

BUILDER DESIGN PATTERN

Home :

```
public class Home {  
    public String floor;  
    public String walls;  
    public String terrace;  
}
```

Builder Interface :

```
public interface Builder {  
    void buildFloor();  
    void buildWalls();  
    void buildTerrace();  
    Home getComplexHomeObject();  
}
```

Earth Quake Resistant Builder :

```
public class EarthQuakeResistantBuilder implements Builder{  
  
    private Home earthQuakeResistantHome = new Home();  
  
    @Override  
    public void buildFloor() {  
        this.earthQuakeResistantHome.floor="wooden floor";  
    }  
  
    @Override  
    public void buildWalls() {  
        this.earthQuakeResistantHome.walls="wooden walls";  
    }  
  
    @Override  
    public void buildTerrace() {  
        this.earthQuakeResistantHome.terrace="wooden terrace";  
    }  
  
    @Override  
    public Home getComplexHomeObject() {  
        return this.earthQuakeResistantHome;  
    }  
}
```

Flood Resistant Builder :

```

public class FloorResistantBuilder implements Builder{

    private Home floodResistantHome = new Home();

    @Override
    public void buildFloor() {
        floodResistantHome.floor= "10 feets above ground level floor";
    }

    @Override
    public void buildWalls() {
        floodResistantHome.walls= "water resistant walls";
    }

    @Override
    public void buildTerrace() {
        floodResistantHome.terrace= "water leakage resistant terrace";
    }

    @Override
    public Home getComplexHomeObject() {
        return this.floodResistantHome;
    }

}

```

Director :

```

public class Director {
    private Builder builder;

    public Director(Builder builderType) {
        this.builder = builderType;
    }

    public Home getComplexObjectOfHome() {
        return this.builder.getComplexHomeObject();
    }

    public void manageRequiredHomeConstruction(){
        this.builder.buildFloor();
        this.builder.buildWalls();
        this.builder.buildTerrace();
    }

}

```

Main Class / Client :

```

public class Client {
    public static void main(String[] args) {
        // create object of required home builder
        EarthQuakeResistantBuilder earthQuakeResistantBuilder = new
EarthQuakeResistantBuilder();

        // create object of director that will keep an eye on your builder

```

```
Director director = new Director(earthQuakeResistantBuilder);

director.manageRequiredHomeConstruction();
Home homeConstructedAtTheEnd = director.getComplexObjectOfHome();

System.out.println(homeConstructedAtTheEnd);
System.out.println(homeConstructedAtTheEnd.floor);

    }
}
```