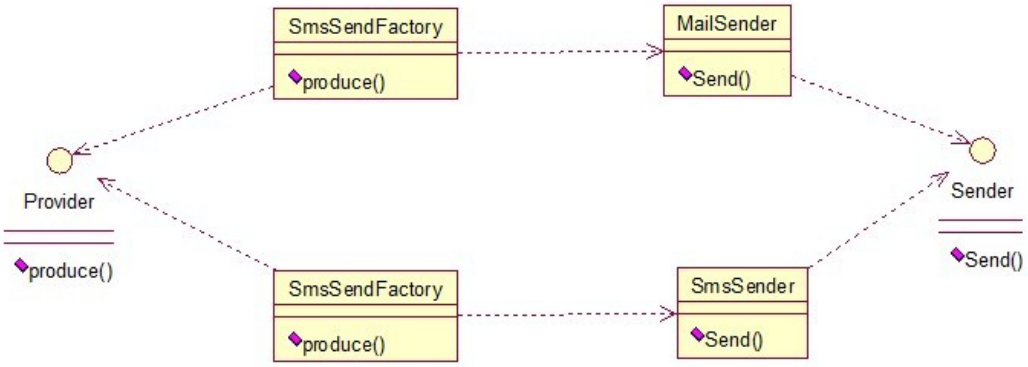
工厂方法模式有一个问题就是，类的创建依赖工厂类，也就是说，如果想要拓展程序，必须对工厂类进行修改，这违背了闭包原则，所以，从设计角度考虑，有一定的问题，如何解决？就用到抽象工厂模式，创建多个工厂类，这样一旦需要增加新的功能，直接增加新的工厂类就可以了，不需要修改之前的代码。因为抽象工厂不太好理解，我们先看看图，然后就和代码，就比较容易理解。



请看例子：

[java] view plaincopy

```
public interface Sender {
    public void Send();
}
```

两个实现类：

[java] view plaincopy

```
public class MailSender implements Sender {
    @Override
    public void Send() {
        System.out.println("this is mailsender!");
    }
}
```

[java] view plaincopy

```
public class SmsSender implements Sender {

    @Override
```

```
    public void Send() {
        System.out.println("this is sms sender!");
    }
}
```

两个工厂类：

[java] view plaincopy

```
public class SendMailFactory implements Provider {

    @Override
    public Sender produce(){
        return new MailSender();
    }
}
```

[java] view plaincopy

```
public class SendSmsFactory implements Provider{

    @Override
    public Sender produce() {
        return new SmsSender();
    }
}
```

在提供一个接口：

[java] view plaincopy

```
public interface Provider {
    public Sender produce();
}
```

测试类：

[java] view plaincopy

```
public class Test {

    public static void main(String[] args) {
```

```
            Provider provider = new SendMailFactory();
            Sender sender = provider.produce();
            sender.Send();
        }
    }
```

其实这个模式的好处就是，如果你现在想增加一个功能：发及时信息，则只需做一个实现类，实现Sender接口，同时做一个工厂类，实现Provider接口，就OK了，无需去改动现成的代码。这样做，拓展性较好！