当开发人员和领域专家都在使用设计中没有的词汇时，可能已经遗漏了某个对象

当你在重构中遇到某几个类具有相似的行为和参数，那么你可能可以提炼一个超类

**示例**

如下，你如何挖掘概念

// 聚合根

class Component {

    ...

    public string MenuName { get; set; }

public string PageName { get; set; }

public string PageIndex { get; set; }

    public CssSetting CssSetting { get; set; }

}

// 怕有些同学不懂 Action<string> 是一个函数，接收 string 类型的参数，返回 void

// 等价于 (string param) => void

class MenuBox {

    public void SetMenu(string menuName, Action<string> changeMenu){...}

}

class PageBox {

    public void SetPage(string pageName, string pageIndex, Action<string, string> submit){...}

}

class CssBox {

    public void SetCss(CssSetting cssSetting, Action<CssSetting> changeCss) {...}

}

1. 为了减少依赖，所以我们的服务应该依赖于聚合根，重构如下：

class MenuBox {

    public void SetMenu(Component component, Action<Component> changeMenu){...}

}

class PageBox {

    public void SetPage(Component Component, Action<Component> submit){...}

}

class CssBox {

    public void SetCss(Component Component, Action<Component> changeCss) {...}

}

1. 重构后几个服务都具有相似的参数，所以提炼基类

abstract class BaseBox {

    public abstract void SetComponent(Component component, Action<Component> submit);

}

class MenuBox : BaseBox {

    public void SetComponent(Component component, Action<Component> submit){...}

}

class PageBox : BaseBox {

    public void SetComponent(Component Component, Action<Component> submit){...}

}

class CssBox : BaseBox {

    public void SetComponent(Component Component, Action<Component> submit) {...}

}

1. 后续如可重构取决于你的设计（如，几个方法的处理流程相同或相似，那么是否可以取消继承等）