java.lang.Exception package
Examples :	**Examples :**
java.lang.StackOverflowError, java.lang.OutOfMemoryError	Checked Exceptions : SQLException, IOException
	Unchecked Exceptions : ArrayIndexOutOfBoundException, NullPointerException, ArithmeticException.

Example of Error

```
public class ErrorExample {
  public static void main(String[] args){
   recursiveMethod(10)
  }
  public static void recursiveMethod(int i){
   while(i!=0){
     i=i+1;
     recursiveMethod(i);
   }
  }
}
```

Output

```
Exception in thread "main" java.lang.StackOverflowError
  at ErrorExample.ErrorExample(Main.java:42)
```

Example of Exception

```
public class ExceptionExample {
  public static void main(String[] args){
   int x = 100;
   int y = 0;
   int z = x / y;
  }
}
```

Output

```
java.lang.ArithmeticException: / by zero
  at ExceptionExample.main(ExceptionExample.java:7)
```

QUIZ:

1. When does Exceptions in Java arises in code sequence?
a) Run Time
b) Compilation Time
c) Can Occur Any Time
d) None of the mentioned

Summary:

In this session, you were able to :

- Learn about Introduction to Exceptions. Difference between error and exception

References:

Books:

1. Balaguruswamy, *Java.*
2. A Primer, E.Balaguruswamy, *Programming with Java,* Tata McGraw Hill Companies
3. John P. Flynt Thomson, *Java Programming.*

ONLINE NOTES LINKS:

https://www.tutorialspoint.com/java/java_exceptions.htm
https://www.javatpoint.com/exception-handling-in-java
https://www.geeksforgeeks.org/exceptions-in-java/
https://beginnersbook.com/2013/04/java-exception-handling/
https://docs.oracle.com/javase/tutorial/essential/exceptions/definition.html

VIDEO LINKS:

https://www.javatpoint.com/exception-handling-in-java
https://www.edureka.co/blog/java-exception-handling
https://marcus-biel.com/advanced-exception-handling-in-java/

THANK YOU