Builder Design Pattern Example

**IBuilder.java**

public interface IBuilder {  
 void buildBody();  
 Product getProduct();  
}

**ConcreteBuilder.java**

public class ConcreteBuilder implements IBuilder {  
 private Product product = new Product();  
  
 @Override  
 public void buildBody() {  
 System.*out*.println("... body building ...");  
 product.setName("Prod Name 1");  
 }  
  
 @Override  
 public Product getProduct() {  
 return product;  
 }  
}

**Director.java**

public class Director {  
  
 private IBuilder builder = null;  
  
 public Director(IBuilder builder ) {  
 this.builder = builder;  
 }  
  
 public void build() {  
 builder.buildBody();  
 }  
  
 public Product getProduct() {  
 return builder.getProduct();  
 }  
}

**Product.java**

public class Product {  
 private String name;  
 private String type;  
  
 public String getName() {  
 return name;  
 }  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getType() {  
 return type;  
 }  
  
 public void setType(String type) {  
 this.type = type;  
 }  
}

**Test.java**

public class Test {  
  
 public static void main(String[] args) {  
 Director director = new Director( new ConcreteBuilder());  
 director.build();  
 Product product = director.getProduct();  
 System.*out*.println("Product Name :::"+product.getName());  
  
 }  
}

The generated class diagram is given below.

