Cumber selenium test cases,mockito, @data Lombok, sonar cube, swagger, gargabe collector analyzer

What is micro services in java??

Ans. <https://dzone.com/articles/why-microservices>

Microservices are very trendy these days. Almost everybody is into it. It is not just Netflix, Amazon, or Google — it appears that almost everyone has adopted this architecture style. Although microservices have been here for quite some time now and a lot has already been written about them, I thought of writing yet another piece today, so please bear with me.

To understand the need for microservices, we need to understand problems with our typical 3-tier monolithic architecture.

**What Is Monolithic Architecture?**

Monolithic means composed all in one piece. A monolithic application is one which is self-contained. All components of the application must be present in order for the code to work.

Take the case of a typical 3-tier traditional web application built in three parts: a user interface, a database, and a server-side application. This server-side application is called a monolith, which is further divided into 3 layers — presentation, business layer, and data layer. The entire code is maintained in the same codebase. In order for the code to work, it is deployed as a single unit. Any small change requires the entire application to be built and deployed.

**What Is Microservices Architecture?**

Microservices architecture is an architectural style where the entire application is divided and designed as loosely-coupled, independent services modeled around a business domain. The "micro" in microservices is very deceiving. It has been debated a lot, but in my humble opinion, it does not dictate how small or big a service has to be. Again, this is another discussion we should have another day. Let's move forward.

The important point at this stage is that each independent service has a business boundary which can be independently developed, tested, deployed, monitored, and scaled. These can be even developed in different programming languages.

In microservices-based architecture, each component or service has its own database. There is no centralized database, as in the case of a monolith. You can even use NoSQL, RDBMS, or any other database as needed for each of the individual microservices. This makes microservices truly independent.

Let's now see what concerns microservices address.

1. How Spring boot application works??
2. down voteaccepted
3. Following is the high level flow of how spring boot works.
4. From the run method, the main application context is kicked off which in turn searches for the classes annotated with @Configuration, initializes all the declared beans in those configuration classes, and based upon the scope of those beans, stores those beans in jvm, specifically in a space inside JVM which is known as IOC container. After creation of all the beans, automatically configures the dispatcher servlet and registers the default handler mappings, messageConverts and all other basic things.
5. Basically spring boot supports three Embedded servers:- Tomcat (default), Jetty and Undertow.
6. You can add cross filters in spring boot in one of the configuration files as
7. @Configuration
8. @EnableWebMvc
9. public class WebConfig extends WebMvcConfigurerAdapter {
10. @Override
11. public void addCorsMappings(CorsRegistry registry) {
12. registry.addMapping("/api/\*\*");
13. }
14. }

2. what is use of @Bean at method level in Spring ??

**2.2. @Bean**

@Bean is a method-level annotation and a direct analog of the XML <bean/> element. The annotation supports most of the attributes offered by <bean/>, such as: [init-method](http://static.springframework.org/spring/docs/2.5.x/reference/beans.html#beans-factory-lifecycle-initializingbean), [destroy-method](http://static.springframework.org/spring/docs/2.5.x/reference/beans.html#beans-factory-lifecycle-disposablebean), [autowiring](http://static.springframework.org/spring/docs/2.5.x/reference/beans.html#beans-factory-autowire), [lazy-init](http://static.springframework.org/spring/docs/2.5.x/reference/beans.html#beans-factory-lazy-init), [dependency-check](http://static.springframework.org/spring/docs/2.5.x/reference/beans.html#beans-factory-dependencies), [depends-on](http://static.springframework.org/spring/docs/2.5.x/reference/beans.html#beans-factory-dependson) and [scope](http://static.springframework.org/spring/docs/2.5.x/reference/beans.html#beans-factory-scopes).

**2.2.1. Declaring a bean**

To declare a bean, simply annotate a method with the @Bean annotation. When JavaConfig encounters such a method, it will execute that method and register the return value as a bean within a BeanFactory. By default, the bean name will be the same as the method name (see [bean naming](https://docs.spring.io/spring-javaconfig/docs/1.0.0.M4/reference/html/ch02s02.html) for details on how to customize this behavior). The following is a simple example of a @Bean method declaration:

@Configuration

**public** **class** AppConfig {

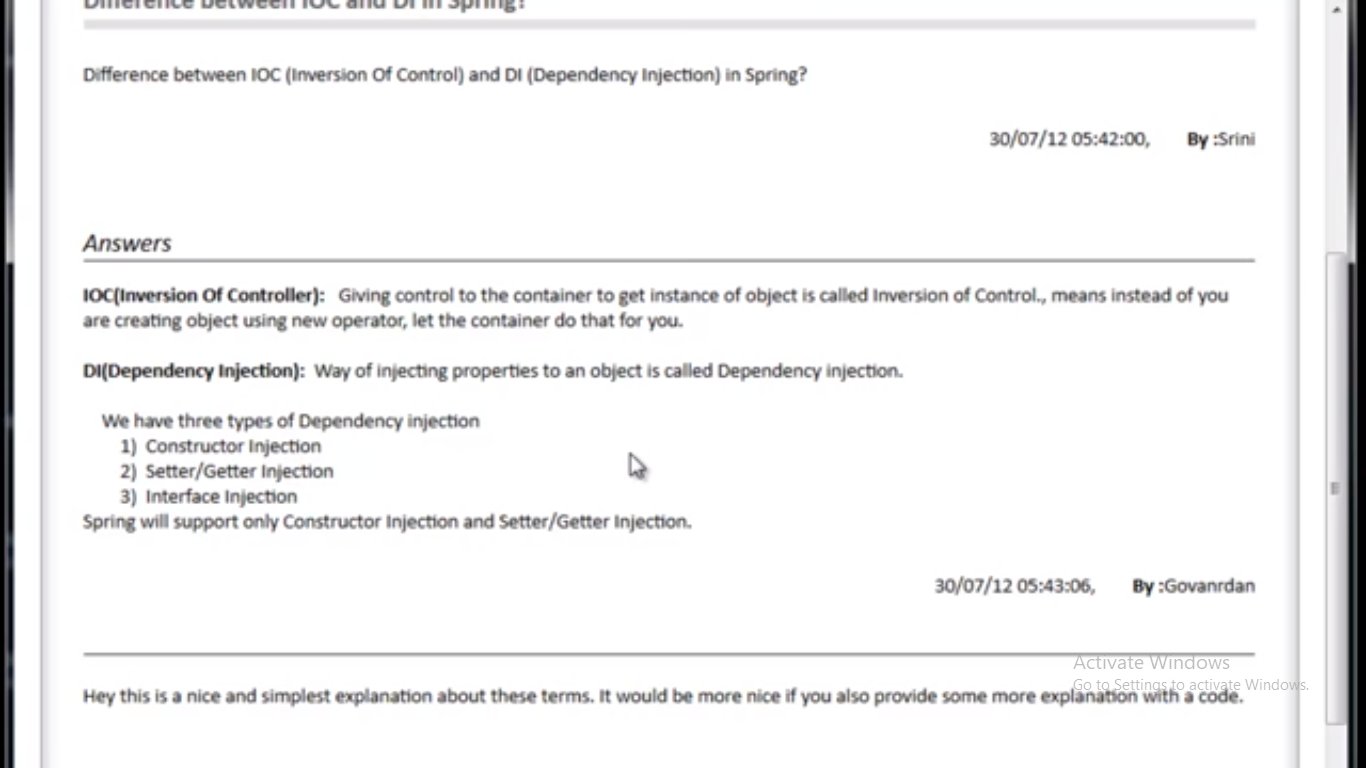
@Bean

**public** TransferService transferService() {

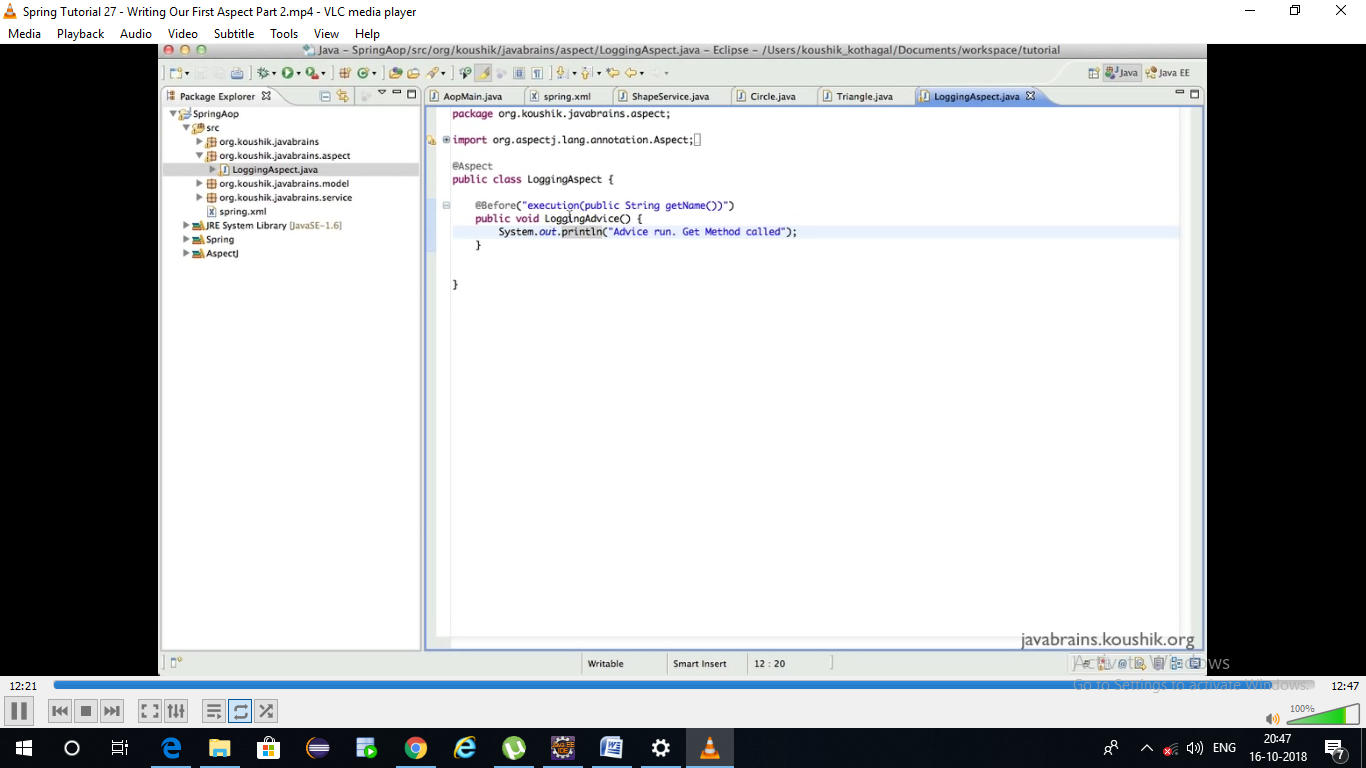
**return** **new** TransferServiceImpl();

TransferService transferService = **new** TransferServiceImpl();

3. Difference between IOC and Di in spring??



4. What is Spring AOP?



To make it clear first aspect means a class which contains different advice methods.

@Aspect

Public class LoggingAspect{

@Before(execute(public String getName()))

Public void LoggingAdvie(){}}

-----------------wild card-----------------------------

@Aspect

Public class LoggingAspect{

@Before(execute(\* get\*(\*)))

Public void LoggingAdvie(){}}

@Before(execute(\* get\*(..))) ------Zero or any no of parameters

What is pointCut: point is a point where Advice to be cut in

@Aspect

Public class LoggingAspect{

@Before(allGetters)

Public void LoggingAdvie(){}

@Before(allGetters)

Public void anotherAdvie(){}}

@PointCut(execute(\* get\*())---------------------🡪>>>>>>>>>>>Point cut

Public void allGetters()

**Java**

**What is the main difference between composition and aggregation??**

**Easily we can remember composition is the combination**

**Composition**

**Eg: If you close college all the departments will get closed which strong relation.**

**Class College{**

**Department department=new Department();**

**}**

**Aggregation**

**iIn university there are lot of people if u close the college still there is a chance of people can exists.**

**Class College{**

**Department department;**

**College(Department department)**

**{**

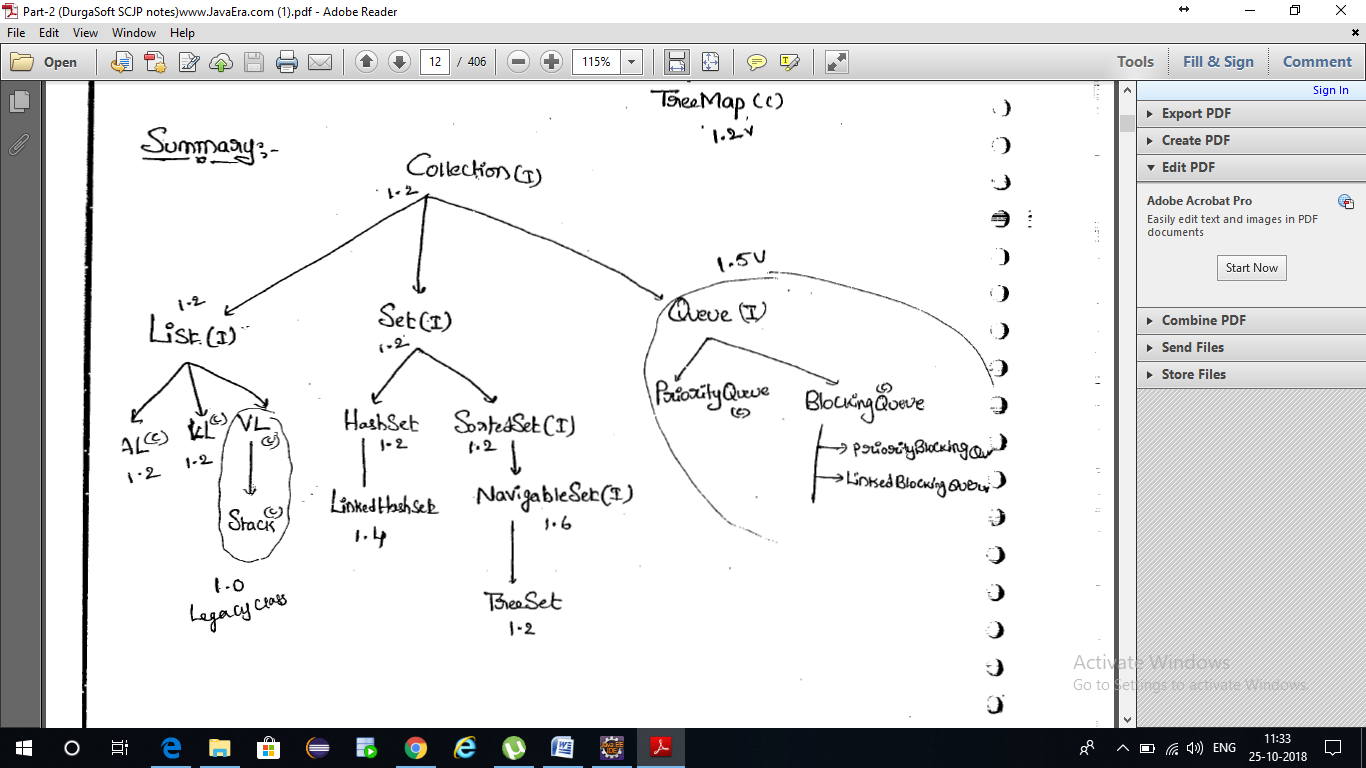
**This.department=department;**

**Collection**

**List—insertion order preserved +duplicates allowed.**

**Set—insertion order is not preserved + duplicates not allowed.**

**Queue—?**

****

Generics

1. As the type parameter we can use any java valid identifier

Class Test<X>

{

}

1. we can pass any no of type parameters and need not be one

Class test<X,T>

Bounded parameters:

Class Test<T extends Number>

{

}

Class A {

Test<Integer> test=new Test<Integer>();

}

If we are try to add other than number we will get compile time error

Test<String> txt=new Test<String>();--------Compile time error.

Conclusion

1. for any generic class

class A<T>{

}

Class A <T, Ramu, A>{

}

1. How to bound generic class

Class <T extends Number>{

}

1. Method generics allowing

Public void m1(List<? Extends Number>){

}

Public void m1(List<? Super Number>){

}

For already defined generic type classes how to bound ??

Example ArrayList already defined Like

Class Arraylist<E> for this how to bound??

ArrayList<? Extends Number> ll=new ArrayList<>();

Test<? Super Number> test=new Test<>();

1. Generics for compile time only that means Compiler will cheks from left side only not from right side.

ArrayList ll=new ArrayList<String> () --------for this we can add all the values along strings

Threads

1. Is Thread scheduler part of jvm or os? Or what is jvm scheduling algorithm?

It doesn't. The JVM uses operating system native threads, so the OS does the scheduling, not the JVM.

1. Synchronized.

Whenever multiple threads acts on same object then only synchronized effect will be there otherwise ther is no impact of synchronized keyword. Whenever multiple threads acts on multiple objects then there is effect of synchronize keyword.

1. Producer and consumer example

2. Dead lock programme

Java.lang package

1. Shallow cloning
2. Deep cloning
3. Write a immutable class like String
4. Why scp introduced in java??

In java lots of string literals will be used if every literal stores in heap the performance will be less so scp introduced. So every String literal will get stored in SCP.

1. Intern method String.

Since interning is automatic for String literals, the intern() method is to be used on Strings constructed with new String()

String s1 = "Javatpoint";

String s2 = s1.intern();

String s3 = **new** String("Javatpoint");

String s4 = s3.intern();

System.***out***.println(s1==s2); // True

System.***out***.println(s1==s3); // False

System.***out***.println(s1==s4); // True

System.***out***.println(s2==s3); // False

System.***out***.println(s2==s4); // True

System.***out***.println(s3==s4); // False

1. Why String made immutable??

To increase the performance SCP has introduced. In SCp every same String literal points to the same object if you do any modification on any object it will get on other objects so to eliminate this process String made immutable.

What are immutable,singleton,mand marker interfaces in java

Immutable ----String and wrapper classes

Singleton-----Runtime class(Runtime.getRunTime.gc())

Marker interface--- cloneable and serializale

String Buffer: whenever loading any video it will be in buffering state. That’s means it will be loading slowly. That means slow performance so only thread is allowed for String buffer.

Wrapper classes

1.The main objective of wrapper classes is to warp primitives into Objects form, so that primitives can be treated like Objects.

2. to define several utility methods for the primitives.

How Class Loader Works

1. Difference between Path and ClassPath in java?

Path is java/jdk/bin for java if you set path for path=java/jdk/bin then os can recognize javac and java commands in command prompt.

Classpath:

The CLASSPATH variable is one way to tell applications, including the JDK tools, where to look for user classes. (Classes that are part of the JRE, JDK platform, and extensions should be defined through other means, such as the bootstrap class path or the extensions directory.

Desc: write a java program in note pad and save it in desktop. Open a command prompt

C:\Users\Ramu B>cd Desktop

C:\Users\Ramu B\Desktop>javac InternExample2.java

InternExample2.java:1: error: error while writing InternExample2: InternExample2.class (The system cannot find the file specified)

public class InternExample2{

^

1 error

We get above error. Because didn’t set any classpath. Set the ClassPath and try we don’t get any error. Finally classpath means

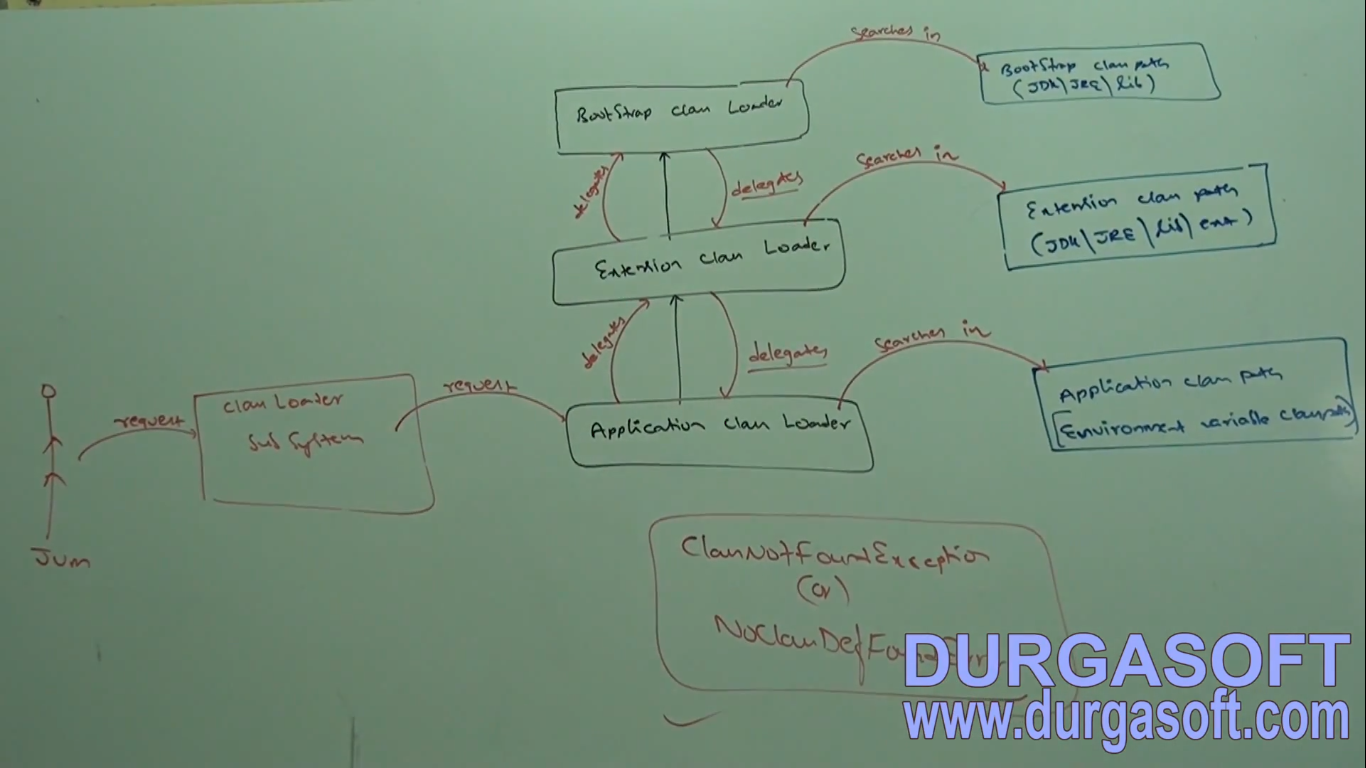
1. Boot Strap class loader(jdk/jre/lib)

2. Extension Class loader(jdk/jre/lib/ext)

3. Application class loader(from classPath)

1. On what hierarchy class loader subsystem works?

Delegation Hierarcchy principle.



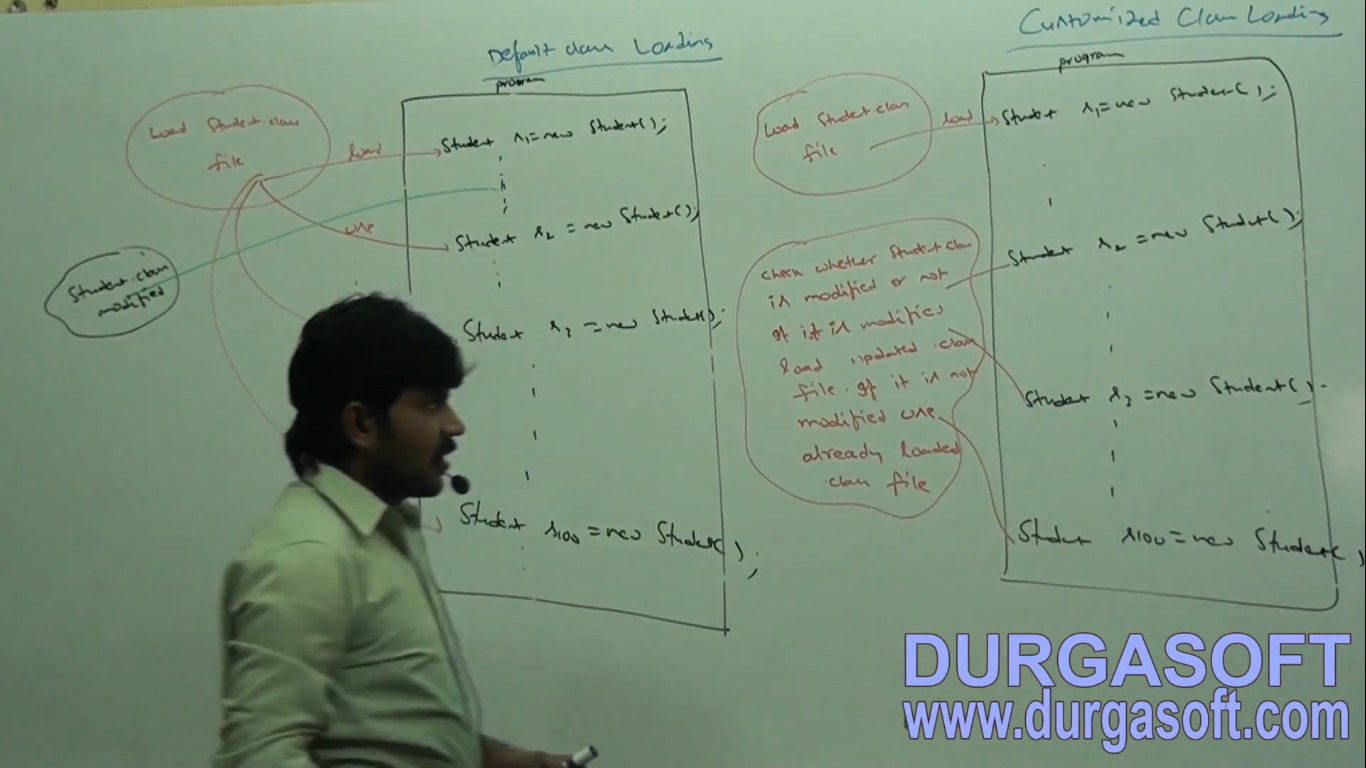
1. **package** com.practrisce.AfterHCL;
2. **import** java.beans.Customizer;
3. **public** **class** ClassloaderExample {
4. **public** **static** **void** main(String[] args) {
5. // **TODO** Auto-generated method stub
6. System.***out***.println(String.**class**.getClassLoader());
7. System.***out***.println(ClassloaderExample.**class**.getClassLoader());
8. System.***out***.println(Customizer.**class**.getClassLoader());
9. }
10. }
11. Out put:
12. null
13. sun.misc.Launcher$AppClassLoader@73d16e93
14. null

Create any Test.java and put this class in extension folder and print we get that particular class loader as extension class loader.

NEED OF Customized Class Loader.

Default class loaders loads .class files only once eventhough we are using multiple times that class in our programme. After .class file if it is modified out side then default class loader wont load updated version of class file(bcz .class file already available in method area). We can resolve this problem by defining our own customized class loader. The main advantage of customized class loader is we can control class loading mechanism based on our requirement.

Eg: we can load .class file seperately every time. so that updated version is available.



Class loader subsystem

1. Loading Verify
2. Linking prepare
3. Initialization Resolve

Memory areas

1. method area

2. Heap

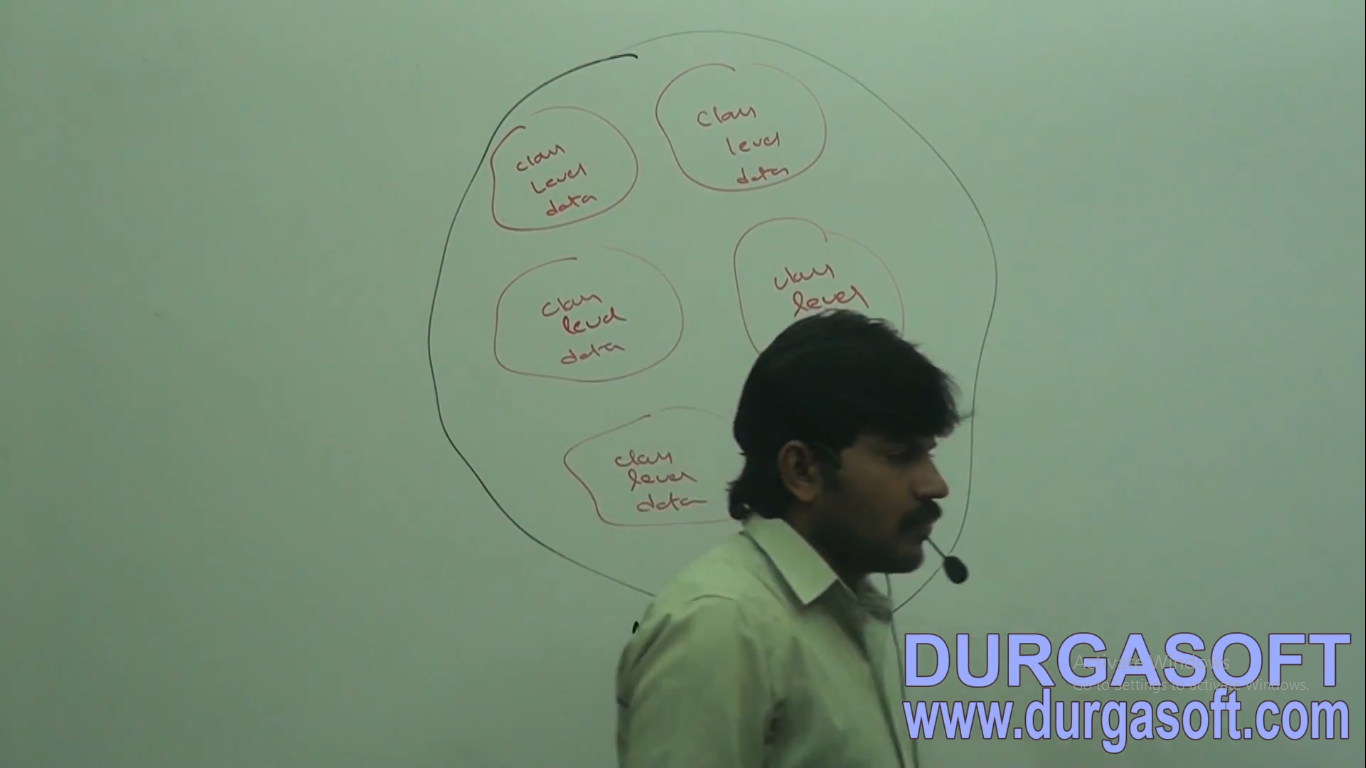
3. stack

4. native method stack

5. pc registers

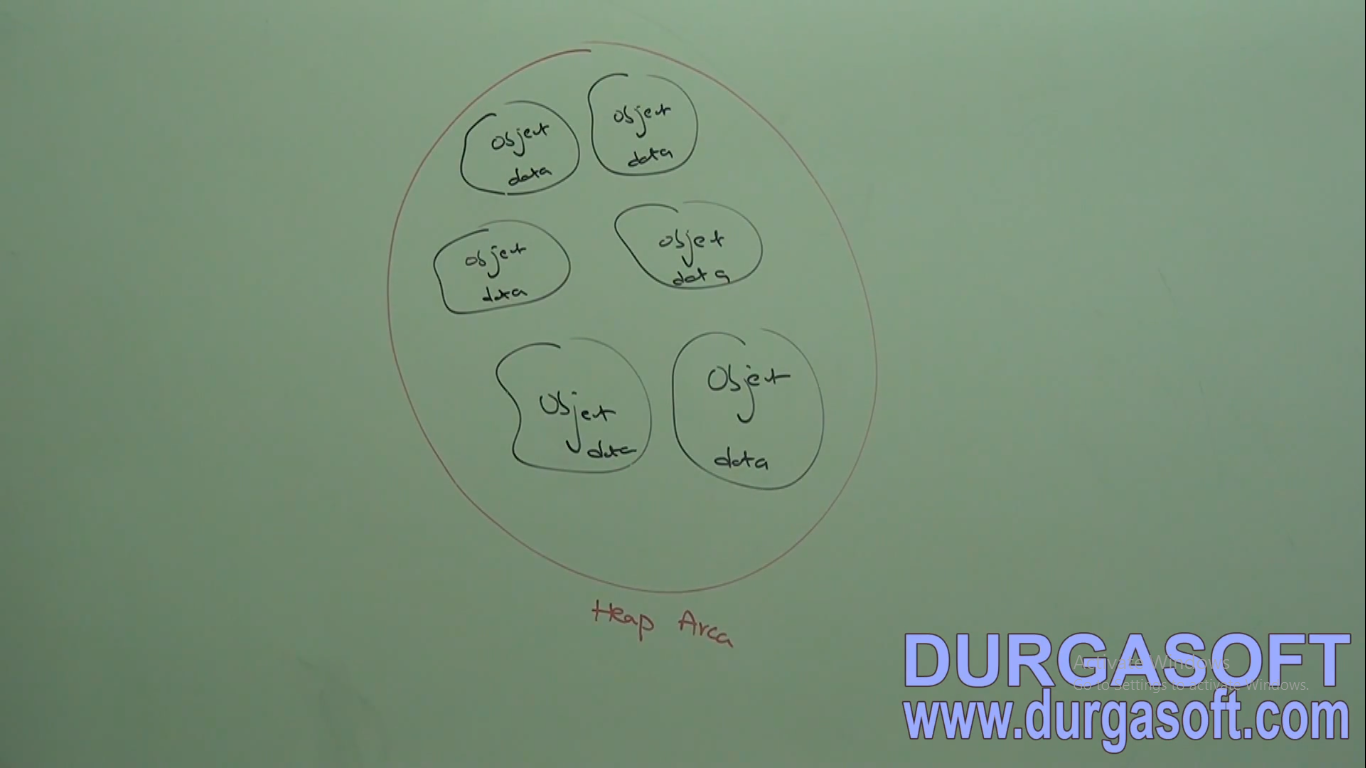
1. Method Area

For every JVM only one Method area available. Method area will be created at the time of jvm start up. Inside method area class level binary data including static variables will be stored. Constant pools of a class will be stored inside method area. Method area can be accessed by multiple threads simultaneously.



1. Heap Area

For every jvm one heap area is available. Heap area will be created at the time of jvm start up. Objects and corresponding instance variables will be stored in heap area. Every array in java is object only hence arrays also will be stored in the heap area. Heap area can be accessed by multiple threads and hence the data stored in the heap memory is not thread safe . heap area need not be continuous



A java application can communicate with jvm by using Runtime Object runtime class present in java.lang.package and it is singleton class. We can create runtime object as follows.

Runtime runtime=Runtime.*getRuntime*();

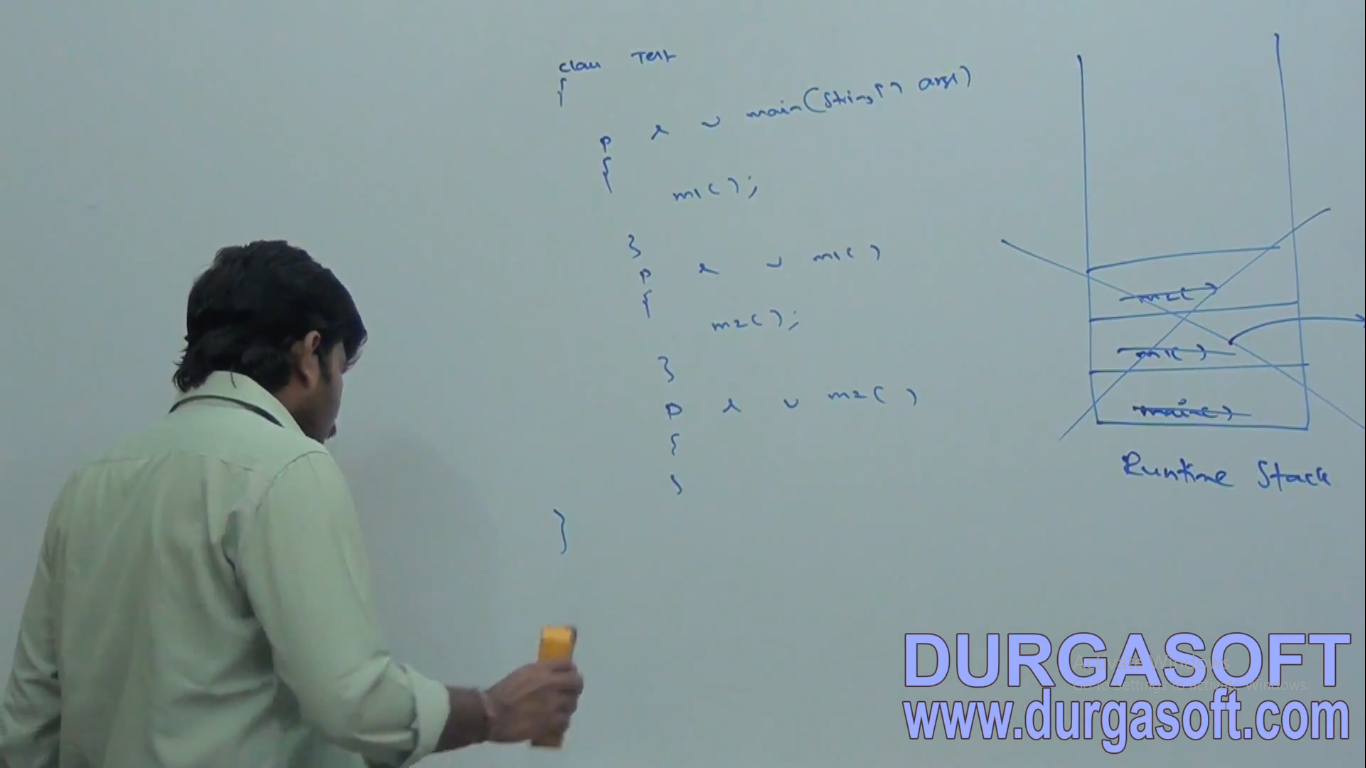
runtime.totalMemory();

runtime.freeMemory();

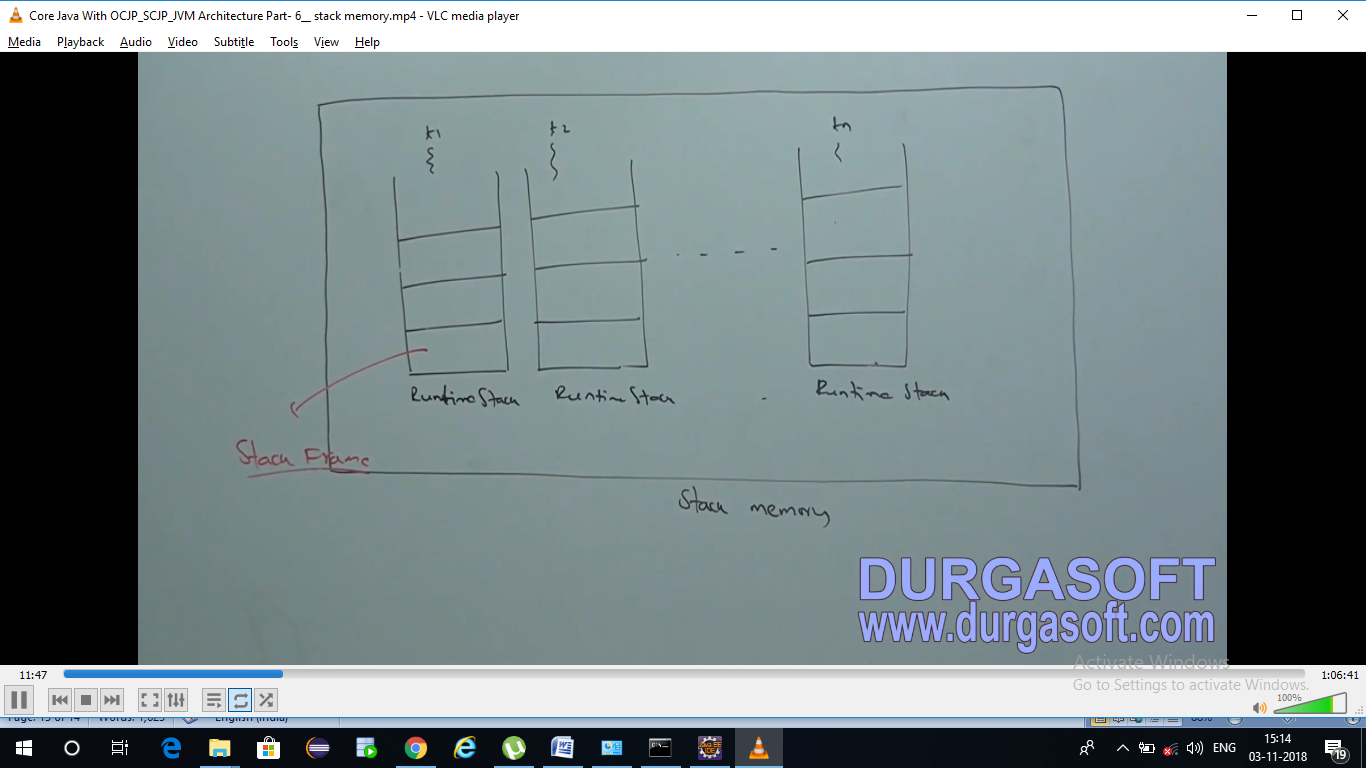
runtime.maxMemory();

Stack memory

Method area and heap per jvm one but stack memory is for every thread one stack . for every thread jvm will create a separate stack at the time of thread creation each and every method call performed by that thread will be stored in the satck including local variables also. After completing a method the corresponding from the stack will be removed after completing all the method calls the stack will become empty and that empty stack will be destroyed by jvm just before terminating the thread.

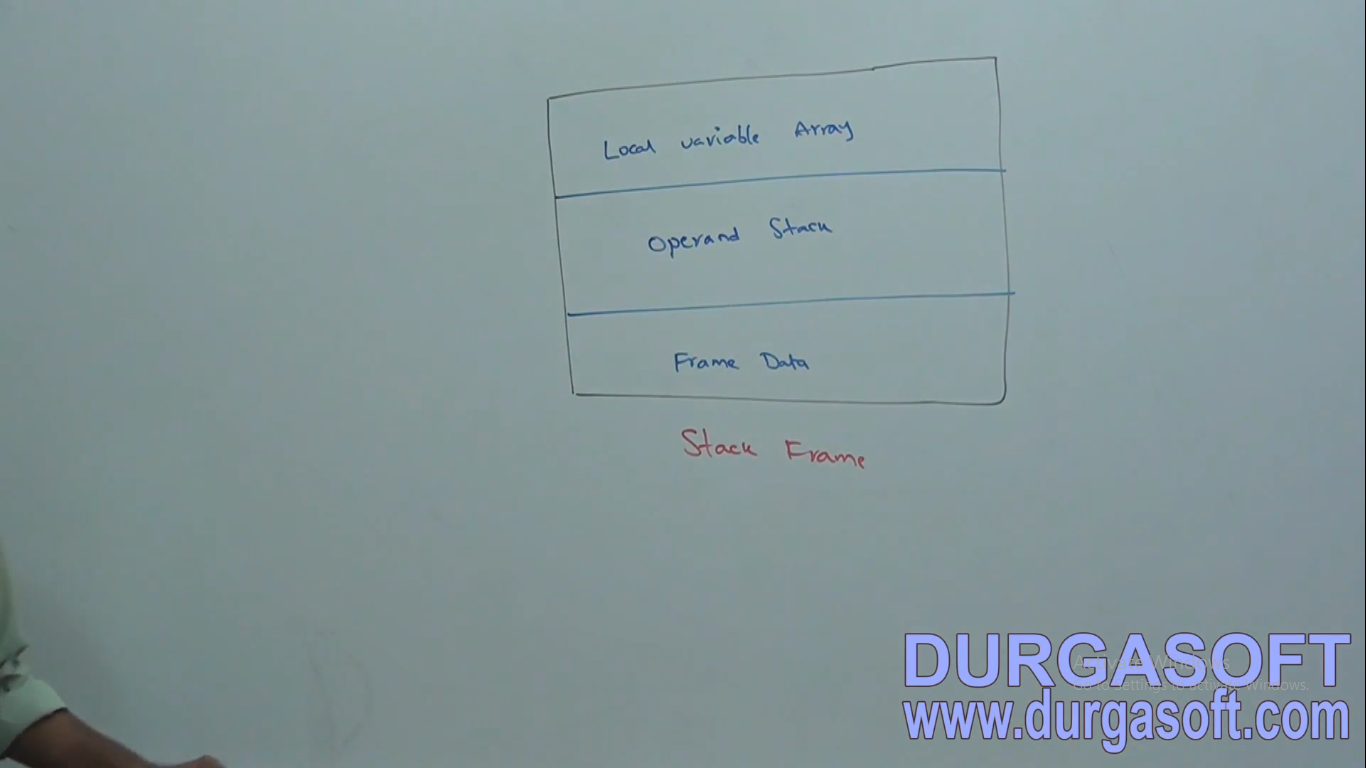


The data sored in the stack is available for the corresponding thread and not available to the remaining thread hence this data thread safe.



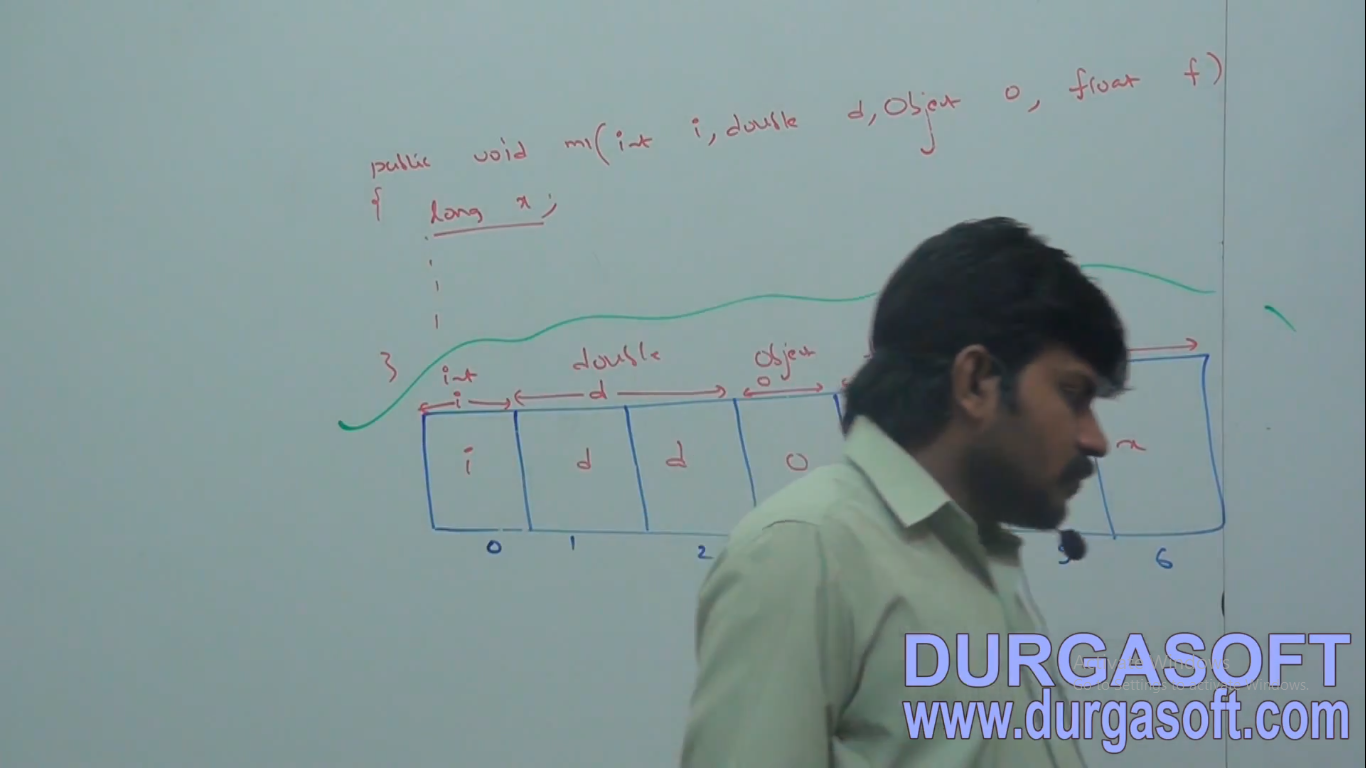
Each stack frame contains three parts

1. Local variable array
2. Operand stack
3. Frame data



Local variable array:

It contains all parameters and local variables of the method. Each slot in the array is of 4 bytes. Values of type int, float and reference occupy one entry in the array. Values of double and long occupy two consecutive entries in the array. Byte, short and char values will be converted to int type before storing and occupy one slot but way of storing Boolean values is varied from jvm to jvm but most of the jvms follow one slot for Boolean values.



# Operand stack

Jvm uses operand stack as work space. Some instructions can push the values to the operand stack and some instructions can pop values from operand stack and some instructions can perform required operations

PC Registers: programme Counters

Pc registers holds current execution instructions address. For every thread a separate PC register will be created at the time of thread creation. PC registers contains address of current execution instruction. Once instruction execution completes automatically pc registers automatically incremented to hold next instruction address.

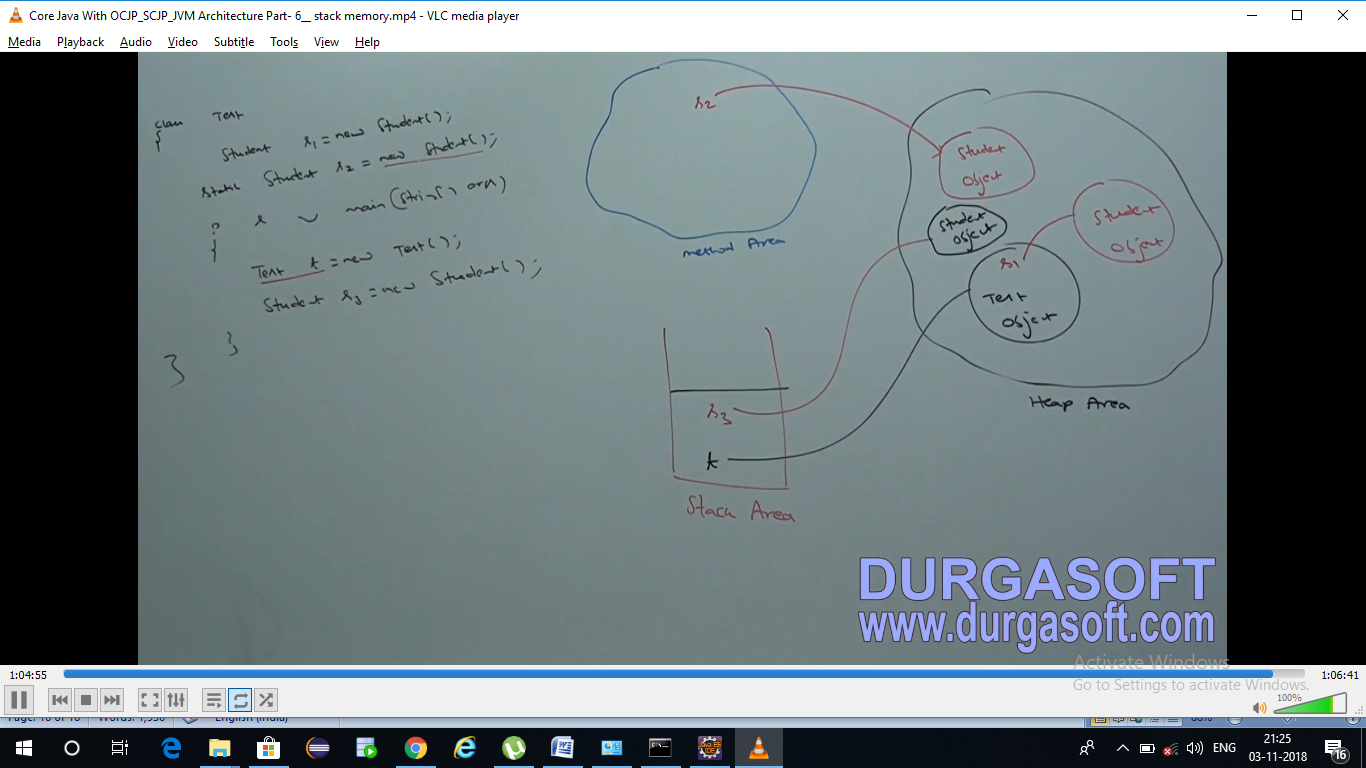
Native Method Stacks

For every thread jvm will create a separate native method stack all native method calls invoked by the thread will be stored in the corresponding native method stack.

For Wait() , hashcode()

Conclusions:

1. Method area, heap area and stack area are considered as important memory areas wrt to programmer.
2. Method area and heap are per jvm
3. Whereas stack area, pc registers and native method stacks are per thread.
4. Per every jvm one method area, heap area and per thread execution stack, pc registers and native method stack.
5. Static variables will be stored in method area, instance variables will be stored in heap area and local variable stored in stack area.



ap 02 th 2156

Execution Engine:

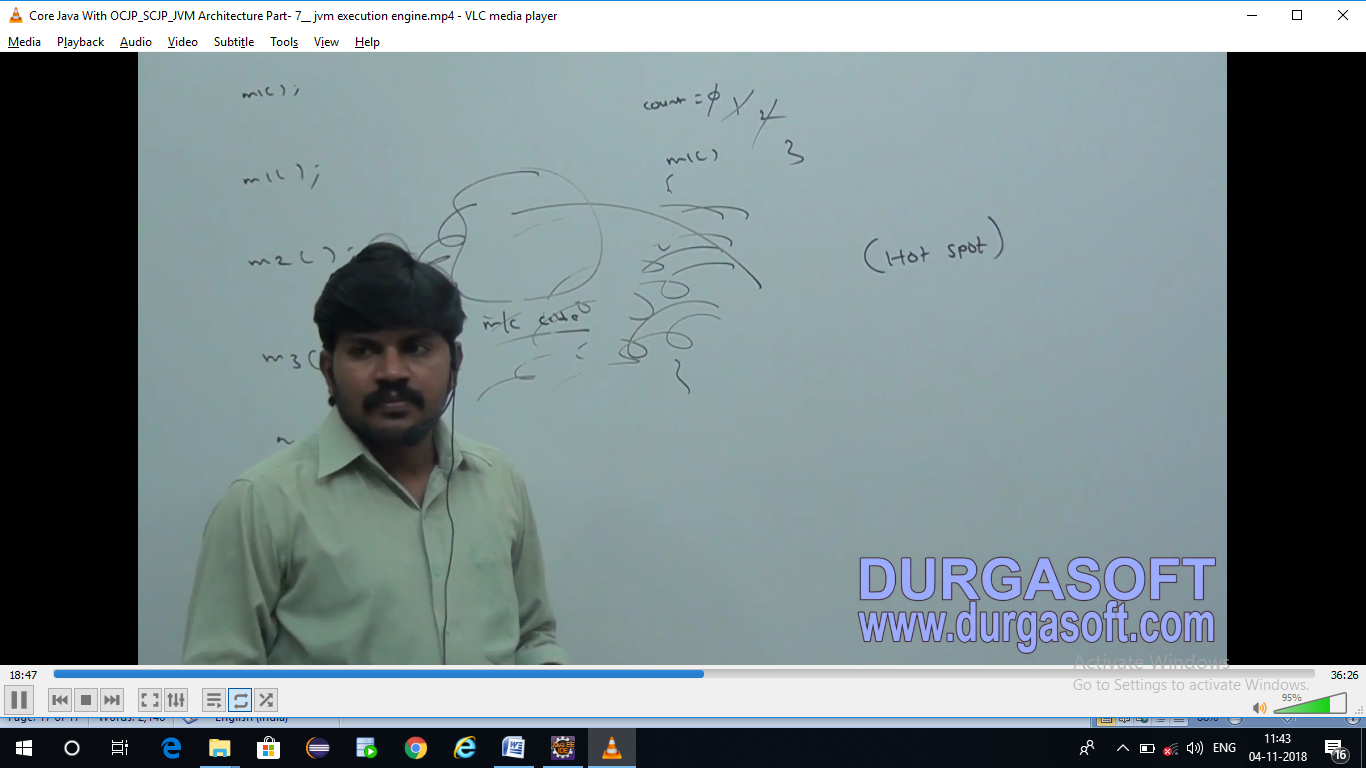
This is the central component of JVM. Execution engine is the responsible to execute java class files. Execution engine mainly contains two componentss1. Interpreter. 2.git compiler

1. Interpreter

It is the responsible to read byte code and interpret into machine code ndd execute that machine code line by line. The problem with interpreter is it interprets every time even same method is invoked multiple times which reduces performance of the system to overcome this problem sun people introduced jit compilers in 1.1 versions.

JIT compiler

The main purpose of jit compler is to improve performance . it internally maintains method count internally. Whenever jvm comes across any method call first that method will interpreted normally by the interpreter and jit compiler increments the corresponding count variable. This process will be continued for every method. Once if any method count reaches threshold value then jit compiler identifies that, that method method is repeatedly used method. Such method is called hot spot immediately jit compiler compiles that method and generates the corresponding native code. Next time jvm come across the method call then jvm uses native code directly and executes it instead of interpreting once again so thst performace of the system will be improved. Threshold count varied from jvm to jvm. Internally profiler , which is the part of jit compiler to identify hotspot. JVm interprets total programme atleast once.jit compilation is applicable only for repeatedly required methods not for every method.

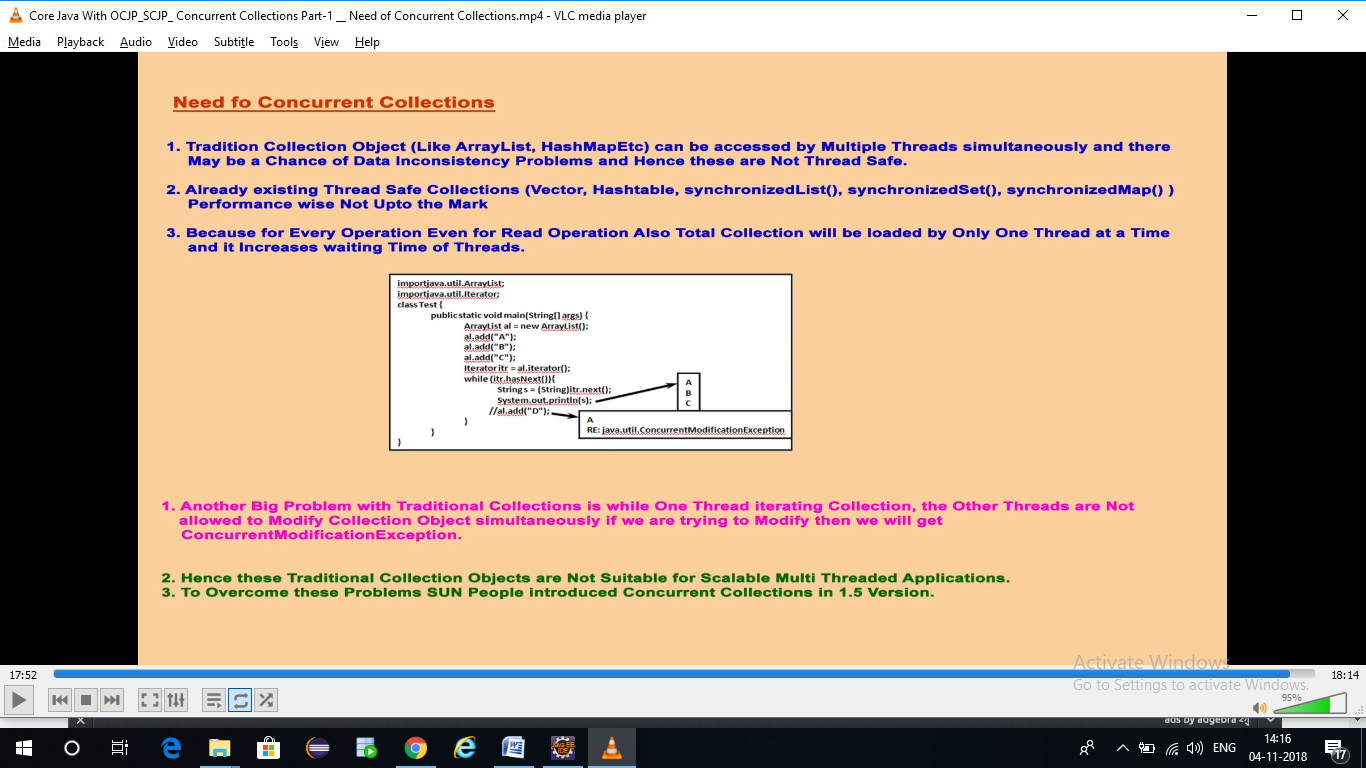
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

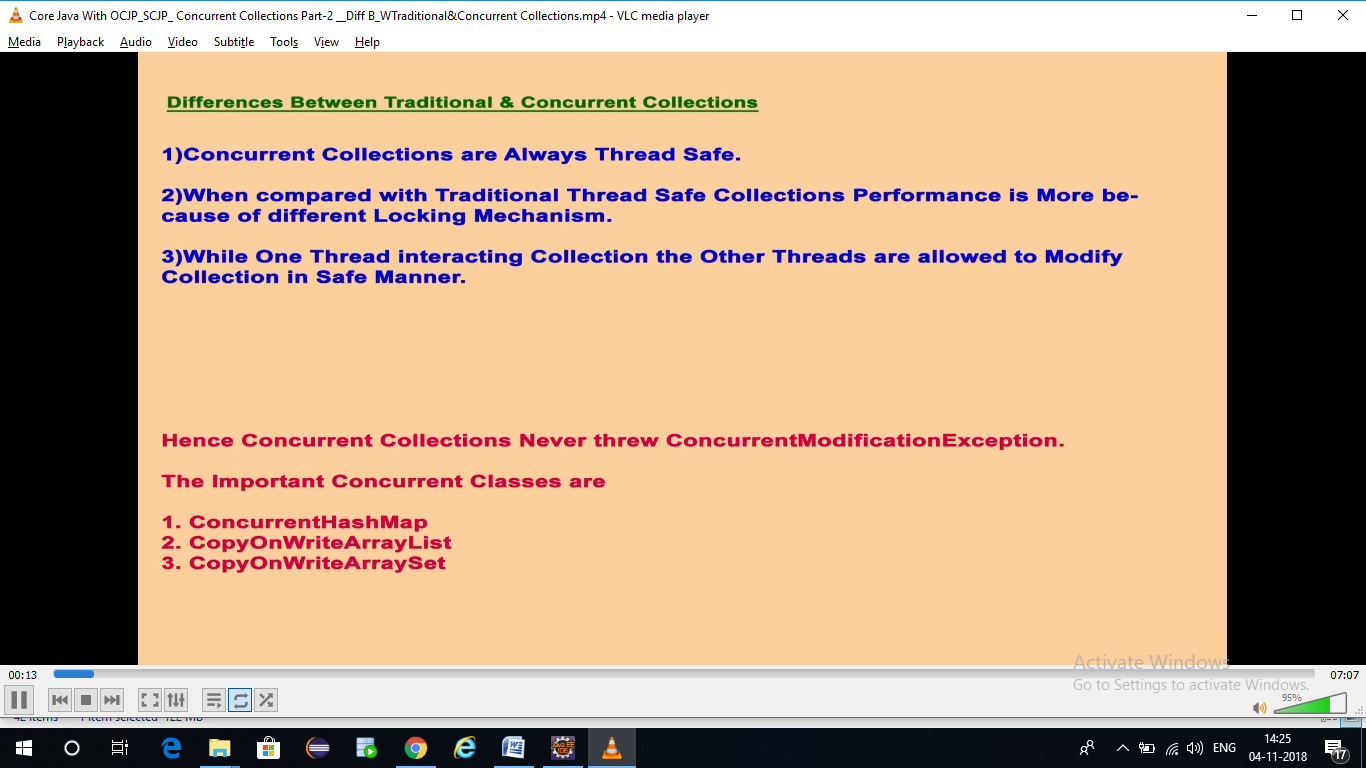
Need Of concurrent collections

1. Most of the collection objects are not thread safe
2. Some of the collection objects like VECTOR, HASTABLE, Synchronized List, SYNCHRONIZEDMAP, SYNCHRONIZEDSET are thread safe but performance is not good.
3. To avoid concurrent modification exception

When we get concurrent modification exception?

Whenever iterating elements of arraylist, hashmap,...etc if you do size modification then we get concurrent modification exception.





Interview programmes in java.

1.producer and consumer programme in java??

<https://javarevisited.blogspot.com/2012/02/producer-consumer-design-pattern-with.html>

2. deadlock programme in threads

3. singleton programme

4. immutable programme.

5. deep cloning and shallow cloning

CTS 09/02/12019.

1. File1: some text

File2: some text

How many times a word present in a file 1 is repeated in file2.

1. **try** {

**return** "try";

} **catch** (Exception e) {

**return** "catch";

}**finally** {

return "finaly";

}

What is the output?

1. How to reverse a linked list? 1 2 3 9 7
2. **try** {

**return** "try";

} **catch** (Exception e) {

**return** "catch";

} **catch** (NullPoinetException e) {

**return** "catch";

}

Is the code compiles??

1. String str=” ramu Bangalore anantapur saying”;

How to reverse each word from the given string?

1. Find the given number is palindrome or not?
2. Public void m1(Object o);

Public void m1(Double i);

Public void m1(Float f);

M1(1); which is being called?

M1(1.1) which is being called?

1. Palindrome program?

EY 11/02/2019

1. When do you prefer linked list and when do you prefer array list in real world?
2. See the Durga soft videos.
3. Difference between array list and linked list?
4. If you are not implementing serialization is that be serialiazed?
5. Exception in thread "main" java.io.NotSerializableException: com.hcl.serializationexample.SerializationExp.
6. What is the use of serial version uid?
7. How do you provide validations in spring?
8. How do you display controller response in the jsp page in spring mvc?
9. What are inheritance types in java? And which is good one?
10. Exception hierarchy?
11. What is the difference between no class def found and class not found exception in java?
12. Difference between path and class path in java?

HCL Internal

1. Design patterns used in your project?

2. what is singleton DP? Can we serialize it? What is lazy loading and early loading in singleton design pattern

3.what is double check in singleton Design pattern?

Altimetrik 20/02/2019

1. Singleton programme? Complete about singleton?
2. Reverse singly linked list?
3. How to sort values based on values?
4. oneTomany programme in hibernate?
5. difference between first level cache and second level cache?
6. Difference truncate and delete?
7. Difference get and post ?
8. What is race around condition?
9. How many ways we can create a thread?
10. Difference hashmap concurrent hash in java?

Aricent 23/02/2019

1. String s=”abcancbdf” find the first repeated character?

2. difference between DDL and DDl?

3. Spring boot flow and Spring mvc flow

4. where do you use wait notify?

5. mappings in Hibernate? Where do you in your project?

6. how do you provide json format in jersey implementation?

7. what is @generated value in Hinbernate?

8. first level cache and second level cache hibernate?

9. what is meant by lazy and early loading in hibernate in mapping ?

10.waht are the fetch types in hibernate?

11.craete two threads where one thread prints even numbers ther prints odd numbers?

12. create two threads where first thread prints 1 and second thread prints 2 and first thread prints 3 and second thread 4 and so on…

13. what is synchronized keyword in java?

14.when do get class level lock?

15. your query is slow how do you improve performance??

16. find third highest salary from salary?

select amount from (select \* from RESTAURANT where ROWNUM<=3 ORDER BY AMOUNT ) where ROWNUM<=1

society general 2/03/2019

1. String s=”ramu bangalore” op=”bangalore ramu”
2. When do you use linked list and arraylist?

HCL Interview internal 08/03/2019

1. String s1=null; String s2=”ramu”; s1.equals(s2);?
2. Difference between beanfactorycontainer and applicationcontext container?
3. String s1=” ramu bandela”;

String s2=”ramu”;

String s3=”bandela”;

String s4=s2+s3;

Sysout(s1==s4);

1. When do you use rest and when do you use soap?
2. When do you use aop?
3. How do you make spring application makes to enable annotations?
4. What are the scopes in spring?

**ApplicationContext beanFactory= new ClassPathXmlApplicationContext("Beans.xml");**

Student bean =(Student) beanFactory.getBean("student");

Student bean1 =(Student) beanFactory.getBean("student");

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

System.***out***.println(bean1);

ApplicationContext beanFactory1= **new** ClassPathXmlApplicationContext("Beans.xml");

Student bean2 =(Student) beanFactory1.getBean("student");

In this scenario how many objects are being created? Ans 2

Indecomm 09/03/2019

1. How to create immutable class?
2. Many to many in hibernate?
3. Hibernate transitions?
4. How to improve query performanece/?
5. Triggers, procedures and view?
6. How to call procedures and functions in hibernate?
7. Class A{
8. Final static List<Employee> list;

Static{

List=getting 10 objects from DB and to assign to list;

//now one more record added into DB then how do you fetch 11 records from DB assign to static variable? How do you refresh the static block?

}

}

1. String s=”assignment”; print the repeated characters?
2. How do you test the Restful web services?
3. Class A{

Static int x=10;

}

Class B extends A

{

Static int x=20;

Public static void main(String args[])

{

Sysout(x);

B b=new B();

Sysout(b.x);

}

}

1. Try with resource?
2. Performance tuning in oracle?
3. Why wrapper classes came into picture?
4. can we serialize immutable class?
5. How do you override hashcode and equals method?
6. Why do we override hashcode and equals methods?

Indecomm 2nd technical

1. int a=[123,456123,7654123]; print pattern 123 in ascending order?

2. OOPs concepts in java

Manger round

1. How linked Hash Map works internally?
2. Why two types of exceptions required in java why cant we use try every where?
3. Write a linked list program to add a element at I th position?
4. If controller scope is singleton and service layer scope is proto type I am calling service layer from controller. Here how many objects created?
5. What is lazy loading in hibernate?
6. Class Student{

int id;

String name;

Address add;

}

Class Address

{

int no;

String street;

}

If given lazy=true how it works in hibernate? How proxy pattern works in hibernate

|  |  |  |
| --- | --- | --- |
| id | name | Reporng to |
| 1 | Ramu Bandela employee | 2 |
| 2 | Anupama manager | 3 |
| 3 | Srinivas ceo | Null |

Write a query to get name and manger name from above table?

1. DataJPA repository is interface where the its methods are implemented?
2. How do you provide security in restful ws?
3. How the hashset works?
4. What is fail fast iterator and fail safe iterator?
5. In table 8 duplicate rows are there how do you delete them?
6. What is table space ?
7. What is data base and what is table and what is synonym?
8. Difference between primary key and unique key?
9. In table can i insert more than one uniq key?
10. What is thread local

Mindtree 23/03/2019

1. oops concepts

2. how do you provide connection pooling hibernate? Second level cache?

3. leftouter join?

4. second highest salary?

5. how arraylist implemented in java?

6. difference between abstract class and interface?

Capgemini telephonic 27/03/2019.

1. memory management in java?

2. what is metaspace?

3. what is thread local?

4. what is memory leak in java?

5. how to vaoid memory leak in java?

6. hibernate spring integration ?

**Singleton Design Pattern**

Why creating a singleton class so hard?

Ans. Singleton class can be violated when

1. Reflection
2. Serialization and deserialization
3. Clone
4. Multi threaded access
5. Multiple class loaders
6. Garbage collection

package com.company;  
  
public class SingletonClass {  
  
 public static SingletonClass *singletonClassObj*=null;  
  
 private SingletonClass()  
 {  
  
 }  
 public static SingletonClass getInstance()  
 {  
if(*singletonClassObj*==null)  
 *singletonClassObj*=new SingletonClass();  
 return *singletonClassObj*;  
 }  
  
 public static void main(String[] args) {  
 SingletonClass s1=SingletonClass.*getInstance*();  
 SingletonClass s2=SingletonClass.*getInstance*();  
 System.*out*.println(s1+" "+" "+s2);  
  
  
 }  
}

com.company.SingletonClass@3fb6a447 com.company.SingletonClass@3fb6a447

Reflection

package com.company;  
  
import java.lang.reflect.Constructor;  
  
 public class SingletonClass {  
  
 public static SingletonClass *singletonClassObj*=null;  
  
 private SingletonClass()  
 {  
  
 }  
 public static SingletonClass getInstance()  
 {  
if(*singletonClassObj*==null)  
 *singletonClassObj*=new SingletonClass();  
 return *singletonClassObj*;  
 }  
  
  
}  
  
  
class Test{  
 public static void main(String[] args)throws Exception {  
 SingletonClass s1=SingletonClass.*getInstance*();  
 SingletonClass s2=SingletonClass.*getInstance*();  
 System.*out*.println("s1:"+s1);  
 System.*out*.println("s2: "+s2);  
  
  
 Class clazz=Class.*forName*("com.company.SingletonClass");  
 Constructor<SingletonClass> constructor= clazz.getDeclaredConstructor();  
 constructor.setAccessible(true);  
 SingletonClass s3=constructor.newInstance();  
 System.*out*.println("s3:"+s3);  
  
  
  
 }  
}

fix: in constructor throw exception if object already created.

private SingletonClass()  
 {  
if(*singletonClassObj !=null*)

{

Throw RunTimeExceptio(“cant create object , Please use the reflection”);

}

Serialization:

class Test{  
 public static void main(String[] args)throws Exception {  
   
 SingletonClass s2=SingletonClass.*getInstance*();  
   
 System.*out*.println("s2: "+s2);  
  
  
   
  
 ObjectOutputStream oos=new ObjectOutputStream(new FileOutputStream("H:\\Java Tricky Questions\\s2.ser"));  
 // ObjectOutputStream oos=new ObjectOutputStream(new FileOutputStream("xyz/s2.ser"));  
 oos.writeObject(s2);  
  
 ObjectInputStream ois=new ObjectInputStream(new FileInputStream("H:\\Java Tricky Questions\\s2.ser"));  
 System.*out*.println("s3 "+ois.readObject());  
  
  
 }  
}

s2: com.company.SingletonClass@3498ed

s3 [com.company.SingletonClass@5ae9a829](mailto:com.company.SingletonClass@5ae9a829)

fix:

**private** Object readResolveObject()**throws** Exception  
{  
 **return** *singletonClassObj*;  
}

Clone

class Test{  
 public static void main(String[] args)throws Exception {  
 // SingletonClass s1=SingletonClass.getInstance();  
 SingletonClass s2=SingletonClass.*getInstance*();  
 // System.out.println("s1:"+s1);  
 System.*out*.println("s2: "+s2);  
SingletonClass s3=(SingletonClass) s2.clone();  
 System.*out*.println("s3:"+s3);

s2: com.company.SingletonClass@3498ed

s3:com.company.SingletonClass@1a407d53

Fix: In clone method return same instance whatever created in getInstance method.

@Override  
**protected** Object clone() **throws** CloneNotSupportedException {  
  
  
 **return** *singletonClassObj*;  
}

Multi Thread environment

**class** Test{  
  
 **static void** printInstances()  
 {  
 SingletonClass s=SingletonClass.*getInstance*();  
 System.***out***.println(s+**" multi thread"**);  
 }  
  
 **public static void** main(String[] args)**throws** Exception {ExecutorService executorService= Executors.*newFixedThreadPool*(2);  
 executorService.submit(Test::*printInstances*);  
 executorService.submit(Test::*printInstances*);  
 }  
}

com.company.SingletonClass@6b11037 multi thread

com.company.SingletonClass@4d4a1eeb multi thread

fix: 1. use Double check and volatile keyword

**private static volatile** SingletonClass *singletonClassObj*=**null**;

**public static** SingletonClass getInstance()  
 {  
**if**(*singletonClassObj*==**null**)  
 **synchronized** (SingletonClass.**class**)  
 {  
 **if**(*singletonClassObj*==**null**)  
 {  
 *singletonClassObj*=**new** SingletonClass();  
 }  
 }  
  
 **return** *singletonClassObj*;  
 }

Multiple Class loaders

Garbage collection

...............................................................................................................................................................

Garbage Collection

Younger generation Older Generation

Eden Space

Survivor Survivor

Space from space to

Visionet 25/01/2024

How to configure any other server than apache tomcat?

What is cold start in AWS Lambda?

How tto create lambda using spring boot?

How garbage collector works in java 8?

Lisknove substition principles in solid design principle?

How to create a lambda function using spring boot?

GSpann 25/01/2024

Cold start in aws lambda

If we don’t call a lambda function for 5 or 10 mints,it goes into sleep mode. it will take time to get up if we call again. start around 5 seconds this is called cold start.

Serevrless-plugin-warmup

How to mitigate cold start in lambda

Lambda limitations

Sqs and sns

Cloud formation

Can multiple consumers can consume data from sqs?

RDS

Tesco 25/01/2024

1. I have 100 employees among them 5 are duplicates, now tell me if I store them into arraylist and hashset sizes?

Arraylist size is 100

Hashset size is 100

Now what to do remove duplicates from hashset

Override equals and hsahcode methods

1. Reverse a string by using recursion.
2. Sort employees based on names and then based salary.

25/01/2024

1. Immutable class how do you create ?
2. If immutable class has any list can we modify the size?

Final class I{

Private final int I;

Private final Arralist<Integer> I;

Public I (int I, List<Integer> list){

This.i=I;

This.list=list;

}

getters

}i

Main(){

New I();

}

1. Thraed Local and AtomicInteger
2. Java memory management
3. Deadlock in java
4. Race condition in java
5. @Springboot application replaces which annotations
6. Triggers in DB
7. Overloading and overriding
8. Overloading with String and Object types
9. Memory management in java
10. list--for loop list stream difference
11. Obserevr design pattern
12. A--singleton

{

B b;--proto type

}

ac.getBean(A.class());

ac.getBean(A.class());

ac.getBean(A.class());

ac.getBean(A.class());

ac.getBean(A.class());

how many instances of A and B created.

1. Can I create like this List<String> ll=List<Object>
2. interface I{
3. public List<Vehicle> get();
4. }
5. class A implements I{
6. public List<Car> get(){
7. }
8. }
9. class B implements I{
10. public List<Boat> get(){
11. }
12. }

Is the above code correct?

1. I want remove duplicates and key value pair should be presented which collection is preferred.