Design Pattern - Builder

Intent

Builder lets you construct complex objects step by step. The pattern allows you to produce different types and representations of an object using the same construction code.

```
b = new ConcreteBuilder1()
d = new Director(b)
d.Make()
Product1 p = b.GetResult()
```

Client

Director

builder: Builder

Director(builder)
ChangeBuilder(builder)
Make(type)○

```
builder.reset()
if(type == "simple")
     builder.BuildStepA()
else
     builder.BuildStepB()
     builder.BuildStepC()
```

<<Interface>>
Builder

Reset()
BuildStepA()
BuildStepB()
BuildStepC()

ConcreteBuilder1

result: Product1

Reset()
BuildStepA()
BuildStepB()
BuildStepC()
GetResult(): Product1

ConcreteBuilder2

result: Product2

Reset()○
BuildStepA()
BuildStepB()○
BuildStepC()
GetResult(): Product2 ○

```
result = new Product2()
```

```
result.AddFeatureB()
```

```
return this.result
```

Product1

Product2