

* What is design pattern? Explain Advantages of design pattern.

7 Design Pattern :-

- Design pattern is a general repeatable solution to a commonly occurring problem in software design.
- A design pattern isn't a finished design that can be transformed directly into code.
- It is a description or template for how to solve problem that can be used in many different situations.

⇒ Use.

- Design pattern speed up the development process by providing tested.
- Reusing design patterns helps to prevent issue that cause major problems and improves code readability.

⇒ Advantages of Design Pattern :-

- They are reusable in multiple projects.
- They provide the solution that help to define system.
- They provide transparency to the design of an application.
- They are well-proved and testified solution since they built the knowledge & experience of S/w developers.
- It don't guarantee an absolute solution to a problem.
- They provide ~~readability~~ clarity & possibility of building better system.

* What are the elements of design pattern? Give brief description on it.

There are 23 design patterns which can be classified in three categories; -

- Creational
 - Structural
 - Behavioral
- } Types

and also discuss other category of design pattern :
J2EE

Creational :-

These design patterns are all about class or object creation.

These can be categorized into class-creational & object-creational patterns.

while class creational patterns use inheritance effectively in the process, object-creational patterns use effectively to get the job done.

Creational patterns are Factory method,

Abstract Factory, Builder
Singleton, Object Pool and
Prototype.

Structural.

- These design patterns are about organizing different classes and objects to form larger structures.
- structural design patterns are Adapter, Bridge, Composite, Decorator, Facade, Private class Data & Proxy.

Behavioral.

- Behavioral patterns are about identifying common communication patterns b/w objects & realize these patterns.
- Behavioral patterns are Chain of responsibility, Command, Interpreter, Iterator, mediator, Null object, State, Template method, Visitor.

⇒ J2EE patterns :-

These design patterns are specifically concerned with the presentation tier.

- These patterns are identified by Sun Java Center.

How design pattern solve design problems?
⇒ Explain in brief.

⇒ Solving Design problems with Design Patterns :-

⇒ Finding Appropriate Objects :-

- Object-oriented programs are made up of objects.
- The hard part about design is decomposing a system into objects.
- Design pattern can help in this process by less abstractions and objects.

⇒ Determining Object Granularity :-

- Objects vary in size and number.
- They can represent everything down to the h/w or entire applications.

- It can help to determine proper object granularity.
- A number of other patterns, such as Composite, describe how to decompose an object into smaller objects.

⇒ Specifying Object Interfaces :-

- Design patterns help programmers to define interfaces by identifying their key elements.
- It can also tell what not to put in the interface.
- For ex, the Memento pattern describes how to save the internal state of an objects,
 - A restricted one that lets clients hold and copy mementos.
 - A privileged one that only the original objects can use to store & retrieve state in mementos.

⇒ Designing for change :-

- Designing a system that is robust to change is a rather hard task to do.
- A design that doesn't take changes into account risks major redesign in future.
- Design patterns can ensure that system can change in specific ways.
- Each design pattern lets some aspect of the system structure of other aspects, thereby making a system more robust to a particular kind of change.

* List all steps to describe design pattern

⇒ Documentation of Design Pattern: -

⇒ Pattern name & classification.

- The pattern name specifies the pattern precisely.

- A good name is a key as it will become a part of our design vocabulary.

- classification specifies the type of pattern.

⇒ Intent.

- A short statement that tell us what the pattern does and which design issue & pattern addresses.

⇒ Also known as.

- Other well known names for the pattern.

Motivation.

- A scenario of design pattern, how the class & object solve the problem.
- Scenario will help to understand the pattern definition.

Applicability.

- It specifies in which situation a design pattern can be applied.

Structure.

- A graphical representation of the classes involved in the pattern.
~~following notations of~~
Object modeling Technique.

Participants.

- The classes and objects participating in the pattern and responsibilities.

⇒ Collaborations :

- How the classes and objects collaborate to carry out responsibilities.

⇒ Consequences :

- Tell us what are the costs and benefits of using the pattern.
- Also tell what part of the system changed independently.

⇒ Implementation :

- Specifies what techniques we should be aware of when implementing the pattern.

⇒ Sample code :

- code fragments that tell us how to implement the pattern in java

⇒ Known users :

- Usage of patterns in the real world

⇒ Related patterns :

- Specifies which other patterns related this pattern.

How to select design pattern ? Explain in brief.

Consider how design patterns solve design problems.

- Considering how design patterns to find appropriate objects, determine object granularity, specify objects in which design patterns solve the problems.

Scan intent sections:

- Looking at the intent section of each design pattern specification lets us choose the appropriate design pattern.

study how patterns interrelate.

- The relationships b/w the patterns will direct us to choose the right pattern or group of patterns.

study patterns of like purpose.

- This will give you an insight into the similarities and

differences b/w patterns of like purpose.

⇒ Examine a cause of redesign.

- Look at your problem and identify if there are any causes of redesign.

- Then look at the catalog of patterns that will help you avoid the causes of redesign.

⇒ Consider what should be variable in your design.

- consider what you want to be able to change without redesign.