# Thread

# Spring

1. What is Spring IOC?

Check spring document

10.Do you know AspectJ

Using ordinary Java, it can be difficult to modularize design concerns such as

* system-wide error-checking strategies
* design patterns
* synchronization policies
* resource sharing
* distribution concerns
* performance optimizations

AspectJ is a simple and practical extension to the Java programming language that adds to Java aspect-oriented programming (AOP) capabilities. AOP allows developers to reap the benefits of modularity for concerns that cut across the natural units of modularity. In object-oriented programs like Java, the natural unit of modularity is the class. In AspectJ, aspects are concerns that affect more than one class.

More information can be found at:  
[**http://aspectj.org/**](http://aspectj.org/)

11. What is Spring AOP

12.In spring have you used Annotations? or XMLs?

13.If Error is there in config file, when will it show the error is it run time or compile time?

20. When spring beans are initiated? What exceptions are thrown when class is missing?

22. What components are present in spring.xml?

23. Can a bean mapped to an interface?

25. What module in spring you have used?

47. What is dependency injection?

48. What are the different types of Dis in spring?

53. Common exceptions in Spring?

Tell me about bean life cycle and why we need so many stages?

1. How transactions are handled in spring?
2. How spring comes to know about the spring config xml?
3. What happens before ApplicationContext is created?
4. 46. What are the scopes available in spring?

# Collections

How you write a LinkedList program?

<http://crunchify.com/how-to-implement-a-linkedlist-class-from-scratch-in-java/>

How does get(Key key) method works internally in HashMap, and Hashtable in Java?  
  
<http://java67.blogspot.com/2013/06/how-get-method-of-hashmap-or-hashtable-works-internally.html#ixzz45ioG9Rsd>

<https://tekmarathon.com/2013/03/11/creating-our-own-hashmap-in-java/>

1. What is blocked queue in jdk 5?
2. How put works in hashmap?
3. What is concurenthashmap?

<http://crunchify.com/hashmap-vs-concurrenthashmap-vs-synchronizedmap-how-a-hashmap-can-be-synchronized-in-java/>

**Map<String,String> myMap = new ConcurrentHashMap<String,String>();**

creates map with Creates a new, empty map with a default initial capacity (16), load factor (0.75) and concurrencyLevel (16).

initialCapacity - the initial capacity. The implementation performs internal sizing to accommodate this many elements.

concurrencyLevel - the estimated number of concurrently updating threads. The implementation performs internal sizing to try to accommodate this many threads.

static final int DEFAULT\_INITIAL\_CAPACITY = 16;

static final int DEFAULT\_CONCURRENCY\_LEVEL = 16;

initial capacity parameter and concurrency level parameters of ConcurrentHashMap constructor (or Object) are  set to 16 by default.  
  
Thus, instead of a map wide lock, ConcurrentHashMap maintains  a list of 16 locks by default ( number of locks equal to the initial capacity , which is by default  16) each of which is used to lock on a single bucket of the Map.This indicates that 16 threads (number of threads equal to the concurrency level , which is by  default 16) can modify the collection at the same time , given ,each thread works on different bucket. So unlike hashtable, we perform any sort of operation ( update ,delete ,read ,create) without locking on entire map in ConcurrentHashMap.

1. How to write own concurrenthash map?
2. How ConcurrentHashMap Internally Works In Java With Example?

<http://javahungry.blogspot.com/2015/02/how-concurrenthashmap-works-in-java-internal-implementation.html>

1. What are copytowrite\* collection class?
2. What is ConcurrentModificationException?
3. How can you write a code which will create ConcurrentModificationException?

<https://examples.javacodegeeks.com/java-basics/exceptions/java-util-concurrentmodificationexception-how-to-handle-concurrent-modification-exception/>

<http://www.journaldev.com/378/java-util-concurrentmodificationexception>

<http://www.ibm.com/developerworks/library/j-jtp08223/>

1. What is difference between collection.remove() and iteration.remove()?
2. What are differences between hashmap, concurrenthashmap and synchronizedhashmap?
3. How concurrent collections works? How to write own concurrent collections?
4. What is difference between Fail Fast Vs Fail Safe Iterator In Java?

<http://javahungry.blogspot.com/2014/04/fail-fast-iterator-vs-fail-safe-iterator-difference-with-example-in-java.html>

final Entry<K,V> More ...nextEntry() {

if (modCount != expectedModCount)

throw new ConcurrentModificationException();

|  |  |  |
| --- | --- | --- |
|  | **Fail Fast Iterator** | **Fail Safe Iterator** |
| Throw ConcurrentModification Exception | Yes | No |
| Clone object | No | Yes |
| Memory Overhead | No | Yes |
| Examples | HashMap,Vector,ArrayList,HashSet | CopyOnWriteArrayList, ConcurrentHashMap |

J2EE

1. How to implement Application security

<https://docs.oracle.com/javaee/7/tutorial/security-webtier002.htm#BNCBM>

4. Give us some example on prototype scope, when can we use that?

<http://www.mkyong.com/spring/spring-bean-scopes-examples/>

13.What all IDEs have you worked on?

14.How will you deploy an application?

15.Where is the installed application visible?

F:\apache tomcat\apache-tomcat-7.0.55\webapps

16. Do we need to Start the server to view changes of class?

Yes

17. Do we need to start the server to view changes of jsp?

Because when Tomcat is asked to execute a JSP, is compares the modification date of the JSP file with the modification time of the compiled class corresponding to this JSP, and if more recent, it recompiles on the fly before executing it.

This is BTW an option that should be turned off in production, because it takes time to perform this check.

Design Pattern

3. Have you worked on Design patterns? if so, how did you implement?

**30. Why we need singleton class?**

**31. How we can stop multiple thread manipulating singleton state?**

40. Why we use factory pattern?

41. How you write singleton class?

39. What design patterns I worked on?

1. What is singleton?
2. How you write singleton?
3. How are different ways of creating singleton class?
4. How a singleton class cannot be singleton?

<http://www.journaldev.com/1377/java-singleton-design-pattern-best-practices-with-examples>

1. Write your own decorative pattern?
2. What is scope of singleton?
3. How can you break singleton?

# GC

1. **What thread is Garbage collection?**
2. Daemon
3. **What is daemon thread?**
4. Daemon thread is a low priority thread, which runs intermittently in the back ground doing the garbage collection operation for the java runtime system.
5. **Which method is used to create the daemon thread?**
6. setDaemon method is used to create a daemon thread.

* You can make any java thread as daemon thread. Daemon threads acts like service providers for other threads running in the same process.
* Daemon threads will be terminated by the JVM when there are none of the other threads running, it includs main thread of execution as well.
* To specify that a thread is a daemon thread, call the setDaemon method with the argument true.
* To determine if a thread is a daemon thread, use the accessor method isDaemon.

package com.myjava.threads;

public class DaemonThread extends Thread{

    public DaemonThread(){

        setDaemon(true);

    }

    public void run(){

        System.out.println("Is this thread Daemon? - "+isDaemon());

    }

    public static void main(String a[]){

        DaemonThread dt = new DaemonThread();

        // even you can set daemon constrain here also

        // it is like dt.setDeamon(true)

        dt.start();

    }

}

<http://www.java2novice.com/java_thread_examples/daemon_threads/#sthash.gwM2ZwYK.dpuf>

1. **What is memory leak? Why it happens?**

There actually four categories of memory issues with similar and overlapping symptoms, but varied causes and solutions:

* **Performance**: usually associated with excessive object creation and deletion, long delays in garbage collection, excessive operating system page swapping, and more.
* **Resource constraints**: occurs when there’s either to little memory available or your memory is too[fragmented](https://en.wikipedia.org/wiki/Fragmentation_(computing)) to allocate a large object—this can be native or, more commonly, Java heap-related.
* **Java heap leaks**: the classic memory leak, in which Java objects are continuously created without being released. This is usually caused by latent object references.
* **Native memory leaks**: associated with any continuously growing memory utilization that is outside the Java heap, such as allocations made by JNI code, drivers or even JVM allocations.

<https://www.toptal.com/java/hunting-memory-leaks-in-java>

<https://dzone.com/articles/what-memory-leak-java>

<http://coderevisited.com/memory-leaks-in-java/>

<http://www.pointsoftware.ch/en/under-the-hood-runtime-data-areas-javas-memory-model/>

<http://cmk-java.blogspot.in/2014/08/java-virtual-machine-internals.html>

<http://blog.ragozin.info/2011/06/understanding-gc-pauses-in-jvm-hotspots.html>

<https://plumbr.eu/blog/garbage-collection/minor-gc-vs-major-gc-vs-full-gc>

<http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/gc01/index.html>

<https://plumbr.eu/handbook/garbage-collection-algorithms>

<http://www.ibm.com/developerworks/library/j-jtp11253/>

1. **What are GC algorithms?**

<https://plumbr.eu/handbook/garbage-collection-algorithms>

<https://www.linkedin.com/pulse/java-memory-architecture-garbage-collection-issues-kumar-puri>

1. **What are strong reference and weak reference?**

<https://community.oracle.com/blogs/enicholas/2006/05/04/understanding-weak-references>

How I check GC logs in WAS server?

1. How to analyze outofmemory exception?
2. How to know the code causing outofmemory exception?
3. How mark and sweep works?
4. How to code memory leak?

OOPs

WebServices

6. Have you used SOAP UI if so, how to use that?

8. Tell us about xml parsing and types of XML parsing which you used

<http://www.tutorialspoint.com/java_xml/java_xml_parsers.htm>

Struts

7. Explain struts validation

Check struts document

SQL

9. Have you worked on SQL queries?

<http://www.w3schools.com/sql/>

Write Code

27. Write code to reverse a sentence?

28. Write code to reverse each word in a sentence?

29. Write code to count occurrence of a string in a sentence?

12.Have you worked on batch scripts?

18. what is war file and ear file?

19. Have you worked on Hibernate?

21. How maven is configured in eclipse? How eclipse knows about .m2?

24. How will you know if one node is up and another one is down in a cluster?

26. How to know if application is correctly installed?

34. Who and how it runs?

35. What is package? Why we need package?

36. What is checked exception?

37. How runtime is not checked exception even though it extends exception?

38. How to code own exception?

42. What are things to keep in mind for performance improvement?

44. What is marker interface?

45. What methods are in Serializable interface?

49. How you can eliminate junk characters from incoming JSON before it is converted into object?

50. What happens when we setup jdbc connection?

51. How connection pools are managed and how much memory is allocated to it?

52. How filters work?

**Persistent Systems**

How to find out null pointer exceptions? What tools to use?

What is immutable class? What are things to take care to make a class immutable?

<http://javarevisited.blogspot.in/2013/03/how-to-create-immutable-class-object-java-example-tutorial.html>

How to make a class for which only 3 instances are allowed?

What is difference between SOAP and RESTful?

**MYSIS round -2**

What are the top 5 things I check during code review?

To develop a new application how will you decide how many classes required?

Why we need multiple classes? Why can’t we put all code in same class under different methods?

Why we need interface? Why can’t we use multiple classes?

Why handle exceptions with try catch instead of declare it and let JVM handle it?

How I check performance of the application?

What is normalization? Why we need multiple tables? Why can’t we use single table?

<http://www.studytonight.com/dbms/database-normalization.php>

How to write an immutable class?

Now show how can you create multiple objects?

What will happen if you assign new object to old ref variable?

When newly created object is lost how can we get it back?

What is encapsulation?

If we do not put setter and getter then is code not encapsulated?

What is abstraction?

Is only abstract class is abstraction?

SAP interview

1. What is classloader?
2. How classloader works?
3. How many types of classloader?
4. Give some design patterns in JDK?

<http://stackoverflow.com/questions/1673841/examples-of-gof-design-patterns-in-javas-core-libraries>

1. Can we write class inside interface?

<http://way2java.com/java-lang/inner-classes-java-lang/class-inside-interface/>

<http://stackoverflow.com/questions/10553475/what-is-the-use-of-writing-a-class-inside-an-interface>

* Nested interface must be public if it is declared inside the interface but it can have any access modifier if declared within the class.
* Nested interfaces are declared static implicitely.

public interface Employee {

class Role{

public String rollname;

public int roleId;

public Object person;

}

Role getRole();

}

The class Role is used by the employee interface in getRole() method.

The designer of the interface decided that this class is so tightly coupled with the interface that it is worth to define it inside that interface to emphasize how important that class is for the interface.

Also it provides semantic namespace for the class: employee.Role.

It limits the scope of the class to where it belongs.

1. What is index in DB?
2. How it is internally works?

<http://www.programmerinterview.com/index.php/database-sql/what-is-an-index/>

1. How to select top 10th salary in table?

SELECT   empno,

  2           ename,

  3           sal,

  4           deptno

  5    FROM   (SELECT   e1.\*, DENSE\_RANK () OVER (ORDER BY sal DESC) rnk

  6              FROM   emp e1)

  7   WHERE   rnk = 3

1. How deadlock happens?
2. How to get rid of deadlock?

<http://www.javacreed.com/what-is-deadlock-and-how-to-prevent-it/>

1. Is servlet multithreaded?
2. How then objects are shared between threads?
3. What difference is between create statement and prepare statement?

<http://www.journaldev.com/2489/jdbc-statement-vs-preparedstatement-sql-injection-example>

1. How consumer producer works?
2. How many types of scans avaialable for DB?

Netcracker:

1. How to write factory pattern?
2. What is difference between String s = “abc”; and String s = new String(“abc”)?
3. What is immutable?
4. How to write Immutable class with date?

<http://javarevisited.blogspot.in/2013/03/how-to-create-immutable-class-object-java-example-tutorial.html>

<http://howtodoinjava.com/core-java/related-concepts/how-to-make-a-java-class-immutable/>

1. Where does string objects and sting values reside?
2. What is enum?
3. What is difference between enum and enumeration?
4. /\*\* Preferred style for singletons. \*/
5. **public** **enum** SantaClaus {
6. INSTANCE;
8. /\*\*Add some behavior to the object. \*/
9. **public** **void** distributePresents(){
10. //elided
11. }
13. /\*\* Demonstrate use of SantaClaus. \*/
14. **public** **static** **void** main(String... aArgs){
15. SantaClaus fatGuy = SantaClaus.INSTANCE;
16. fatGuy.distributePresents();
18. //doesn't compile :
19. //SantaClaus fatGuy = new SantaClaus();
20. }
21. }
22. What is scope of singleton?

Every singleton class is per JVM

1. How to create singleton java class for multiple jvm support?

<http://www.techspot.co.in/2009/07/singleton-in-cluster.html>

1. What is readresolve?

3.7 The readResolve Method

For Serializable and Externalizable classes, the readResolve method allows a class to replace/resolve the object read from the stream before it is returned to the caller. By implementing the readResolve method, a class can directly control the types and instances of its own instances being deserialized. The method is defined as follows:

ANY-ACCESS-MODIFIER Object readResolve()

throws ObjectStreamException;

The readResolve method is called when ObjectInputStream has read an object from the stream and is preparing to return it to the caller. ObjectInputStream checks whether the class of the object defines the readResolve method. If the method is defined, the readResolve method is called to allow the object in the stream to designate the object to be returned. The object returned should be of a type that is compatible with all uses. If it is not compatible, a ClassCastException will be thrown when the type mismatch is discovered.

For example, a Symbol class could be created for which only a single instance of each symbol binding existed within a virtual machine. The readResolve method would be implemented to determine if that symbol was already defined and substitute the preexisting equivalent Symbol object to maintain the identity constraint. In this way the uniqueness of Symbol objects can be maintained across serialization.

Note - The readResolve method is not invoked on the object until the object is fully constructed, so any references to this object in its object graph will not be updated to the new object nominated by readResolve. However, during the serialization of an object with the writeReplace method, all references to the original object in the replacement object's object graph are replaced with references to the replacement object. Therefore in cases where an object being serialized nominates a replacement object whose object graph has a reference to the original object, deserialization will result in an incorrect graph of objects. Furthermore, if the reference types of the object being read (nominated by writeReplace) and the original object are not compatible, the construction of the object graph will raise a ClassCastException.

1. How can you break singleton?

<https://www.securecoding.cert.org/confluence/display/java/MSC07-J.+Prevent+multiple+instantiations+of+singleton+objects>

1. How server sets up session?
2. How to save session?
3. What is hierarchical query?
4. How to use hints in sql?
5. Write select query to get employee\_name and manager\_name?

|  |  |  |
| --- | --- | --- |
| emp\_id | emp\_name | manager\_emp\_id |
| 1 | Ram | 3 |
| 2 | Shyam | 3 |
| 3 | Ravi | 7 |
| 4 | Pritam | 7 |
| 5 | Laxman | 4 |
| 6 | Pradeep | 4 |
| 7 | Sunil | 7 |
| 8 | Raghu | 3 |

1. Write a program consists of two threads printing as mentioned below.

1-t1

2-t2

3-t1

4-t2

5-t1

1. What is explain?
2. How to run explain in dev?
3. How you analyze explain?
4. Write a program which will except any type of collection and return back sorted same collection.
5. How serialization breaks singleton?

<http://howtodoinjava.com/design-patterns/creational/singleton-design-pattern-in-java/>

public class DemoSingleton implements Serializable {

    private volatile static DemoSingleton instance = null;

    public static DemoSingleton getInstance() {

        if (instance == null) {

            instance = new DemoSingleton();

        }

        return instance;

    }

    protected Object readResolve() {

        return instance;

    }

    private int i = 10;

    public int getI() {

        return i;

    }

    public void setI(int i) {

        this.i = i;

    }

}

Oracle

What will happen if we call static method in child with parent reference. Both Parent and Child have same static method.

What is connection pool?

<http://docs.oracle.com/javase/jndi/tutorial/ldap/connect/pool.html>

How to sort a collection?

How to improve performance of a program reading huge file with multithread?

How to reverse a string?

How to reverse string without using array?

How to reverse each character of string?

How to find anagaram in a string?

Write a method that will take decimal and base and convert to hex if base is hex and to binary if base is binary.

How to find if a duplicate character in a string?

How to find out if a node in linked list is looping back?

Do you know about lock class in java?

How to implement own Hashset?

How locking is different in hashtable, synchronized hashmap and concurrenthashmap?

Does copy of map happens in concurrent hashmap?

How to reverse a stack?

How to schedule threads?

How to lock an object to be updated from different threads?

How you pass from one servlet to another one?

In your servlet1.

request.setAttribute("attributeName",yourStringVAlue);

RequestDispatcher rd = request.getRequestDispatcher("yourServletPattern");

rd.forward(request,response);

In your Servlet2

String someName = (String)request.getAttribute("attributeName");

What is actionerror?

<http://www.mkyong.com/struts2/struts-2-actionerror-actionmessage-example/>

What is actionmap?

What is actionform?

<https://dzone.com/tutorials/java/struts/struts-example/struts-login-page-example-1.html>

<http://www.allapplabs.com/struts/struts_example.htm>

What is tiles?

Name some frameworks in Java to implement SOAP web services?

We can create SOAP web services using JAX-WS API, however some of the other frameworks that can be used are Apache Axis and Apache CXF. Note that they are not implementations of JAX-WS API, they are totally different framework that work on Servlet model to expose your business logic classes as SOAP web services. Read more at Java SOAP

Web Service Eclipse example.

What you use Executer?

What is the difference between Futures and Future task?

<http://www.journaldev.com/1090/java-callable-future-example>

What you know about callable?

How you write your own IOC container?

How reflection api helps in spring IOC?

How you write a class to parse XML?

How SubString method works in Java - Memory Leak Fixed in JDK 1.7  
  
Read more: <http://javarevisited.blogspot.com/2011/10/how-substring-in-java-works.html#ixzz4BlUiAMXH>

How to find palindrome in linked list?

How to kill a thread in java?

HCL

1. How Hashmap internaly work?
2. What is collision?

<http://javarevisited.blogspot.com/2011/02/how-hashmap-works-in-java.html>

1. Why linkedlist is used?
2. What is difference between iterator and listiterator?

<http://javahungry.blogspot.com/2015/08/difference-between-iterator-and-listiterator-with-example.html>

## Iterator vs ListIterator

1) Iterator is used for traversing List and Set both.

We can use ListIterator to traverse List only, we cannot traverse Setusing ListIterator.

2) We can traverse in only forward direction using Iterator.

Using ListIterator, we can traverse a List in both the directions (forward and Backward).

3) We cannot obtain indexes while using Iterator

We can obtain indexes at any point of time while traversing a list using ListIterator. The methods nextIndex() and previousIndex() are used for this purpose.

4) We cannot add element to collection while traversing it using Iterator, it throws ConcurrentModificationException when you try to do it.

We can add element at any point of time while traversing a list using ListIterator.

5) We cannot replace the existing element value when using Iterator.

By using set(E e) method of ListIterator we can replace the last element returned by next() or previous() methods.

6) Methods of Iterator:

* hasNext()
* next()
* remove()

Methods of ListIterator:

* add(E e)
* hasNext()
* hasPrevious()
* next()
* nextIndex()
* previous()
* previousIndex()
* remove()
* set(E e)

1. What is concurrent hashmap?

<http://javarevisited.blogspot.com/2013/02/concurrenthashmap-in-java-example-tutorial-working.html>

1. What is difference between fail safe and fail fast?
2. How ArrayList internally works?

<http://netjs.blogspot.in/2015/08/how-arraylist-works-internally-in-java.html>

1. How to write code for redirect and forward?

Redirect:

**import** java.io.\*;

**import** javax.servlet.\*;

**import** javax.servlet.http.\*;

**public** **class** DemoServlet **extends** HttpServlet{

**public** **void** doGet(HttpServletRequest req,HttpServletResponse res)

**throws** ServletException,IOException

{

res.setContentType("text/html");

PrintWriter pw=res.getWriter();

response.sendRedirect("http://www.google.com");

pw.close();

}}

<http://www.javatpoint.com/sendRedirect()-method>

Forward

**import** java.io.\*;

**import** javax.servlet.\*;

**import** javax.servlet.http.\*;

**public** **class** Login **extends** HttpServlet {

**public** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

    response.setContentType("text/html");

    PrintWriter out = response.getWriter();

    String n=request.getParameter("userName");

    String p=request.getParameter("userPass");

**if**(p.equals("servlet"){

        RequestDispatcher rd=request.getRequestDispatcher("servlet2");

        rd.forward(request, response);

    }

**else**{

        out.print("Sorry UserName or Password Error!");

        RequestDispatcher rd=request.getRequestDispatcher("/index.html");

        rd.include(request, response);

        }

    }

}

1. How to write code for comparator?
2. Does it use compare or comparator?
3. Do you know any design pattern other than singleton or factory?
4. Do you know about builders pattern?
5. What is decorator pattern?

<http://www.tutorialspoint.com/design_pattern/decorator_pattern.htm>

**Decorator design pattern** is used to modify the functionality of an object at runtime. At the same time other instances of the same class will not be affected by this, so individual object gets the modified behavior.

The concept of a decorator is that it adds additional attributes to an object dynamically. A real world example of this would be a picture frame. The picture is our object, which has it's own characteristics. For display purposes we add a frame to the picture, in order to decorate it. You're probably already familiar with the concept of wrapper objects, and in essence, that is what a Decorator is.

<https://dzone.com/articles/design-patterns-decorator>

1. How many ways to create objects?

**// Create Object1**

CrunchifyObj object1 = new CrunchifyObj();

**// Create Object2**

CrunchifyObj object2 =

(CrunchifyObj) Class.forName("crunchify.com.tutorial.CrunchifyObj").newInstance();

**// Create Object3**

CrunchifyObj secondObject = new CrunchifyObj();

CrunchifyObj object3 = (CrunchifyObj) secondObject.clone();

**// Create Object4**

object4 = CrunchifyObj.class.getClassLoader().loadClass("crunchify.com.tutorial.CrunchifyObj")

.newInstance();

**// Create Object5: Create a new file with an ObjectOutputStream**

FileOutputStream out = new FileOutputStream("crunchify.txt");

ObjectOutputStream oout = new ObjectOutputStream(out);

oout.writeObject(object4); // write object4 to file

oout.flush();

// create an ObjectInputStream for the file we created before

ObjectInputStream ois = new ObjectInputStream(new FileInputStream("crunchify.txt"));

CrunchifyObj object5 = (CrunchifyObj) ois.readObject();

log(object5.toString());

**// Create Object6: use the Constructor class from the java.lang.reflect**

Class clazz = CrunchifyObj.class;

Constructor crunchifyCon = clazz.getDeclaredConstructors()[0];

CrunchifyObj obj = (CrunchifyObj) crunchifyCon.newInstance();

1. How to connect multiple DBs using same class?

<http://www.java2s.com/Code/Java/Database-SQL-JDBC/Connecttomorethanonedatabase.htm>

String driver = "oracle.jdbc.driver.OracleDriver";

String url = "jdbc:oracle:thin:@localhost:1521:scorpian";

String username = "userName";

String password = "pass";

**Step 1: load driver by registering it to CopyOnWriteArrayList registeredDrivers.**

Load drivers: Class.forName(driver); // load Oracle driver

**public** **class** OracleDriver **implements** Driver

{

.

.

.

**static**

{

Timestamp localTimestamp = Timestamp.valueOf("2000-01-01 00:00:00.0");

**try**

{

**if** (defaultDriver == **null**)

{

defaultDriver = **new** OracleDriver();

DriverManager.registerDriver(defaultDriver);

}

}

**catch** (RuntimeException localRuntimeException) {}**catch** (SQLException localSQLException) {}

}

.

.

.   
**public** **class** DriverManager {

// List of registered JDBC drivers

**private** **final** **static** CopyOnWriteArrayList<DriverInfo> ***registeredDrivers*** = **new** CopyOnWriteArrayList<>();

.

.

.

**public** **static** **synchronized** **void** registerDriver(java.sql.Driver driver)

**throws** SQLException {

*registerDriver*(driver, **null**);

}

**public** **static** **synchronized** **void** registerDriver(java.sql.Driver driver,

DriverAction da)

**throws** SQLException {

/\* Register the driver if it has not already been added to our list \*/

**if**(driver != **null**) {

***registeredDrivers***.addIfAbsent(**new** DriverInfo(driver, da));

} **else** {

// This is for compatibility with the original DriverManager

**throw** **new** NullPointerException();

}

*println*("registerDriver: " + driver);

}

.

.

.

**Step 2: Get the connection**

Connection are created based on url. So we can load multiple drivers and get different connections based on url.

Connection conn = DriverManager.getConnection(url, username, password);

**public** **class** DriverManager {

**.**

.

.

**public** **static** Connection getConnection(String url,

java.util.Properties info) **throws** SQLException {

**return** (*getConnection*(url, info, Reflection.*getCallerClass*()));

}

**public** **static** Connection getConnection(String url,

String user, String password) **throws** SQLException {

java.util.Properties info = **new** java.util.Properties();

**if** (user != **null**) {

info.put("user", user);

}

**if** (password != **null**) {

info.put("password", password);

}

**return** (*getConnection*(url, info, Reflection.*getCallerClass*()));

}

// Worker method called by the public getConnection() methods.

**private** **static** Connection getConnection(

String url, java.util.Properties info, Class<?> caller) **throws** SQLException {

/\*

\* When callerCl is null, we should check the application's

\* (which is invoking this class indirectly)

\* classloader, so that the JDBC driver class outside rt.jar

\* can be loaded from here.

\*/

ClassLoader callerCL = caller != **null** ? caller.getClassLoader() : **null**;

**synchronized**(DriverManager.**class**) {

// synchronize loading of the correct classloader.

**if** (callerCL == **null**) {

callerCL = Thread.*currentThread*().getContextClassLoader();

}

}

**if**(url == **null**) {

**throw** **new** SQLException("The url cannot be null", "08001");

}

*println*("DriverManager.getConnection(\"" + url + "\")");

// Walk through the loaded registeredDrivers attempting to make a connection.

// Remember the first exception that gets raised so we can reraise it.

SQLException reason = **null**;

**for**(DriverInfo aDriver : ***registeredDrivers***) {

// If the caller does not have permission to load the driver then

// skip it.

**if**(*isDriverAllowed*(aDriver.driver, callerCL)) {

**try** {

*println*(" trying " + aDriver.driver.getClass().getName());

Connection con = aDriver.driver.connect(url, info);

**if** (con != **null**) {

// Success!

*println*("getConnection returning " + aDriver.driver.getClass().getName());

**return** (con);

}

} **catch** (SQLException ex) {

**if** (reason == **null**) {

reason = ex;

}

}

} **else** {

*println*(" skipping: " + aDriver.getClass().getName());

}

}

// if we got here nobody could connect.

**if** (reason != **null**) {

*println*("getConnection failed: " + reason);

**throw** reason;

}

*println*("getConnection: no suitable driver found for "+ url);

**throw** **new** SQLException("No suitable driver found for "+ url, "08001");

}

**public** Connection connect(String paramString, Properties paramProperties)

**throws** SQLException

{

**if** (paramString.regionMatches(0, "jdbc:default:connection", 0, 23))

{

String str1 = "jdbc:oracle:kprb";

**int** j = paramString.length();

**if** (j > 23) {

paramString = str1.concat(paramString.substring(23, paramString.length()));

} **else** {

paramString = str1.concat(":");

}

str1 = **null**;

}

**int** i = oracleDriverExtensionTypeFromURL(paramString);

**if** (i == -2) {

**return** **null**;

}

**if** (i == -3) {

DatabaseError.throwSqlException(67);

}

OracleDriverExtension localOracleDriverExtension = **null**;

localOracleDriverExtension = **this**.driverExtensions[i];

**if** (localOracleDriverExtension == **null**) {

**try**

{

**synchronized** (**this**)

{

**if** (localOracleDriverExtension == **null**)

{

localOracleDriverExtension = (OracleDriverExtension)Class.forName(driverExtensionClassNames[i]).newInstance();

**this**.driverExtensions[i] = localOracleDriverExtension;

}

**else**

{

localOracleDriverExtension = **this**.driverExtensions[i];

}

}

}

**catch** (Exception localException)

{

**throw** **new** SQLException(localException.toString());

}

}

walletLocation = paramProperties.getProperty("oracle.net.wallet\_location");

**if** (walletLocation == **null**) {

walletLocation = getSystemProperty("oracle.net.wallet\_location", **null**);

}

Hashtable localHashtable = parseUrl(paramString);

**if** (localHashtable == **null**) {

**return** **null**;

}

String str2 = paramProperties.getProperty("user");

String str3 = paramProperties.getProperty("password");

String str4 = paramProperties.getProperty("database");

**if** (str4 == **null**) {

str4 = paramProperties.getProperty("server");

}

**if** (str2 == **null**) {

str2 = (String)localHashtable.get("user");

}

str2 = parseLoginOption(str2, paramProperties);

**if** (str3 == **null**) {

str3 = (String)localHashtable.get("password");

}

**if** (str4 == **null**) {

str4 = (String)localHashtable.get("database");

}

String str5 = (String)localHashtable.get("protocol");

paramProperties.put("protocol", str5);

**if** (str5 == **null**)

{

DatabaseError.throwSqlException(40, "Protocol is not specified in URL");

**return** **null**;

}

**if** ((str5.equals("oci8")) || (str5.equals("oci"))) {

str4 = translateConnStr(str4);

}

String str6 = paramProperties.getProperty("oracle.jdbc.TcpNoDelay");

**if** (str6 == **null**) {

str6 = getSystemProperty("oracle.jdbc.TcpNoDelay", **null**);

}

**if** ((str6 != **null**) && (str6.equalsIgnoreCase("true"))) {

paramProperties.put("TCP.NODELAY", "YES");

}

String str7 = paramProperties.getProperty("oracle.jdbc.ReadTimeout");

**if** (str7 != **null**) {

paramProperties.put("oracle.net.READ\_TIMEOUT", str7);

}

**int** k = DriverManager.getLoginTimeout();

**if** ((k != 0) && (paramProperties.get("oracle.net.CONNECT\_TIMEOUT") == **null**)) {

paramProperties.put("oracle.net.CONNECT\_TIMEOUT", "" + k \* 1000);

}

String str8 = paramProperties.getProperty("prefetch");

**if** (str8 == **null**) {

str8 = paramProperties.getProperty("rowPrefetch");

}

**if** (str8 == **null**) {

str8 = paramProperties.getProperty("defaultRowPrefetch");

}

**if** ((str8 != **null**) && (Integer.parseInt(str8) <= 0)) {

str8 = **null**;

}

String str9 = paramProperties.getProperty("batch");

**if** (str9 == **null**) {

str9 = paramProperties.getProperty("executeBatch");

}

**if** (str9 == **null**) {

str9 = paramProperties.getProperty("defaultExecuteBatch");

}

**if** ((str9 != **null**) && (Integer.parseInt(str9) <= 0)) {

str9 = **null**;

}

String str10 = paramProperties.getProperty("defaultNChar");

**if** (str10 == **null**) {

str10 = getSystemProperty("oracle.jdbc.defaultNChar", **null**);

}

String str11 = paramProperties.getProperty("useFetchSizeWithLongColumn");

**if** (str11 == **null**) {

str11 = getSystemProperty("oracle.jdbc.useFetchSizeWithLongColumn", **null**);

}

String str12 = paramProperties.getProperty("remarks");

**if** (str12 == **null**) {

str12 = paramProperties.getProperty("remarksReporting");

}

String str13 = paramProperties.getProperty("synonyms");

**if** (str13 == **null**) {

str13 = paramProperties.getProperty("includeSynonyms");

}

String str14 = paramProperties.getProperty("restrictGetTables");

String str15 = paramProperties.getProperty("fixedString");

String str16 = paramProperties.getProperty("AccumulateBatchResult");

**if** (str16 == **null**) {

str16 = "true";

}

String str17 = paramProperties.getProperty("disableDefineColumnType");

**if** (str17 == **null**) {

str17 = "false";

}

String str18 = paramProperties.getProperty("oracle.jdbc.convertNcharLiterals");

**if** (str18 == **null**) {

str18 = getSystemProperty("oracle.jdbc.convertNcharLiterals", "false");

}

Enumeration localEnumeration = DriverManager.getDrivers();

**while** (localEnumeration.hasMoreElements())

{

localObject2 = (Driver)localEnumeration.nextElement();

**if** ((localObject2 **instanceof** OracleDriver)) {

**break**;

}

}

**while** (localEnumeration.hasMoreElements())

{

localObject2 = (Driver)localEnumeration.nextElement();

**if** ((localObject2 **instanceof** OracleDriver)) {

DriverManager.deregisterDriver((Driver)localObject2);

}

}

Object localObject2 = (PhysicalConnection)localOracleDriverExtension.getConnection(paramString, str2, str3, str4, paramProperties);

**if** (str8 != **null**) {

((PhysicalConnection)localObject2).setDefaultRowPrefetch(Integer.parseInt(str8));

}

**if** (str9 != **null**) {

((PhysicalConnection)localObject2).setDefaultExecuteBatch(Integer.parseInt(str9));

}

**if** (str12 != **null**) {

((PhysicalConnection)localObject2).setRemarksReporting(str12.equalsIgnoreCase("true"));

}

**if** (str13 != **null**) {

((PhysicalConnection)localObject2).setIncludeSynonyms(str13.equalsIgnoreCase("true"));

}

**if** (str14 != **null**) {

((PhysicalConnection)localObject2).setRestrictGetTables(str14.equalsIgnoreCase("true"));

}

**if** (str15 != **null**) {

((PhysicalConnection)localObject2).setDefaultFixedString(str15.equalsIgnoreCase("true"));

}

**if** (str10 != **null**) {

((PhysicalConnection)localObject2).setDefaultNChar(str10.equalsIgnoreCase("true"));

}

**if** (str11 != **null**) {

((PhysicalConnection)localObject2).useFetchSizeWithLongColumn = str11.equalsIgnoreCase("true");

}

**if** (str16 != **null**) {

((PhysicalConnection)localObject2).setAccumulateBatchResult(str16.equalsIgnoreCase("true"));

}

String str20 = getSystemProperty("oracle.jdbc.J2EE13Compliant", **null**);

String str19;

**if** (str20 == **null**)

{

str19 = paramProperties.getProperty("oracle.jdbc.J2EE13Compliant");

**if** (str19 == **null**) {

str19 = "false";

}

}

**else**

{

str19 = str20;

}

((PhysicalConnection)localObject2).setJ2EE13Compliant(str19.equalsIgnoreCase("true"));

((PhysicalConnection)localObject2).disableDefineColumnType = str17.equalsIgnoreCase("true");

((PhysicalConnection)localObject2).convertNcharLiterals = str18.equalsIgnoreCase("true");

localHashtable = **null**;

((PhysicalConnection)localObject2).protocolId = i;

**return** (Connection)localObject2;

}

1. How to write comparator for Employees class with name, department and age.

Write comparator for string.

public static Comparator<Fruit> FruitNameComparator

= new Comparator<Fruit>() {

public int compare(Fruit fruit1, Fruit fruit2) {

String fruitName1 = fruit1.getFruitName().toUpperCase();

String fruitName2 = fruit2.getFruitName().toUpperCase();

//ascending order

return fruitName1.compareTo(fruitName2);

//descending order

//return fruitName2.compareTo(fruitName1);

}

};

1. How to create cache in java?

<http://crunchify.com/how-to-create-a-simple-in-memory-cache-in-java-lightweight-cache/>

1. How to find 2nd highest number in an array?
2. How to find all leaf nodes in an binary tree?
3. Write an method that takes and array and a number and returns two numbers whose summation is the number passed to the method?
4. How to write a program that reads all the files of different format from a sever location and create single output by merging them.
5. How to design an application which will read data from DB and show on UI.