Factory Design Pattern

Through Maya Mnaizel

- Is a creational design pattern that provides an interface for creating objects in a superclass, but allows subclasses to alter the type of objects that will be created
- It involves defining an interface (or abstract class) for creating objects, and concrete classes that implement this interface to create specific types of objects
- The goal is to delegate the responsibility of instantiating objects to a separate class, promoting loose coupling between client code and the actual class implementation

Key components of the Factory pattern:

1. Product interface or abstract class
2. Concrete products: These are the classes that implement the product interface, representing different types of object
3. Factory interface or abstract class: This declares the method for creating objects (the factory method)
4. Concrete factories: The classes that implement the factory interface, providing the implementation for creating specific types of objects

```
//Product Interface
//represents the product that concrete factories will create
public interface Shape {
  void draw():
}

//Concrete products
//implements the interface 'Shape'
public class Circle implements Shape {
  @Override
  public void draw(){
    System.out.println("Drawing a circle");
  }
}

public class Square implements Shape{
  @Override
  public void draw(){
    System.out.println("Drawing a square");
  }
}

//Concrete interface declares the factory method
public interface ShapeFactory{
  Shape createShape();
}

//Concrete Factories implement the factory interface
public class CircleFactory implements ShapeFactory{
  @Override
  public Shape createShape(){
    return new Circle();
  }
}

public class SquareFactory implements ShapeFactory{
  @Override
  public Shape createShape(){
    return new Square();
  }
}


//Client code
public class Client {
  public static void main(String[] args){
    //Create a circle
    ShapeFactory circleFactory = new CircleFactory();
    Shape circle = circleFactory.createShape();
    circle.draw();

    //Create a square
    ShapeFactory squareFactory = new SquareFactory();
    Shape square = squareFactory.createShape();
    square.draw();
```

```
    }
}
```