----------------------------------------------Singleton------------------------------------------------https://www.oracle.com/technetwork/java/index.html

http://www.codespaghetti.com/java-design-pattern-interview-questions/

1. What is Singleton Design Pattern?
   1. Single point of access and shared resources
   2. Loading configuration files
   3. Memory consumption
   4. cons: loosing loose coupling, introducing global state, increasing dependency
   5. <https://javarevisited.blogspot.com/2011/03/10-interview-questions-on-singleton.html>
2. How enum searlization works?
3. Is it better to make whole get Instance () method synchronized or just critical section is enough? Which one you will prefer?

- Just critical section, because synchronized cause performance issue.

1. What is double checked locking in Singleton?

Answer: Double checked locking is a technique to prevent creating another instance of Singleton when call to getInstance () method is made in multi-threading environment. In Double checked locking pattern as shown in below example, singleton instance is checked two times before initialization.

public static Singleton getInstance () {

if (\_INSTANCE == null) {

synchronized (Singleton.class) {

//double checked locking - because second check of Singleton instance with lock

if (\_INSTANCE == null) {

\_INSTANCE = new Singleton ();

}}}

return \_INSTANCE;}

1. How do you prevent for creating another instance of Singleton using clone () method?

* Throw Exception in clone method.

1. How do you prevent for creating another instance of Singleton using reflection?

* This can be avoided by throwing Exception from constructor as “Singleton already initialized”.

1. How do you prevent for creating another instance of Singleton during serialization?

* by using read Resolve () method, since during serialization read Object () is used to create instance and it return new instance every time but by using read Resolve you can replace it with original Singleton instance

1. Difference between Singleton Pattern vs Static Class in Java

* Static class is faster than single Tone class, due to compile time binding.
* If object state needs to maintain then single tone classes are good.
* <http://www.crazyforcode.com/cant-static-class-singleton>.
* <http://www.crazyforcode.com/cant-static-class-singleton>
* https://javarevisited.blogspot.com/2013/03/difference-between-singleton-pattern-vs-static-class-java.html

1. What is the difference b/w java Single Tone and Spring single Tone?

* https://stackoverflow.com/questions/15252284/what-is-the-difference-between-aspring-singleton-and-a-java-singeletondesign-p
* The Java singleton is scoped by the Java class loader, the Spring singleton is scoped by the container context.

1. - Why you should avoid the singleton anti-pattern at all and replace it with DI?

* Singleton Dependency Injection: every class that needs access to a singleton gets the object through its constructors or with a DI-container.

1. Why Singleton is Anti pattern

* With more and more classes calling getInstance () the code gets more and more tightly coupled, monolithic, not testable and hard to change and hard to reuse because of not configurable, hidden dependencies. Also, there would be no need for this clumsy double-checked locking if you call getInstance less often (i.e. once).

1. Singleton Design Pattern scope at jvm, class Loader and application level.

* Class Loader level.

---------------------------------------Factory and Abstract Factory------------------------------------------

1. What is Factory Design Pattern?

* multiple subclasses and one super class based on input create object.
* every time need to change client code.

1. What is Abstract Factory Design Pattern?

* Abstract Factory Design Pattern is factory of Factory. Each type having his own factory to create object.

1. When use Factory and when use Abstract Factory?
2. What is difference b/w Factory and when use Abstract Factory?

* Factory pattern based on inheritance.
* Abstract factory pattern based on composition.
* The main difference between a “factory method” and an “abstract factory” is that the factory method is a single method, and an abstract factory is an object.
* The factory method is just a method, it can be overridden in a subclass, whereas the abstract factory is an object that has multiple factory methods on it.
* The Factory Method pattern uses inheritance and relies on a subclass to handle the desired object instantiation.

---------------------------------------------Builder Design Pattern--------------------------------------------

1. What is builder Design Pattern?

* Used to create complex immutable object.
* Used inner classes to create obj.
* Ex. String Buffer, StringBuilder

1. When to use Builder Design Pattern?

* To get immutable object with optional attribute.

----------------------------------------Prototype Design Pattern----------------------------------------------

1. Prototype Design pattern?

* When object creation costly and time consuming then we use prototype design pattern.
* Use shallow or deep cloning.

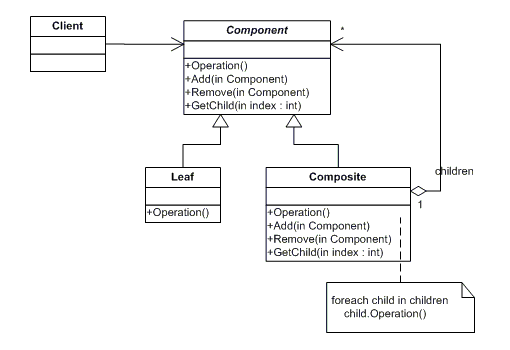
-----------------------------Structure Design Pattern---------------------------------------------------------

1. What is Adapter Design pattern?

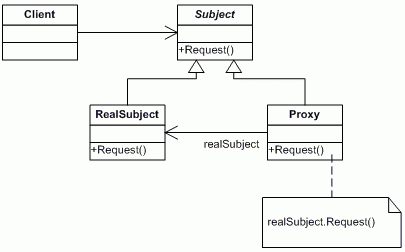
<https://javabypatel.blogspot.com/2016/06/adapter-design-pattern-java.html>



1. What is Composite Design Pattern?



1. What is Proxy Design Pattern?



1. What is Flyweight Design Pattern?

* When we need to create n Number of object again and again then use Flyweight Design Pattern ?

----------------------------------------------Behavior Design Pattern -------------------------------

----------------------------------Template method Design--------------------------

- Better way of interaction b/w objects.

- use composition rather than inheritance.

- super class provide Skelton and matain order of execution.

houseconstruction, Order processing

https://sourcemaking.com/design\_patterns/mediator/java/2

Mediator Desing Pattern :

- One object maintain different object interation.

- Promote many-to-many relationship b/w objects.

- Desing intermidiatry to decople object;

------------------------------------Spring Inerviwe Questin ;--------------------------

https://spring.io/docs/reference

- How to escape to a bean from autowiring ?

- How Handler Mapping works internally ?

- What is look-up method ?

----------------------------------------------------------------------------------------------------------------------

http://web.cs.ucla.edu/classes/winter15/cs144/projects/axis/services.html

glide.wsdl.schema.UnqualifiedElementFormDefault=false

https://github.com/eugenp/tutorials

https://github.com/skprasadu

https://www.logicbig.com/tutorials/spring-framework/spring-web-mvc/message-converter-registration.html

https://www.baeldung.com/spring-httpmessageconverter-rest

https://www.baeldung.com/spring-httpmessageconverter-rest

https://www.baeldung.com/spring-mvc-content-negotiation-json-xml

https://stackoverflow.com/questions/16332092/spring-mvc-pathvariable-with-dot-is-getting-truncated

https://www.baeldung.com/spring-httpmessageconverter-rest

https://www.baeldung.com/spring-mvc-image-media-data

https://diterius.wordpress.com/2013/09/11/spring/

http://bitbucket.spartasystems.com/projects/TWTA/repos/tw-web-ta/pull-requests/1093/diff

-----------------------------------------------------------------------------------------------------------

cas with springboot:

https://github.com/kakawait/cas-security-spring-boot-starter/blob/master/README.md

----------------------------------------------------------------------------------------------------

sessionid :

https://julienprog.wordpress.com/2017/08/17/session-id-in-the-url-is-it-a-vulnerability/

https://fralef.me/tomcat-disable-jsessionid-in-url.html

-----------------------------------------------------------------------------------------------------

Ordermanagement Project :

https://github.com/thangbn/order-management

https://github.com/PatrickCallaghan/order-management-demo

https://github.com/amanganiello90/java-angular-web-app

https://github.com/creativetimofficial/material-dashboard-angular2

https://github.com/DavideViolante/Angular-Full-Stack

https://github.com/ikismail/ShoppingCart

https://github.com/sivadass/react-shopping-cart

https://github.com/ikismail/Angular-ShoppingCart

ebook:

https://github.com/clojurians-org/blockchain-ebook

https://github.com/xieren58/the-best-blockchain-books/tree/master/books

https://www.educba.com/design-pattern-interview-questions/