```
interface MessageStrategy {
   public void sendMessage();
}
abstract class AbstractStrategyFactory {
   public abstract MessageStrategy createStrategy(MessageBody mb);
}

class MessageBody {
   Object payload;
   public Object getPayload() {
      return payload;
   }
   public void configure(Object obj) {
      payload = obj;
   }
   public void send(MessageStrategy ms) {
      ms.sendMessage();
   }
}
class DefaultFactory extends AbstractStrategyFactory {
   private DefaultFactory() {
   }
   static DefaultFactory instance;
   public static AbstractStrategyFactory getInstance() {
      if (instance == null)
         instance = new DefaultFactory();
      return instance;
   }
   public MessageStrategy createStrategy(final MessageBody mb) {
      return new MessageStrategy() {
         MessageBody body = mb;
         public void sendMessage() {
            Object obj = body.getPayload();
            System.out.println(obj);
         }
      };
   }
}
public class HelloWorld {
   public static void main(String[] args) {
      MessageBody mb = new MessageBody();
      mb.configure("Hello World!");
      AbstractStrategyFactory asf = DefaultFactory.getInstance();
      MessageStrategy strategy = asf.createStrategy(mb);
      mb.send(strategy);
   }
}
```