|  |
| --- |
| **7. What is the purposes of JDBC?**  1. **Provide Existing Enterprise Data** Businesses can continue to use their installed databases and access information even if it is stored on different database management systems.  2. **Simplified Enterprise Development** The combination of the Java API and the JDBC API makes application development easy and cost effective.  3. **Zero Configuration for Network Computers** No configuration is required on the client side centralizes software maintenance. Driver is written in the Java, so all the information needed to make a connection is completely defined by the JDBC URL or by a DataSource object. DataSource object is registered with a Java Naming and Directory Interface (JNDI) naming service.  4. **Full Access to Metadata**  The underlying facilities and capabilities of a specific database connection need to be understood. The JDBC API provides metadata access that enables the development of sophisticated applications.   5. **No Installation** A pure JDBC technology-based driver does not require special installation.  6. **Database Connection Identified by URL** The JDBC API includes a way to identify and connect to a data source, using a DataSource object. This makes code even more portable and easier to maintain. |
|  |

* [Next Page »](https://www.careerride.com/JDBC-Class-forName.aspx)

[**What will Class.forName do while loading drivers?**](https://www.careerride.com/JDBC-Class-forName.aspx)

JDBC drivers - When you have loaded a driver, it is available for making a connection with a DBMS...

[**What are the different types of driver?**](https://www.careerride.com/JDBC-driver-types.aspx)

JDBC Driver Types - JDBC Net pure Java driver(Type IV) is the fastest driver because it converts the jdbc calls to network protocol used by DBMS Server...

[**What are the standard isolation levels defined by JDBC?**](https://www.careerride.com/JDBC-isolation-levels.aspx)

JDBC Isolation Levels - The values are defined in the class java.sql.Connection and are: ...

**1. Draw phases of water fall life cycle.**

## What is WaterFall Model?

The ***Waterfall Model*** was first Process Model to be introduced. It is very simple to understand and use. In a Waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. Waterfall model is the earliest [***SDLC***](http://toolsqa.com/software-testing/software-development-life-cycle/) approach that was used for software development.

In “***The Waterfall***” approach, the whole process of software development is divided into separate phases. The outcome of one phase acts as the input for the next phase sequentially. This means that any phase in the development process begins only if the previous phase is complete. The waterfall model is a sequential design process in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of ***Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance.***

As the ***Waterfall Model*** illustrates the software development process in a linear sequential flow; hence it is also referred to as a ***Linear-Sequential Life Cycle Model***.

### *WaterFall Model*

### ***Sequential Phases in Waterfall Model***

* ***Requirements:***The first phase involves understanding what need to be design and what is its function, purpose etc. Here, the specifications of the input and output or the final product are studied and marked.
* ***System Design:*** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The software code to be written in the next stage is created now.
* ***Implementation:*** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
* ***Integration and Testing:*** All the units developed in the implementation phase are integrated into a system after testing of each unit. The software designed, needs to go through constant software testing to find out if there are any flaw or errors. Testing is done so that the client does not face any problem during the installation of the software.
* ***Deployment of System:*** Once the functional and non-functio

nal testing is done, the product is deployed in the customer environment or released into the market.

* ***Maintenance:*** This step occurs after installation, and involves making modifications to the system or an individual component to alter attributes or improve performance.These modifications arise either due to change requests initiated by the customer, or defects uncovered during live use of the system. Client is provided with regular maintenance and support for the developed software.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name “Waterfall Model“.

### ***Advantages of Waterfall Model***

* The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.
* The waterfall model progresses through easily understandable and explainable phases and thus it is easy to use.
* It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
* In this model, phases are processed and completed one at a time and they do not overlap. Waterfall model works well for smaller projects where requirements are very well understood.

### ***Disadvantages of Waterfall Model***

* It is difficult to estimate time and cost for each phase of the development process.
* Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
* Not a good model for complex and object-oriented projects.
* Not suitable for the projects where requirements are at a moderate to high risk of changing.

***For more updates on Software Testing, please*** [***Subscribe***](https://feedburner.google.com/fb/a/mailverify?uri=ToolsQA)***to our Newsletter.***

***Please ask any questions on [ForumsQA](http://www.forumsqa.com/" \o "Question & Answer Forum for Selenium WebDriver" \t "_blank), in case of any issues or doubts.***

**2. What is the function of boundary class?**

Boundary Interpolation Problem in the Classesof Generalized Nevanlinna Matrix FunctionsA.

A. AmirshadyanReceived April 1, 2002Abstract—The paper deals with the boundary indefinite interpolation problem in the classesof generalized Nevanlinna matrix functions. A one-to-one correspondence between the set ofall solutions of the problem and the class of so-calledG-regular self-adjoint extensions of themodel symmetric operator associated with the problem is established. Sufficient conditions fortheG-regularity of self-adjoint extensions (in terms of the Weyl function) are given. A formulafor the description of all the solutions of the problem is obtained.Key words:symmetric operator, generalized resolvent, Weyl function, boundary interpolation,Pick matrix, Nevanlinna pair.The scalar Nevanlinna–Pick boundary-value problem was studied in [1] for the class of contract-ing functions and in [2 ,3] for the Nevanlinna and Stieltjes classes. The matrix boundary-valueproblem with multiple interpolation points was considered in [4]. In the present paper ,within theframework of the operator approach ,we study the boundary indefinite interpolation problem in theclassesNκ(Cn) of generalized Nevanlinna matrix functions. A one-to-one correspondence betweenthe set of all solutions of the problem and the class of so-calledG-regular self-adjoint extensionsof the model symmetric operator associated with the problem is established. Sufficient conditionsfor theG-regularity of self-adjoint extensions (in terms of the Weyl function) are given. Anotherapproach to this problem related to the indefinite version of the Berling–Lax theorem was usedin [5].Definition 1.The pair ofn×nmatrix functions{φ(λ),ψ(λ)}holomorphic in the domainO=O⊂C\Ris called ageneralized Nevanlinna pair(orNκ-pair,κ∈Z+)if(1) the kernelNφψ(λ,μ)=(φ(μ)∗ψ(λ)−ψ(μ)∗φ(λ))/(λ− ̄μ)hasκnegative squares inO;(2)ψ( ̄λ)∗φ(λ)−φ( ̄λ)∗ψ(λ) = 0 for allλ∈O;(3) rank{φ(λ)∗:ψ(λ)∗}=nfor allλ∈O.The two pairs{φ,ψ}and{φ1,ψ1}are said to beequivalentifφ1(λ)=φ(λ)H(λ),ψ1(λ)=ψ(λ)H(λ) for some holomorphic and invertible (inO) matrix functionH(λ). By ̃Nκ(Cn)wede-note the set of equivalentNκ-pairs. Identifying the matrix functionF(λ) with the pair{I,F(λ)},we regard the classNκ(Cn) of matrix functions as a subset of ̃Nκ(Cn) . Let us recall that by aPon-tryagin spacewe mean a Hilbert space Π endowed with an indefinite inner product [·,·]=(J·,·),whereJ=J∗=J−1is a linear operator in Π with a finite negative index sq−Π=dim(I−J)Π .Suppose thatSis a nondensely defined closed symmetric linear operator in the Pontryagin space(Π,[·,·]) with a nonempty set ˆρ(S) of points of regular type such that the defect subspacesNλ=ker(S+−λ),λ∈ˆρ(S) ,are finite-dimensional and the deficiency indices aren+(S)=n−(S)(= dimNλ),λ∈C±∩ˆρ(S). HereS+is the linear relation in Π adjoint toS.0001-4346/2003/7312-0163$25.00c©2003 Plenum Publishing Corporation163

**Boundary Functions**  
T.J. Kaczynski  
Doctoral dissertation abstract, University of Michigan

Let H denote the set of all points in the Euclidean plane having positive y-coordinate, and let X denote the x-axis. If p is a point of X, then by an arc at p we mean a simple arc v, having one endpoint at p, such that v - {p} ( H. Let f be a function mapping H into the Riemann sphere. By a boundary function for f we mean a function t defined on a set E ( X such that for each p ( E there exists an arc v at p for which

lim f(z) = t(p).

z -> p

z ( v

The set of curvilinear convergence of f is the largest set on which a boundary function for f can be defined; in other words, it is the set of all points p ( X such that there exists an arc at p along which f approaches a limit. A theorem of J.E. McMillan states that if f is a continuous function mapping H into the Riemann sphere, then the set of curvilinear convergence of F is of type F(sd). In the first of two chapters of this dissertation we give a more direct proof of this result than McMillan's, and we prove, conversely, that if A is a set of type F(sd) in X, then there exists a bounded continuous complex-valued function in H having A as its set of curvilinear convergence. Next, we prove that a boundary function for a continuous function can always be made into a function of Baire class 1 by changing its values on a countable set of points. Conversely, we show that if t is a function mapping a set E ( X into the Riemann sphere, and if t can be made into a function of Baire class 1 by changing its values on a countable set, then there exists a continuous function in H having t as a boundary function. (This is a slight generalization of a theorem of Bagemihl and Piranian.) In the second chapter we prove that a boundary function for a function of Baire class e > 1 in H is of Baire class at most e + 1. It follows from this that a boundary function for a Borel-measurable function is always Borel-measurable, but we show that a boundary function for a Lebesgue-measurable function need not be Lebesgue-measurable. The dissertation concludes with a list of problems remaining to be solved.

**10. What is the function of executeQuery()?**

# The executeQuery( ) Method

Now that you’ve learned how to insert, update, and delete data in a table, it’s time to learn how to use a SELECT statement to retrieve data. Whereas the execute( ) and executeUpdate( ) methods discussed in previous sections return primitive data types -- a boolean and int, respectively -- the method normally used with a SELECT statement, executeQuery( ), returns a ResultSet object. The executeQuery( ) method effectively combines the execute( ) and getResultSet( ) methods into one call:

ResultSet rset = null;

Statement stmt = null;

try {

stmt = conn.createStatement( );

rset = stmt.executeQuery("select last\_name, first\_name from person");

. . .

}

Difference between execute, executeQuery and executeUpdate in JDBC.

**execute** method can be used with any type of SQL statements and it returns a boolean. A true indicates that the execute method returned a result set object which can be retrieved using getResultSet method. false indicates that the query returned an int value or void. execute method can run both select and insert/update statements.

**executeQuery** method execute statements that returns a result set by fetching some data from the database. It executes only select statements.

**executeUpdate** method execute sql statements that insert/update/delete data at the database. This method return int value representing number of records affected; Returns 0 if the query returns nothing. The method accepts only non-select statements.