10/05/2024

Prototype Design Pattern

Prototype Pattern says that **cloning of an existing object instead of creating new one and can also be customized as per the requirement**.

This pattern should be followed, if the cost of creating a new object is expensive and resource intensive.

Advantage of Prototype Pattern

The main advantages of prototype pattern are as follows:

- It reduces the need of sub-classing.
- It hides complexities of creating objects.
- The clients can get new objects without knowing which type of object it will be.
- It lets you add or remove objects at runtime.

Usage of Prototype Pattern

- When the classes are instantiated at runtime.
- When the cost of creating an object is expensive or complicated.
- When you want to keep the number of classes in an application minimum.
- When the client application needs to be unaware of object creation and representation.

Example-1

```
class Bike implements cloneable
{
private int gears;
private String biktype;
Private String model;
Bike()
{
bike type = "Standard";
model = "Leopard";
gears = 4;
```

```
}
public Bike clone()
{
return new Bike();
}
void makeAdvancd()
{
bike type = "Advance";
model ="jaguar";
gears = 6;
}
String getModel()
{
return model;
}
public String toString()
{
return "Bike[gears"+gears+",biketype="+biketype+,model+"model+"];
}
}
public static void main(String[] args)
{
Bike bike = new Bike();
Bike basicBike = bike.clone();
prototypeTest pt = new PrototypeTest();
Bike AdvancedBike = pt.makeJaguar(basicBike);
System.out.println("Prototype Design Pattern Test-1:"+advancedBike.getModel());
System.out.println("prototype Design Pattern Test-1:"+AdvancedBike.toString());
}
}
```