DESIGN PATTERN

DEFINATION :

- In software development, a pattern (or design pattern) is a written document that describes a general solution to a design problem that recurs repeatedly in many projects. Software designers adapt the pattern solution to their specific project. Patterns use a formal approach to describing a design problem, its proposed solution, and any other factors that might affect the problem or the solution.

DESIGN PATTERN IN OOP :

- In object-oriented programming, a pattern can contain the description of certain objects and object classes to be used, along with their attributes and dependencies, and the general approach to how to solve the problem.
- Often, programmers can use more than one pattern to address a specific problem.
- A collection of patterns is called a pattern framework.

DESIGN PATTERN INCLUDE THE FOLLOWING INFORMATION :

- Name that describes the pattern.
- Problem to be solved by the pattern.
- Context, or settings, in which the problem occurs.
- Forces that could influence the problem or its solution.
- Context for solution.
- Rationale behind the solution.
- Known uses & related patterns.
- Author and date information.
- References & keywords used or searching.
- Sample code related to the solution, if it helps.

TYPES OF DESIGN PATTERN :

- Creational :
    1. Prototype.
    2. Builder.
    3. Singleton.
    4. Factory.

- Structural :
    1. Proxy.
    2. Decorator.
    3. Facade.
    4. Adapter.
    5. Flyweight.

TYPES OF DESIGN PATTERN :

- Behavioural :
    1. Chain of Responsibility.
    2. Iterator.
    3. State.
    4. Strategy.
    5. Observer.
    6. Visitor.
    7. Template Method.
    8. Command.
    9. Memento.
    10. Mediator.

DIFFERENT DESIGN PATTERNS :

- Creational Design pattern :

  The creational design pattern build with creation a object, how can you creats object, how can you efficiently creats object.

- Behavioural Design pattern :

  The behavioural design pattern discuss the different objects and how they interact with each other, about how an object talks with another object.

- Structural Design pattern :

  The structural design pattern are how the objects are composed, if there a inheritance relationship between them. One object present in another object.

SINGLETON DESIGN PATTERN (CREATIONAL) :

- A class of which only a single instance can exist.
- Example :
    1. President of a country.
    2. java.lang.system.
- Things to remember :
    1. Private constructor.
    2. Enum – Effective java.
    3. Difficult to unit test.
- Spring (by default).

PROXY DESIGN PATTERN (STRUCTURAL) :

- An object representing another object.

- Example :

  Credit card is a proxy for what is in our bank account.

There are two types of objects.

1. Remote object : Remote object represents class with others.
2. Home object access the proxy.

CHAIN OF RESPONSIBILITY (BEHAVIOURAL) PATTERN :

- A way of passing a request between a chain of object.
- Example :
    1. Loan or leave approval process.
    2. Exception handling in java.
    3. java.util.logging.Logger#log()

Handler

Targets[]=logger.getHandlers();

THANK YOU