Interface

Interface is a type of reference data type. It is also known as 100% pure abstract class.

Syntax

```
Interface interfacename {
        Property
        Behaviour
}
```

By default in interface all property are public static and final.

By default all method interface are public and abstract.

```
interface Abc {
        public static final int A=10;
        static final int A=10;
        final int A=10;
        int A=10;
        public abstract void dis1();
        abstract void dis1()
        void dis1();
}
```

Interface contains only final variable and abstract methods.

```
interface Abc {                                 super interface
        int A=10;
        void dis1();
}
interface Xyz {                                 super interface
        int B=20;
        void dis2();
}
interface Mno extends Abc,Xyz{                  multiple inheritance.
        int C=30;                               sub interface
        void dis3();
}
```

Like a class one interface can extends another interface. But interface can extends more than one interface but class can't. class always implements interface. Even class can implements more than one interface. Whichever class implements one or more than one interface that class must be provide the body for all methods belongs to interface mandatory.

```
class Demo implements Abc,Xyz {
                must be provide the body for dis1 part of Abc and dis2 part of Xyz interface.
}
```

Interface Vs Abstract class

1. Interface contains only final variable but abstract class can contain final as well as normal variable.
2. Interface contains only abstract method but abstract class can contains normal as well as abstract methods.
3. Interface doesn't contains constructor even though we can't write empty as well as parameter constructor. But abstract class can contains default constructor as well as we can write empty and parameter constructor
4. Using interface we can achieve 100% abstraction but using abstract class we can achieve partial abstraction.
   Ie interface use to provide specification and class provide implementation.

```
abstract class Bank {
        abstract void withdraw();
        abstract void deposit ();
        void checkBalance(int accno) {

        }
}
interface  Bank {
        void withdraw();
        void deposit ();
        void checkBalance(int accno);
}
```

Common between interface and abstract class.

1. We can't create the object of interface as well as abstract class.
2. Both are use to achieve abstraction.
3. Whichever class extends abstract class or implements interface that class must be provide the body for all abstract method belong to that interface or that abstract class.

Run time polymorphism using object creation.

We can create sub class object and super class/ interface reference.

Super can be normal class or abstract class or interface. Using that reference we can call only those methods which belong that interface or class or abstract class.

Package

Package is a collection of classes and interface which have same name but different purpose.

Package is like a directory or folder when two classes we need to same name but different purpose.

Package mainly divided into 2 types.

1. user defined package
2. pre defined package

education

school college

Attendance Attendance

Access specifiers : Access specifiers help us to provide the visibility or accessibility of variable, method and class within a same package as well as other package.

4 types

1. private
    a. we can use with instance variable, static variable, method ie static as well as non static method, constructor but not with class and local variable.
    b. Scope : within a same class.
2. default (nothing is known as default)
    a. we can use with all.
    b. Scope : within a same package.
3. protected
    a. we can use with instance variable, static variable, method ie static as well as non static method, constructor but not with class and local variable.
    b. Scope within a same package other package if it sub class.
4. public
    a. we can use with instance variable, static variable, method ie static as well as non static method, constructor, class but not local variable.
    b. Scope : same package as well as other package.

Pre defined package

java	javax → root directory or package
lang	sql
io	swing
util	net
sql	servlet
net	ejb
awt	jms

by default every java program imported lang package.

By default every java program extends pre defined class ie Object. Object part of lang package.

String

StringBuffer

StringBuilder

Wrapper classes

Exception and type of exception classes

Thread

Runnable interface

System classes and interfaces part of lang package.

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	

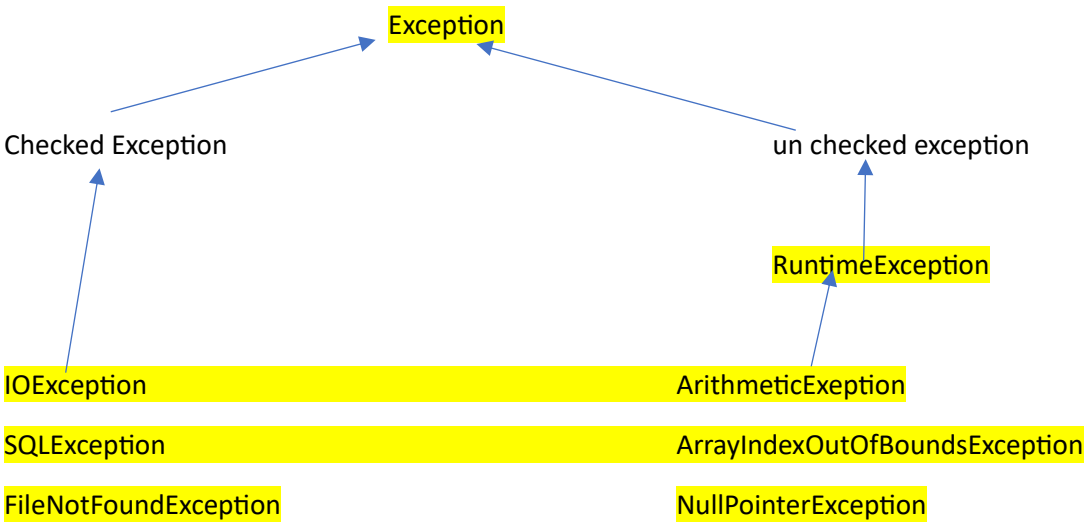
Run time error

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

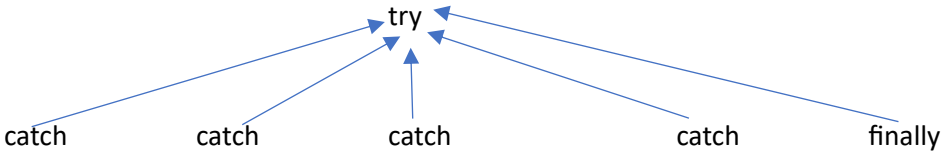
```
try {

}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
}
```