Interface: interface is keyword

Interface is not allowed constructor

Interface is class, inside the class you can write abstract method.

**IndianAirportAuthorityUtil.java**

**package** com.xworkz.xworkzapp;

**import** com.xworkz.xworkzapp.airport.KGIA;

**public** **class** IndianAirportAutorityUtil {

**public** **static** **void** main(String[] args) {

KGIA kgia = **new** KGIA("KGIA", 10, "KA01");

kgia.id();

kgia.seucrity();

}

}

**Airport.java**

**package** com.xworkz.xworkzapp.airport;

**public** **interface** Airport {

**public** **void** seucrity();

**public** **void** id();

}

**KGIA.java**

**package** com.xworkz.xworkzapp.airport;

**public** **class** KGIA **implements** Airport{

String airportName;

**int** noOfRunWays;

String id;

**public** KGIA(String airportName, **int** noOfRunWays, String id) {

**super**();

**this**.airportName = airportName;

**this**.noOfRunWays = noOfRunWays;

**this**.id = id;

System.***out***.println("Airport Name= "+airportName+" No Of RunWays= "+noOfRunWays+" id= "+id);

}

@Override

**public** **void** seucrity() {

System.***out***.println("Security checked ");

}

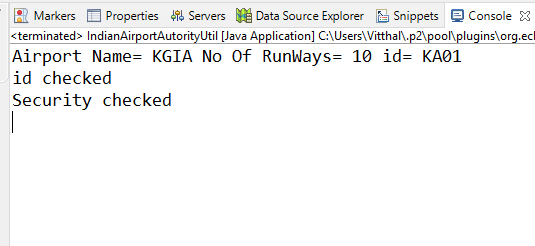
@Override

**public** **void** id() {

System.***out***.println("id checked");

}

}

**Output:** 

**Example-2**

BankOperationUtil.java

**package** com.xworkz.xworkzapp;

**import** com.xworkz.xworkzapp.bankopreation.BankOpreation;

**import** com.xworkz.xworkzapp.bankopreation.Loan;

**import** com.xworkz.xworkzapp.bankopreation.Savings;

**public** **class** BankOpreationUtil {

**public** **static** **void** main(String[] args) {

System.***out***.println("-----------Loan Account -----------");

BankOpreation loan = **new** Loan();

System.***out***.println("---------Deposit ----------");

loan.deposit(1200);

System.***out***.println("---------Withdraw ----------");

loan.withdraw(2000);

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

System.***out***.println("\*\*\*\*\*\*\*\*\*\*Saving Account \*\*\*\*\*\*\*\*\*\*\*\* ");

BankOpreation savings = **new** Savings(); //upcasting

System.***out***.println("\*\*\*\*\*\*\*\*Deposit\*\*\*\*\*\*\*\*\*\*");

savings.deposit(5000);

System.***out***.println("\*\*\*\*\*\*\*\*Withdraw\*\*\*\*\*\*\*\*\*\*");

savings.withdraw(400);

}

}

**BankOpeartion.java**

**package** com.xworkz.xworkzapp.bankopreation;

**public** **interface** BankOpreation {

**public** **int** withdraw(**int** amount);

**public** **int** deposit(**int** amount);

}

**Loan.java**

**package** com.xworkz.xworkzapp.bankopreation;

**public** **class** Loan **implements** BankOpreation {

@Override

**public** **int** withdraw(**int** amount) {

**int** wd = 10000;

System.***out***.println("Your Loan is : " + wd );

System.***out***.println("Withdraw amount : "+amount+"\nwithdraw successfully..!");

wd = wd - amount;

System.***out***.println("Balance is : " + wd);

**return** wd;

}

@Override

**public** **int** deposit(**int** amount) {

**int** dp = 300;

System.***out***.println("Old Amount is : " + dp);

System.***out***.println("Your Deposit is : " + amount);

dp = dp + amount;

System.***out***.println("Total Balance is : " + dp);

**return** dp;

}

}

**Savings.java**

**package** com.xworkz.xworkzapp.bankopreation;

**public** **class** Savings **implements** BankOpreation {

@Override

**public** **int** withdraw(**int** amount) {

**int** saving = 1000;

System.***out***.println("Savimg amount is : " + saving);

saving = saving - amount;

System.***out***.println("Withdraw Amount is " + amount + "\nwithdraw successfully...! ");

System.***out***.println("Balance amount is : " + saving);

**return** saving;

}

@Override

**public** **int** deposit(**int** amount) {

**int** saving = 1000;

System.***out***.println("Old Balance : " + saving);

System.***out***.println("Deposit amount " + amount);

saving = saving + amount;

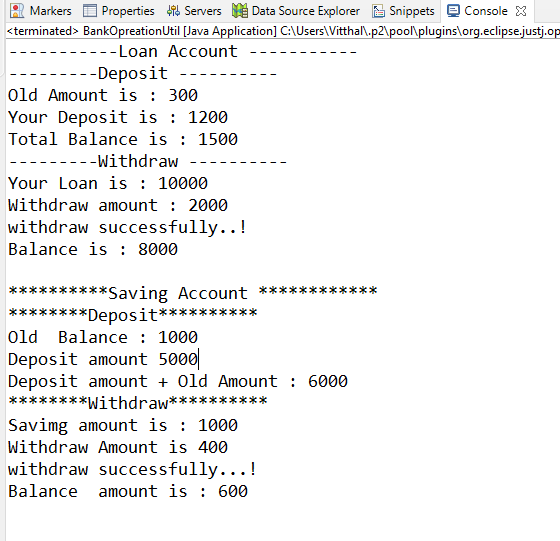
System.***out***.println("Deposit amount + Old Amount : " + saving);

**return** saving;

}

}

**Output:**



Multiple inheritance in java :

Multiple inheritance not allowed java because example assume that we have one parent class two child class, We can access parent class to using the extend keyword . than