阅读Abstract Factory的例子的代码，举例说明使用Abstract Factory模式的其他应用

答：比如开发一个跨平台的界面系统，支持 Windows 和 Linux 两种操作系统的界面组件。每个系统包含两种组件：按钮（Button）和文本框（TextField）。使用抽象工厂模式可以将不同系统的组件家族（产品族）封装，客户端通过抽象接口创建相关组件，无需关心具体实现。

Code：

// 抽象按钮接口（产品等级结构1）

interface Button {

void click(); // 按钮点击行为

}

// 抽象文本框接口（产品等级结构2）

interface TextField {

void displayText(String text); // 显示文本

}

// Windows按钮（具体产品，属于Windows产品族）

class WindowsButton implements Button {

@Override

public void click() {

System.out.println("Windows按钮被点击（方形风格）");

}

}

// Linux按钮（具体产品，属于Linux产品族）

class LinuxButton implements Button {

@Override

public void click() {

System.out.println("Linux按钮被点击（圆形风格）");

}

}

// Windows文本框（具体产品，属于Windows产品族）

class WindowsTextField implements TextField {

@Override

public void displayText(String text) {

System.out.println("Windows文本框：" + text);

}

}

// Linux文本框（具体产品，属于Linux产品族）

class LinuxTextField implements TextField {

@Override

public void displayText(String text) {

System.out.println("Linux文本框：" + text);

}

}

// 抽象工厂：定义创建产品族的接口

interface UIFactory {

Button createButton(); // 创建按钮

TextField createTextField(); // 创建文本框

}

// Windows工厂：生产Windows产品族

class WindowsFactory implements UIFactory {

@Override

public Button createButton() {

return new WindowsButton();

}

@Override

public TextField createTextField() {

return new WindowsTextField();

}

}

// Linux工厂：生产Linux产品族

class LinuxFactory implements UIFactory {

@Override

public Button createButton() {

return new LinuxButton();

}

@Override

public TextField createTextField() {

return new LinuxTextField();

}

}

public class Client {

public static void main(String[] args) {

// 使用Windows工厂创建组件

UIFactory windowsFactory = new WindowsFactory();

Button windowsButton = windowsFactory.createButton();

TextField windowsTextField = windowsFactory.createTextField();

// 使用Linux工厂创建组件

UIFactory linuxFactory = new LinuxFactory();

Button linuxButton = linuxFactory.createButton();

TextField linuxTextField = linuxFactory.createTextField();

// 调用组件功能

System.out.println("=== Windows 组件 ===");

windowsButton.click();

windowsTextField.displayText("Hello, Windows!");

System.out.println("\n=== Linux 组件 ===");

linuxButton.click();

linuxTextField.displayText("Hello, Linux!");

}

}