Assignment 5 May 2022

**What are run time exceptions in Java. Give example?**

🡺*Runtime Exception is a child classes of exception, these are un checked exceptions*

*E.g. 🡺 athematic exception, null pointer exception*

**What is the difference between ClassNotFoundException and NoClassDefFoundError in Java?**

🡺 *Noclassdeffounderror is a error so it is a unchecked exception*

*🡺Classnotfound is a checked exception*

*🡺At hardcoded time if a corresponding .class file is not present it will throw noclassdeffound error*

*🡺For dynamically provided class name at runtime if corresponding .class file is not there then it give noclassnotfound exception*

**What is StackOverflowError in Java? Give an Example?**

**🡺** *it is a child class of virtual machine error, vme is a child class of error, error is a child class of throwable,*

*Whenever a we are perform recursive method it causes error*

**E.g 🡺** class Test{

Public static void main(String[] args){

m1();

}

Public static void m1(){

m1();

}

**}**

**What are try with resource?**

🡺it got introduced in i.7 version *Once control reaches to end of the try block it will automatically closes the resource in try bock implicitly*

**Can we handle more than one Exception in single catch block?**

**🡺** *yes from java 1.7*

*class Test*

*{*

*public static void main(String[] args) throws Exception*

*{*

*try*

*{*

*System.out.println(10/0);*

*Strings=null;*

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

*}*

*catch (ArithmeticException|NullPointerException e)*

*{*

*System.out.println(e);*

*}*

*}*

**What are the benefits of try-with-resources?**

**🡺***what ever resources we are opening in try block we are not required to close the resources explicitly.*

*It improves readability, reduces Complexity, reduces length of code.*

*E.g🡺*

***Before 1.6V After 1.7V***

***try{ try{***

***BufferReader br=new BufferReader*** ***BufferReader br=new BR(new FR(“”))***

***(new FineReader(“abc.txt”)); }***

***}***

***Catch(IoException e){ Catch(IoException e){***

***//handling code //handling code***

**} }**

**finally{**

**br.close();**

**//closing the resource in finally block**

**}**

**What is fail-fast and fail safe?**

*🡺it is not generally permissible for one thread to modify a Collection while another thread is iterating over it, the results of the iteration are undefined under these circumstances. Some Iterator implementations) may choose to throw this* ***ConcurrentModificationException***

*if this behavior is detected. Iterators that do this are known as fail-fast iterators, as they fail quickly and cleanly,*

*🡺* ***Note*** *that this exception does not always indicate that an object has been concurrently modified by a different thread.*

***For example*** *if a thread modifies a collection directly while it is iterating over the collection with a fail-fast iterator, the iterator will throw this exception.*

*Fail Safe🡺 if one thread is iterating over a collection and other thread is modification no* ***ConcurrentModificationException is thrown***

**Can we throw checked exceptions from the static block**?

**🡺**No not only checked but un checked exceptions can not be thrown because We don’t have any control over the execution of static blocks because they are executed when class is loaded. We cannot call static block at our will, so throwing exception from static initialization block does not make any sense.

**Do checked exceptions occur at compile time?**

**🡺***No, All the exceptions occurs at runtime only.*

**Are compile-time errors exceptions?**

**🡺**No, compile time errors are due to syntax errors(syntactical errors).

**Does Java compiler check Runtime exceptions at compilation and Why?**

**🡺***Runtime exceptions can occur anywhere in a program, and in a typical one they can be very numerous. Having to add runtime exceptions in every method declaration would reduce a program's clarity. Thus, the compiler does not require that you catch or specify runtime exceptions (although you can).*

**🡺***Most Runtime Exceptions come from a problem in your code logic, rather than a condition that fails at runtime in ways that you cannot predict or prevent. A try/catch is for handling exceptional situations, not flaws in your code, Use your catch blocks to try to recover from situations you can't guarantee wlll succeed*

**What happens when an exception is thrown by the main method?**

**🡺** *if an exception thrown by main method then it is responsible of jvm to handle the exception , it handled by default exceptional handler then it will be a abnormal termination*

**What is unreachable catch block error in Java?**

**🡺** *While using multiple catch statements, it is important to remember*

*that exception sub classes inside catch must come before any of their*

*super classes otherwise it will lead to compile time error.*

**What is the difference between final, finally and finalize in java?**

***Final.***

*final is a modifier applicable for classes, methods and variables. If a*

*class declared as final then we can't extend that class.*

*i.e. we can't create child class for that class.*

*. If a method declared as final then we can't override that method in*

*the child class.*

*. If a variable declared as final then it will become constant and we*

*can't perform re-assignment for that variable.*

***Finally***

*finally is a block always associated with try catch to maintain cleanup code****.***

***Finalize***

*finalize() is a method which is always invoked by garbage*

*collector just before destroying an object to perform*

*cleanup activities****.***

**Does catch block rethrow an exception in java?**

**🡺**Yes from 1.7 version this concept is called rethrowing exception, it is used to converting one exception type to another exception type

**e.g.**

**try{**

**System.out.println(10/0);**

**}**

**Catch(ArthimeticException ae){**

**throw new NullPointerException();**

**}**