Dt : 4/1/2025(Day-6)

=>we use getConnection()-method from 'java.sql.DriverManager' class to create implementation

object for 'Connection-Interface'.

=>This getConnection()-method internally holding 'Anonymous Local InnerClass as

implementation class of Connection-Interface' and the Connection-Object is generated.

=>This Connection-Object internally connected to Database product.

Method Signature of getConnction():

public static java.sql.Connection getConnection(java.lang.String, java.lang.String,

java.lang.String) throws java.sql.SQLException;
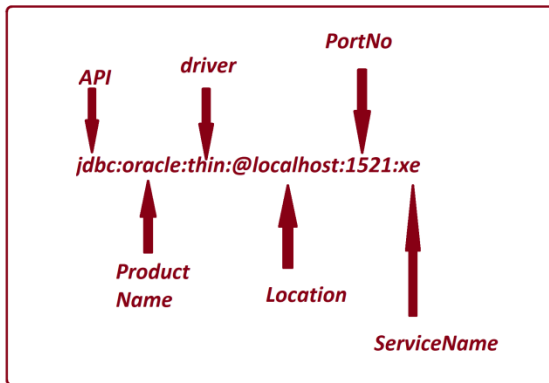
syntax:

Connection con = DriverManager.getConnction("DB-URL","DB-UName","DB-PWord");

DB-URL   => jdbc:oracle:thin:@localhost:1521:xe

DB-UName => system

DB-PWord => tiger

Diagram:

```
                        PortNo
        API   driver    ↓
        ↓     ↓
        jdbc:oracle:thin:@localhost:1521:xe
              ↑          ↑          ↑
        Product    Location
        Name
                              ServiceName
```

----

*imp

JDBC statements:

 =>The statements which specify the actions(operations) to be performed on database product

   are known as JDBC-statements:

 =>These JDBC-statements are categorized into three types:

    1.Statement

    2.PreparedStatement

    3.CallableStatement

1.Statement:

  =>'Statement' is an interface from java.sql package and which is used execute Normal

    queries without IN-Parameters.

    (Normal queries means Create,Insert,Select(retrieve),update and delete)

  =>we use createStatement()-method from 'Connection-Interface' to create implementation

    object for 'Statement-Interface'

  =>This createStatement()-method internally holding 'Anonymous Local InnerClass as

implementation class of Statement-Interface' and generates Statement-Object.

Method Signature of createStatement():

public abstract java.sql.Statement createStatement() throws java.sql.SQLException;

syntax:

Statement stm = con.createStatement();

=>The following are two important methods of 'Statement':

(a)executeQuery()

(b)executeUpdate()

(a)executeQuery():

=>executeQuery()-method is used to execute select-queries.

Method Signature:

public abstract java.sql.ResultSet executeQuery(java.lang.String)

throws java.sql.SQLException;

syntax:

ResultSet rs = stm.executeQuery("select-query");

(b)executeUpdate():

=>executeUpdate()-method is used to execute NonSelect-queries.

Method Signature:

public abstract int executeUpdate(java.lang.String) throws java.sql.SQLException;

syntax:

int k = stm.executeUpdate("NonSelect-query");

--

java.sql

DriverManager(c) ← **Normal SubClass**

Connection-Implementation-Object

static Connection getConnection(String,String,String)

{

Connection ob = new Connection()

{

createStatement(){}
prepareStatement(){}
prepareCall(){}

};

}

java.sql

Normal Interface → **Connection(I)**

All abstract methods →
createStatement()
prepareStatement()
prepareCall()
..
...

Anonymous Local InnerClass as Implementation class of Connection Interface

implemented methods

createStatement(){
}
prepareStatement(){
}
prepareCall(){
}

Connection con = DriverManager.getConnection("","","");

0x2

Database (Oracle)

java.sql

Normal Interface → **Statement(I)**

All abstract methods →
executeQuery(..)
executeUpdate(..)
....

0x2

executeQuery(...){}
executeUpdate(...){}

0x3

Statement stm = con.createStatement();

0x2