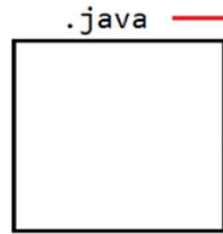


JAR -> Java Archive File
Collection of .class files

SUNMS -> 1st Party
JavaDev -> 2nd Party
DBVend -> 3rd Party

Write Once



Classes

1. DriverManager
2. Date
3. Time
4. TimeStamp
5. Types

path

rt.jar

***.jar
classpath

jdk=>rt.jar(java.sql)

JDBC API

(100%abstract)
Interfaces

1. Driver
2. Connection
3. Statement
4. PreparedStatement
5. CallableStatement
6. ResultSet
7. ResultSetMetaData
8. DatabaseMetaData

jar

ojdbc6.jar

Oracle

Run

jar

mysql-conn
.jar

MySQL

Run

jar

postgresql
.jar

PostgreSQL

Run

```
Command Prompt
Microsoft Windows [Version 10.0.22621.819]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nitin>set path=C:\Program Files\Java\jdk1.8.0_202\bin;
C:\Users\nitin>set classpath=D:\jars\mysql-connector-j-8.0.31.jar
C:\Users\nitin>D:|
```

```
D:\>echo %path%
C:\Program Files\Java\jdk1.8.0_202\bin;
```

```
D:\>echo %classpath%
D:\jars\mysql-connector-j-8.0.31.jar
```

```
D:\>javap com.mysql.cj.jdbc.Driver
Compiled from "Driver.java"
public class com.mysql.cj.jdbc.Driver extends com.mysql.cj.jdbc.NonRegisteringDriver implements java.sql.Driver {
    public com.mysql.cj.jdbc.Driver() throws java.sql.SQLException;
    static {};
}

D:\>|
```

```

D:\>javap java.sql.DriverManager
Compiled from "DriverManager.java"
public class java.sql.DriverManager {
    static final java.sql.SQLPermission SET_LOG_PERMISSION;
    static final java.sql.SQLPermission DEREGISTER_DRIVER_PERMISSION;
    public static java.io.PrintWriter getLogWriter();
    public static void setLogWriter(java.io.PrintWriter);
    public static java.sql.Connection getConnection(java.lang.String, java.util.Properties) throws java.sql.SQLException;
    public static java.sql.Connection getConnection(java.lang.String, java.lang.String, java.lang.String) throws java.sql.SQLException;
    public static java.sql.Connection getConnection(java.lang.String) throws java.sql.SQLException;
    public static java.sql.Driver getDriver(java.lang.String) throws java.sql.SQLException;
    public static synchronized void registerDriver(java.sql.Driver) throws java.sql.SQLException;
    public static synchronized void registerDriver(java.sql.Driver, java.sql.DriverAction) throws java.sql.SQLException;
    public static synchronized void deregisterDriver(java.sql.Driver) throws java.sql.SQLException;
    public static java.util.Enumeration<java.sql.Driver> getDrivers();
    public static void setLoginTimeout(int);
    public static int getLoginTimeout();
    public static void setLogStream(java.io.PrintStream);
    public static java.io.PrintStream getLogStream();
    public static void println(java.lang.String);
    static {};
}

```

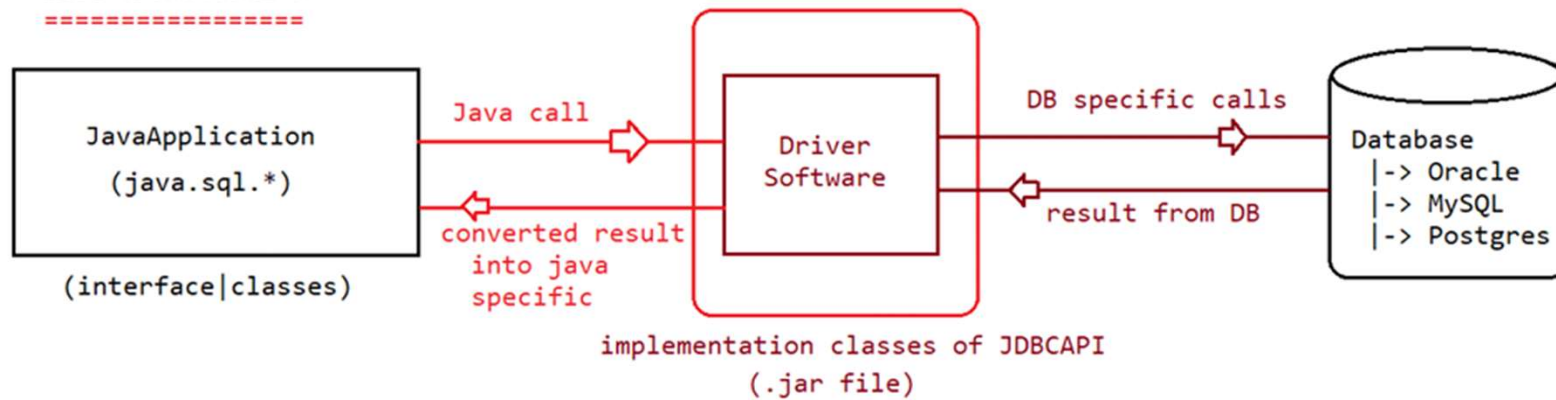
D:\>

```

D:\JDBCPGMS>javap java.sql.Connection
Compiled from "Connection.java"
public interface java.sql.Connection extends java.sql.Wrapper, java.lang.AutoCloseable {
    public static final int TRANSACTION_NONE;
    public static final int TRANSACTION_READ_UNCOMMITTED;
    public static final int TRANSACTION_READ_COMMITTED;
    public static final int TRANSACTION_REPEATABLE_READ;
    public static final int TRANSACTION_SERIALIZABLE;
    public abstract java.sql.Statement createStatement() throws java.sql.SQLException;
    public abstract java.sql.PreparedStatement prepareStatement(java.lang.String) throws java.sql.SQLException;
    public abstract java.sql.CallableStatement prepareCall(java.lang.String) throws java.sql.SQLException;
    public abstract java.lang.String nativeSQL(java.lang.String) throws java.sql.SQLException;
    public abstract void setAutoCommit(boolean) throws java.sql.SQLException;
    public abstract boolean getAutoCommit() throws java.sql.SQLException;
    public abstract void commit() throws java.sql.SQLException;
    public abstract void rollback() throws java.sql.SQLException;
    public abstract void close() throws java.sql.SQLException;
    public abstract boolean isClosed() throws java.sql.SQLException;
    public abstract java.sql.DatabaseMetaData getMetaData() throws java.sql.SQLException;
    public abstract void setReadOnly(boolean) throws java.sql.SQLException;
    public abstract boolean isReadOnly() throws java.sql.SQLException;
    public abstract void setCatalog(java.lang.String) throws java.sql.SQLException;
    public abstract java.lang.String getCatalog() throws java.sql.SQLException;
    public abstract void setTransactionIsolation(int) throws java.sql.SQLException;
    public abstract int getTransactionIsolation() throws java.sql.SQLException;
    public abstract java.sql.SQLWarning getWarnings() throws java.sql.SQLException;
    public abstract void clearWarnings() throws java.sql.SQLException;
}

```

JDBC Architecture

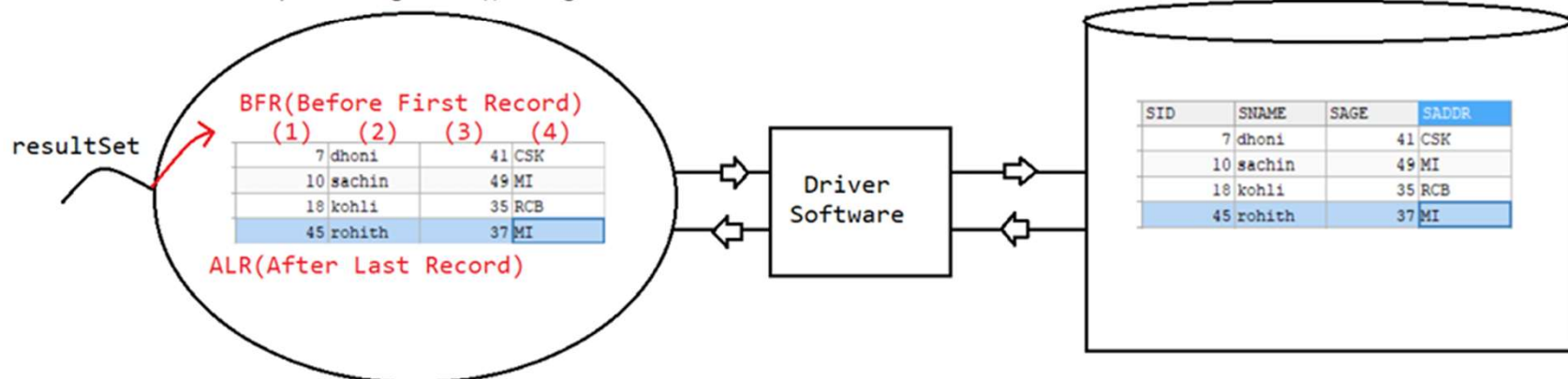


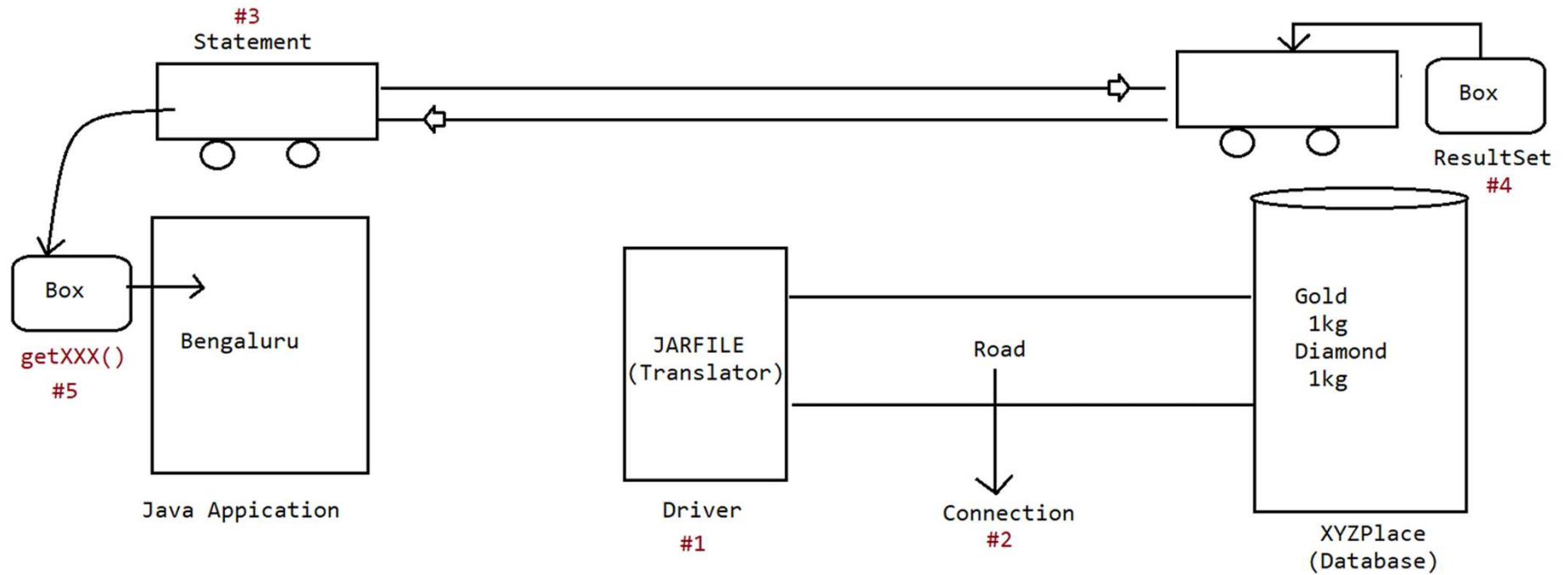
DriverSoftware

1. Type-1 driver
2. Type-2 driver
3. Type-3 driver
- ☆ 4. Type-4 driver

Enapsulation

|-> use getXXX() to get the data





Path : It is an environmental variable associated with Operating System.
These environmental variables are used to inform the location of .exe files
of the softwares which needs to run from command prompt.

```
set path =...../bin
|=> .exe files
```

ClassPath : It is an environmental variables associated with "Java Programs".
Java developer uses this environmental variable and inform the jdk s/w
to search for the required .class files(normally used in jdbc,servlet,hibernate,springmvc,..)

```
set classpath=****.jar
|=> .class files
```