

Patrones de Diseño

...

75.10 - Técnicas de Diseño

Aplicación

Se tiene el siguiente código, correspondiente a una aplicación de diseño gráfico:

```
public class Application {  
  
    public static void main() {  
        WindowsButton button = new WindowsButton();  
  
        WindowsPanel panel = new WindowsPanel();  
        panel.add(button);  
  
        panel.draw();  
    }  
}
```

Requerimiento

Dado el positivo impacto que la aplicación tuvo en el mercado, nos han solicitado no sólo soportar el sistema operativo Windows (que es el soportado actualmente), sino también algunas variantes de Unix y OSX. Las implementaciones de Button y Panel son particulares para el sistema operativo Windows, se desea poder mantener el código core de la aplicación cerrado antes cambios, es decir, ante el agregado del soporte para un nuevo sistema operativo.

Cómo podemos modificar/extender el código anterior para lograr cumplir con este nuevo requerimiento?

1- Crear Abstracciones

```
public class Application {  
  
    public static void main() {  
        Button button = new WindowsButton();  
  
        Panel panel = new WindowsPanel();  
        panel.add(button);  
  
        panel.draw();  
    }  
  
}
```

2- Crear Implementaciones Concretas

```
Button button = new WindowsButton();  
Panel panel = new WindowsPanel();
```

```
Button button = new OSXButton();  
Panel panel = new OSXPanel();
```

3- Abstraer la creación

```
class WindowsControlFactory {  
    public Button createButton() {  
        return new WindowsButton();  
    }  
  
    public Panel createPanel() {  
        return new WindowsPanel();  
    }  
}
```

```
class OSXControlFactory {  
    public Button createButton() {  
        return new OSXButton();  
    }  
  
    public Panel createPanel() {  
        return new OSXPanel();  
    }  
}
```

4- Abstraer la creación

```
class Application {  
  
    public static void main() {  
        WindowsControlFactory windowsFactory = new WindowsControlFactory();  
        OSXControlFactory osxFactory = new OSXControlFactory();  
  
        ....  
  
        Button button = windowsFactory.createButton();  
  
        Panel panel = windowsFactory.createPanel();  
        panel.add(button);  
  
        panel.draw();  
  
        Que factory uso??  
  
    }  
}
```

5-Cuál de las dos utilizo?

Son dos clases concretas que también son dependientes del sistema operativo.

Necesito una **abstracción** de la fábrica.

6-Cuál de las dos utilizo?

```
interface ControlFactory {  
    public Button createButton();  
    public Panel createPanel();  
}
```

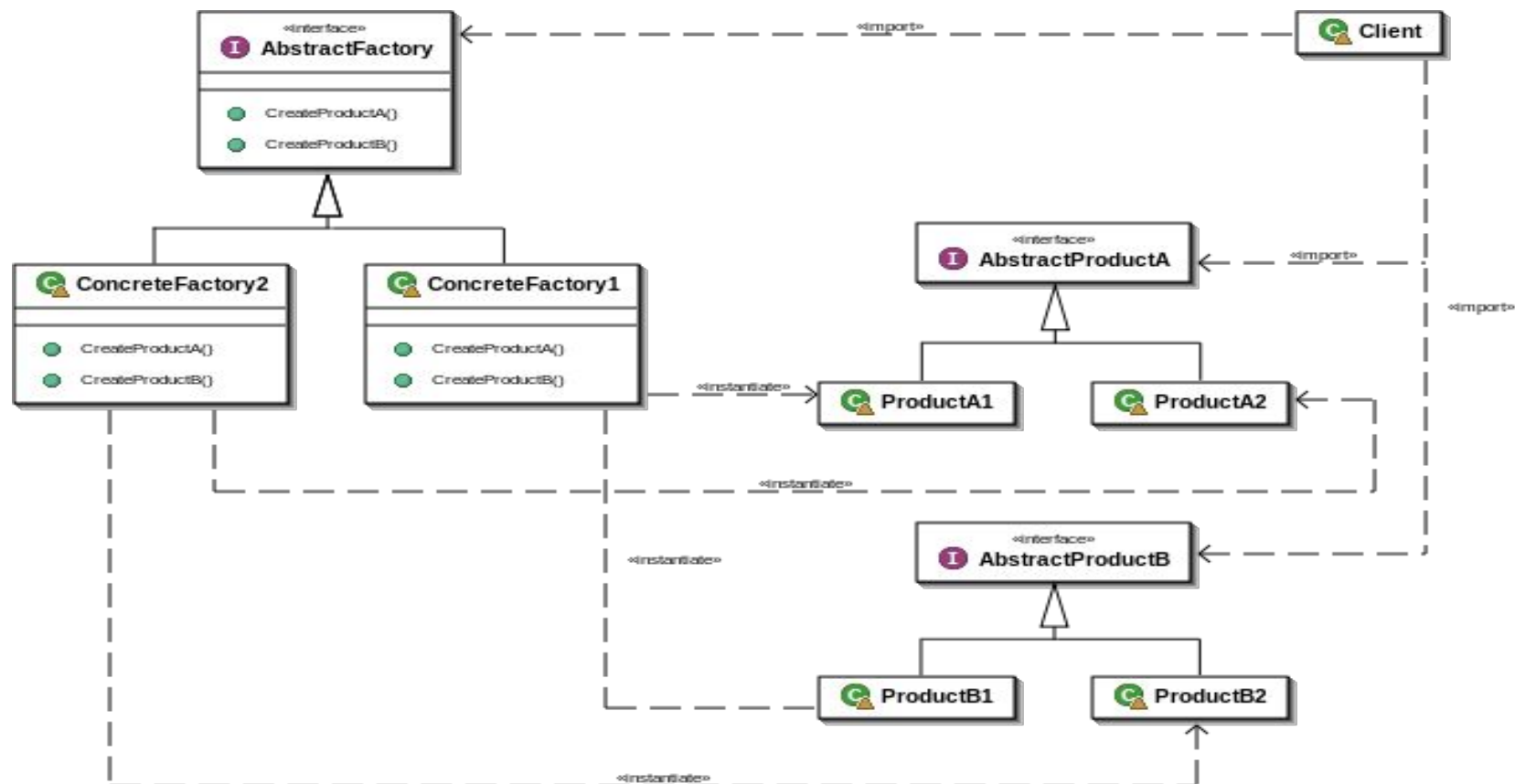
```
class WindowsControlFactory implements ControlFactory { ...  
class OSXControlFactory implements ControlFactory { ...
```

```
class Application {  
    public static void main() {  
        ControlFactory factory = new WindowsControlFactory();  
        ....  
        Button button = factory.createButton();  
        Panel panel = factory.createPanel();  
        panel.add(button);  
  
        panel.draw();  
    }  
}
```

7- Cómo decido cuál utilizar? Dónde?

```
public static void main(final String[] args) {  
  
    // variable de entorno  
    // parametro de configuration  
    ControlFactory factory = new WindowsFactory();  
  
    Application application = new Application(factory);  
  
    ...  
}
```

Abstract Factory



?