## Title: Refactoring: Encapsulate Classes with Factory

Clients directly instantiate classes that reside in one package and implement a common interface.

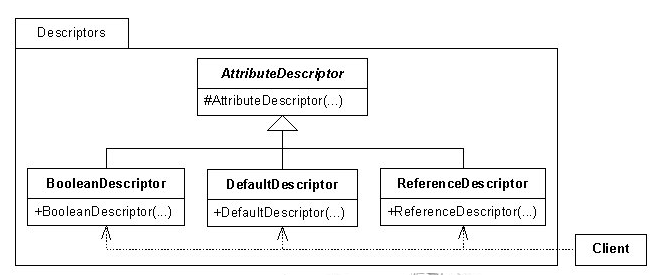
Make the class constructors non-public and let clients create instances of them using a Factory.

Motivation to refactor:

A client’s ability to directly instantiate classes is useful so long as the client needs to know about the very existence of those classes. But what if the client doesn’t need that knowledge? What if the classes live in one package and implement one interface and those conditions aren’t likely to change? In that case, the classes in the package could be hidden from clients outside the package by giving a Factory the responsibility of creating and returning instances that implement a common interface.

Pattern: The **Factory pattern** is helpful in this context. It uses one class to encapsulate both creation logic and a client’s instantiation/configuration preferences

**Sample :**



Sample of code:

package descriptors;

public abstract class AttributeDescriptor...

   protected AttributeDescriptor(...)

public class BooleanDescriptor extends AttributeDescriptor...

public BooleanDescriptor(...) {

super(...);

}

public class DefaultDescriptor extends AttributeDescriptor...

public DefaultDescriptor(...) {

super(...);

}

public class ReferenceDescriptor extends AttributeDescriptor...

public ReferenceDescriptor(...) {

super(...);

}