DESIGN PATTERNS

**Creational design pattern** -- How we create Objects .. for ex:- Singleton dp, Prototype dp

**Structural design pattern** -- How we organize the Objects .. whether there is inheritance reln or association ..

**Behavioural design pattern** -- How objects behave or interact with each other ..

**Creational :-** (few examples)

Singleton - runtime

factory - calender.getInstane()

prototype - Object.clone

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Structural :-** (few examples)

Adapter pattern - One format to another format.

ex:- Arrays.asList().. List is converted to Array and Marshalling & UnMarshalling ( marshalling -> Converts from Objects to XML).

flyweight :- Where a particular object gets used multiple times .. ex. Integer.valueOf(int) .. (all wrapper classes)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Behavioral:-** (few examples)

1. Chain of Responsibility - .. Exception Handling delgation

2. Command - How Request to Response cycle works...

ex. Sending HTML data in form is a type of Command pattern . Hence the request from client side is sent to server using browser.

Servlet takes this command object as input, processes it, and sends the response back to the client.

ex. Creating thread using java.lang.Runnable

3. Iterator - way of looping around

4. Strategy - Uses different Algorithms..

For ex:- in Collections.sort().. we can send the compartor for the way we want to sort the objects in collection

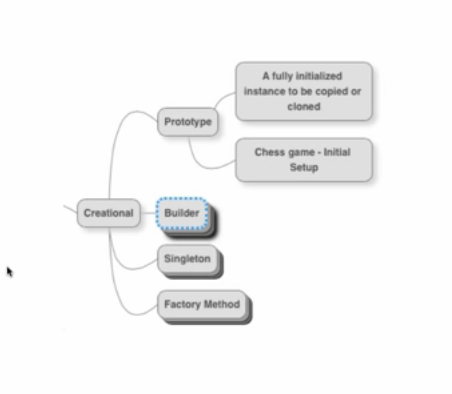
5. Template - Recognized by behaviour methods which already have a default behaviour defined by an abstract type.

For ex:- All non-abstract methods of java.util.AbstractSet, AbstractList and AbstractMap gives the default behaviour.

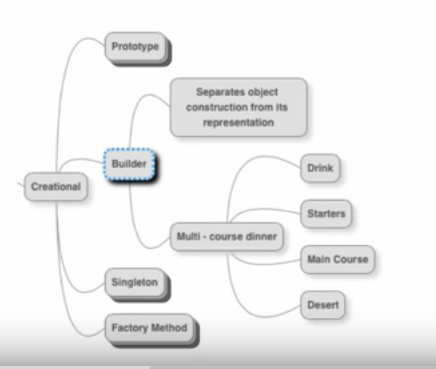
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CREATIONAL DESIGN PATTERN

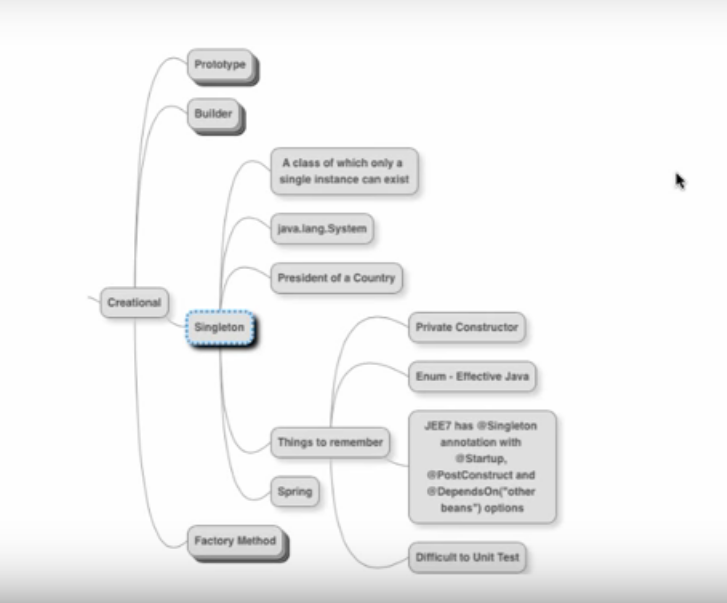
1. PROTOTYPE DESIGN PATTERN
2. BUILDER DESIGN PATTER
3. FACTORY DESIGN PATTERN
4. SINGLETON DESIGN PATTERN
5. PROTOTYPE DESIGN PATTERN



1. BUILDER DESIGN PATTERN



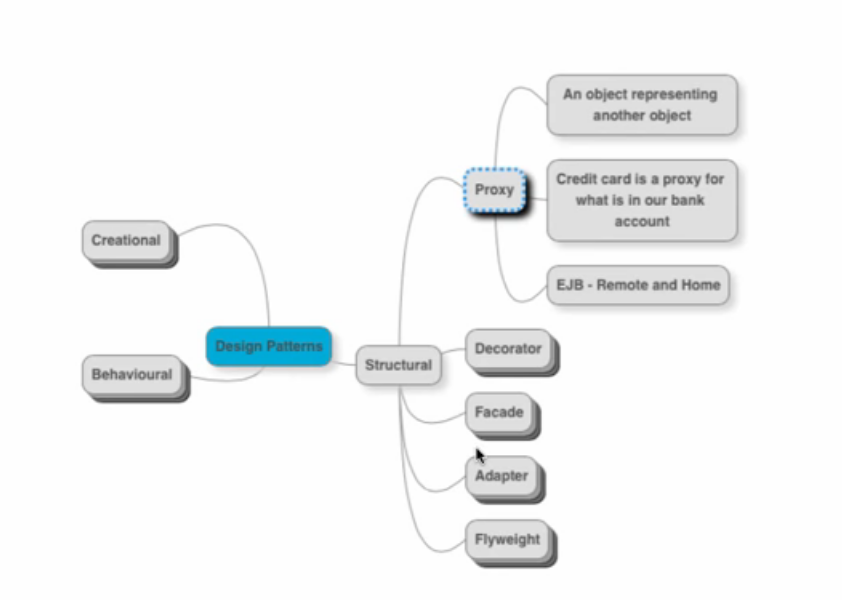
1. Singleton Design pattern:-



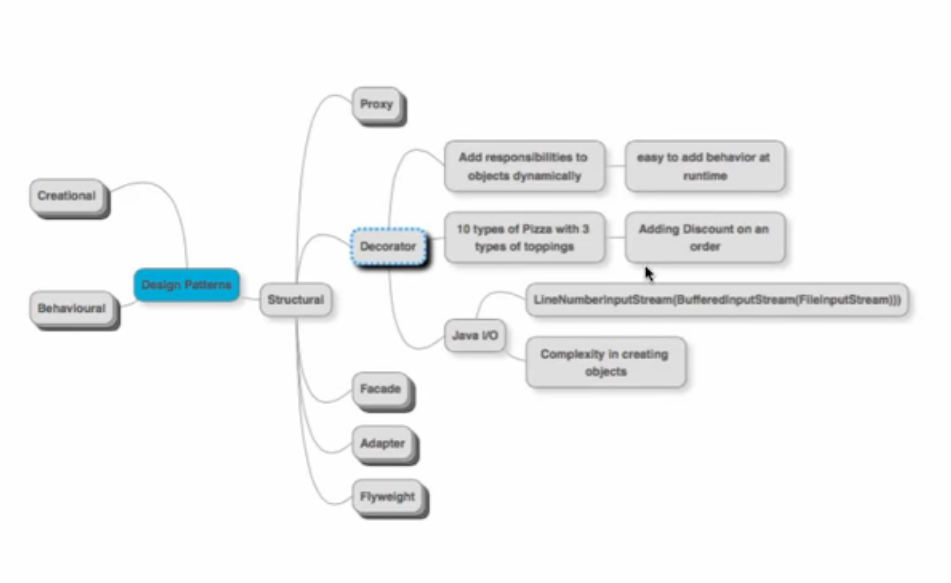
1. Factory Design Pattern

Structural Design pattern:-

1. Proxy Design pattern:-

De

1. Decorator Design pattern:-



1. Facacde Design pattern



1. Adaptor Design Pattern:-

