Design Patterns In java

Design Patterns help developers to create maintainable , flexible and understandable code.

A design pattern is a generic repeatable solution to common problems that arise during the design and development of software.

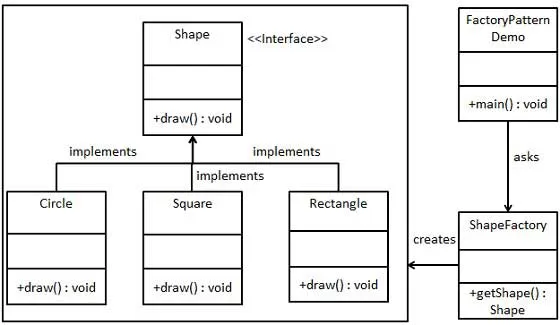
Types of design pattern in java

1. Creational
2. Structural
3. Behavioral
4. Creational design patterns - Creational design patterns are a subset of design patterns in software development. They deal with the process of object creation, trying to make it more flexible and efficient. It makes the system independent and how its objects are created, composed, and represented.

Types of creational design pattern

1. Factory method - Factory Method is a creational design pattern that provides an interface for creating objects in superclass, but subclasses are responsible to create the instance of the class.

Factory Method Design Pattern defines an interface for creating an object, but let subclasses decide which class to instantiate. Factory Method lets a class defer instantiation to subclass.

Q1. 

Code :

import java.util.\*;

//product

Interface Shape{

Void getArea();

}

Class Circle implements Shape {

@override

Public static void getArea(int radius){

Int area= math.pi\*radius\*radius;

System.out.println(“Area of circle “ + area);

}

}

Class Square implements Shape {

@override

Public static void getArea(int side){

Int area= side\*side;

System.out.println(“Area of circle “ + area);

}

}

Class Rectangle implements Shape {

@override

Public static void getArea(int length , int width){

Int area= length\*width;

System.out.println(“Area of circle “ + area);

}

}

//creator

Abstract class ShapeFactory{

Public Shape factoryMethod();

}

Class CircleArea extends Shape {

@override

Public Shape factoryMethod(){

return new Circle();

}

}

Class SquareArea extends Shape {

@override

Public Shape factoryMethod(){

return new Square();

}

}

Class RectangleArea extends Shape {

@override

Public Shape factoryMethod(){

return new Rectangle();

}

}

Public class FactoryDesign{

Public static void main(String[] args){

Circle circle = new CircleArea();

Shape s1 = circle.factoryMethod();

Square square = new SquareArea();

Shape s2 = square.factoryMethod();

Rectangle rectangle = new RectangleArea();

Shape s3 = rectangle.factoryMethod();

}

}