Exception handling

Exception is an object or memory when un expected or abnormal thing/conditions happen during the execution of program is known as Exception. To handle generated exception using some technique is known as exception handling.

Java program

Compile the program	Run the program
javac Demo.java	java Demo
compile time error	run time error
syntax error or typo error	
if we can t compile we cant get byte	
code and we can't run the program	

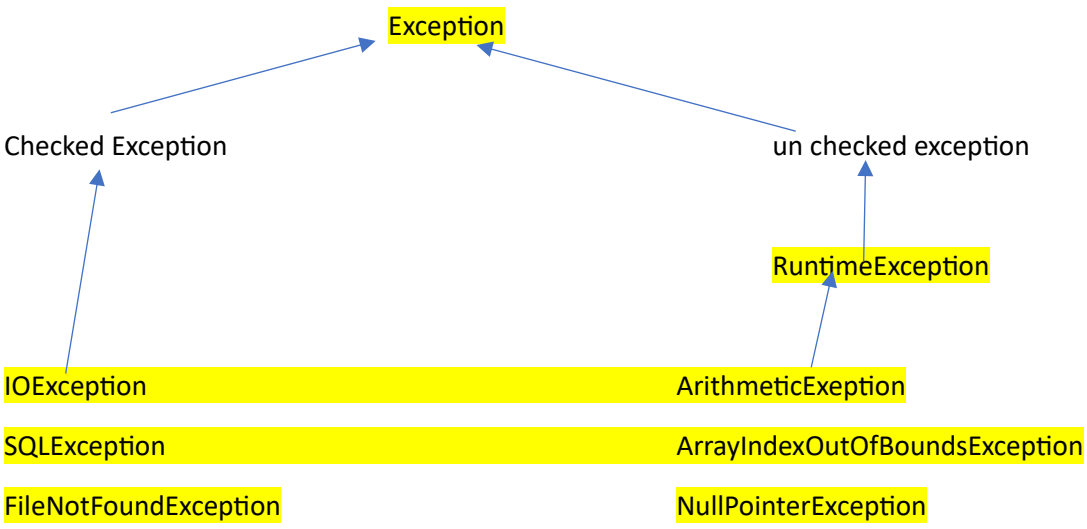
Run time error

Error Exception

Error and Exception both are pre defined classes part of lang package.

Error : The error which generated at run time which can't handle it Ex :JVM crash , software or hardware issue etc.

Exception : it is a type of run time error which we can handle it : Ex : divided by zero.



All unchecked exception are sub class of RuntimeException.

To handle both the type of exception java provided 5 keyword.

Ie

1. try
2. catch
3. finally
4. throw
5. throws

unchecked exception using try catch block

syntax

```
try {
        try block
}catch(Exception e) {
        catch block
}
```

Try with single catch is good

If

1. if any exception generate we want to do common task like display error message.
2. If we don't know which code generate which type of exception.

Try with multiple catch block

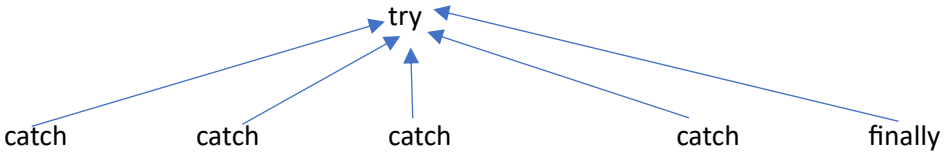
```
try {

}catch(ArithmeticException e) {

}catch(ArrayIndexOutOfBoundsExcepton | NullPointerException  e) {

}
```

Finally block

Try block : the code one line or more than one line code which generate exception we need to keep in try block.

Catch : this block only execute if any exception generate. No exception no catch block.

Finally is a type of block which execute compulsory if any exception generate or not.



catch finally catch

catch finally

file handling program

data base connectivity program ie jdbc

```
try{
        open file
        read and write

}catch(Exception e){

}finally {
        close file                  it is use to close the resource
}
```

throw : throw keyword is use to throw any pre defined or user defined (custom exception) depending upon you conditions.

Using throw we can raise or generate pre defined or user defined exception.

Syntax

throw new Exception();

or

throw new ExceptionSubClass();

or

throw new UserDefinedException();

by default every sub class constructor super() parameter present.

Which help to call super class empty constructor.

We can take the help of super() parameter to call other constructor of super class.

throws throws keyword is use to throw the checked or unchecked exception to caller methods.

Throws keyword we use with method signature.

void display() throws Exception {

}

Checked exception

Checked exception we can't avoid we need to handle mandatory.

Using try – catch or throws.

Checked exception check twice at compile time as well as run time.

Unchecked exception like Arithmetic Exception divided by zero

Array index out of bounds exception wrong index position.

Unchecked exception we can avoid with proper code.

Multi threading

Program : set of instruction to perform specific task.

Processor :processor is responsible to execute the code or process the code.

Process : time taken to execute the code.

Thread : Thread is a small execution of a code within a process.

Thread also known as light weighted process.

It take less resources or memory of our machine.

Java is thread based programming language

In java, inside a main method always one default thread execute.

currentThread() is a pre defined method part of Thread class and method is static method.

Method return type is same class reference.

Thread is pre defined class part of lang package.

Thread t = Thread.currentThread();

t provided default thread details.

Thread[main,5,main];

main → name of the thread

5→ priority of the thread

main → group of the thread

t.setName("Demo Thread");

t.setPriority(2); min 1 max 10 norm priority

t.setPriority(Thread.MAX_PRIORITY);

String name = t.getName();

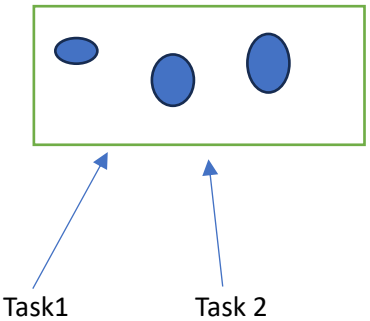
int i = t.getPriority();

Multi tasking :

Process based

Thread based

Multi tasking using thread base is faster than multi tasking us



In Java we can achieve multi threading using 2 ways

1. Extends Thread class
    a. We need to create normal java class and that class must be extends Thread class.
    b. Now inside main method create Thread class reference.
    c. With help of that reference we need to call start() method.
    d. Start() is a pre defined method part of thread class is use to ready to start the thread.
    e. Start method internally call run() method. run method is a part of thread with empty body.
    f. If we want to do custom logic we need to override run method in user defined class.

2. Implements Runnable interface.
    a. We need to create normal java class and that class must be implements Runnable interface.
    b. Runnable interface contains one method ie run() method. we need to override run method mandatory.
    c. We need to create the thread class object and pass the Runnable interface reference.
    d. Using reference call start() method it will call run methods.
    e. Inside run method give your own custom logic.
    f.

When our class extends any class that class can't extends any other class.

We are doing some task

Book the ticker

Transfer the amount

Add the product cart.

Number of people is equal to number of thread.

Life cycle of thread

Created	ready to run	running	destroy
obj1	obj1.start();	run	i=10,j=10
t1	t1.start();	run	

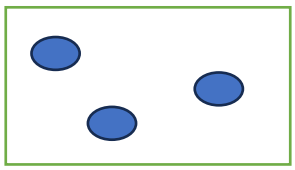
isAlive()	: running or not or status of thread
sleep()	:it is use to pause the execution of thread
wait()	: it is use to suspend the thread
notify()	: it is use to resume the thread only one
notifyAll()	: it is use to resume all suspended thread

synchronization: it is use to block or lock the thread. It allow only one thread to use all resource at time.

To achieve the synchronization we have to use synchronized keyword. This keyword we can use with method.

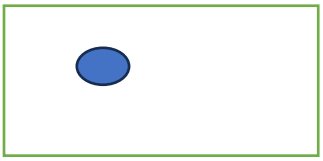
Inner thread communicate

wait()	suspend the thread
notify()	resume only one
notifyAll()	resume all



Pts

1. More than one thread in same memory.
2. Method must be synchronized
3. These three method throws checked exception
4. They are part of Object class methods.



Consumer and producer multithreading wait, notify and notifyAll example