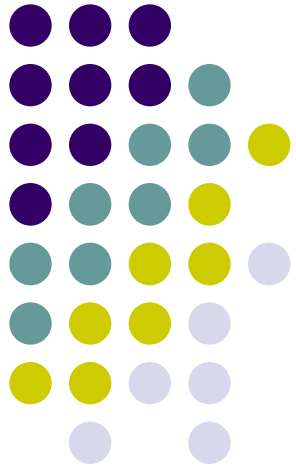
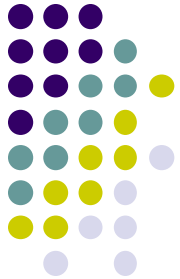
在Eclipse中实施重构

高 原

2010.11.11

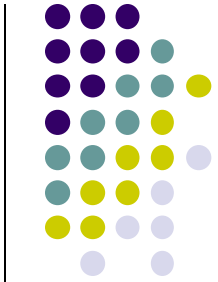




内容概要

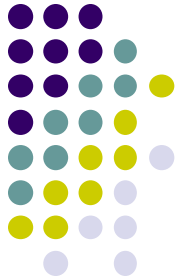
- 重构介绍
- 代码坏味
- Eclipse提供的重构功能

什么是重构



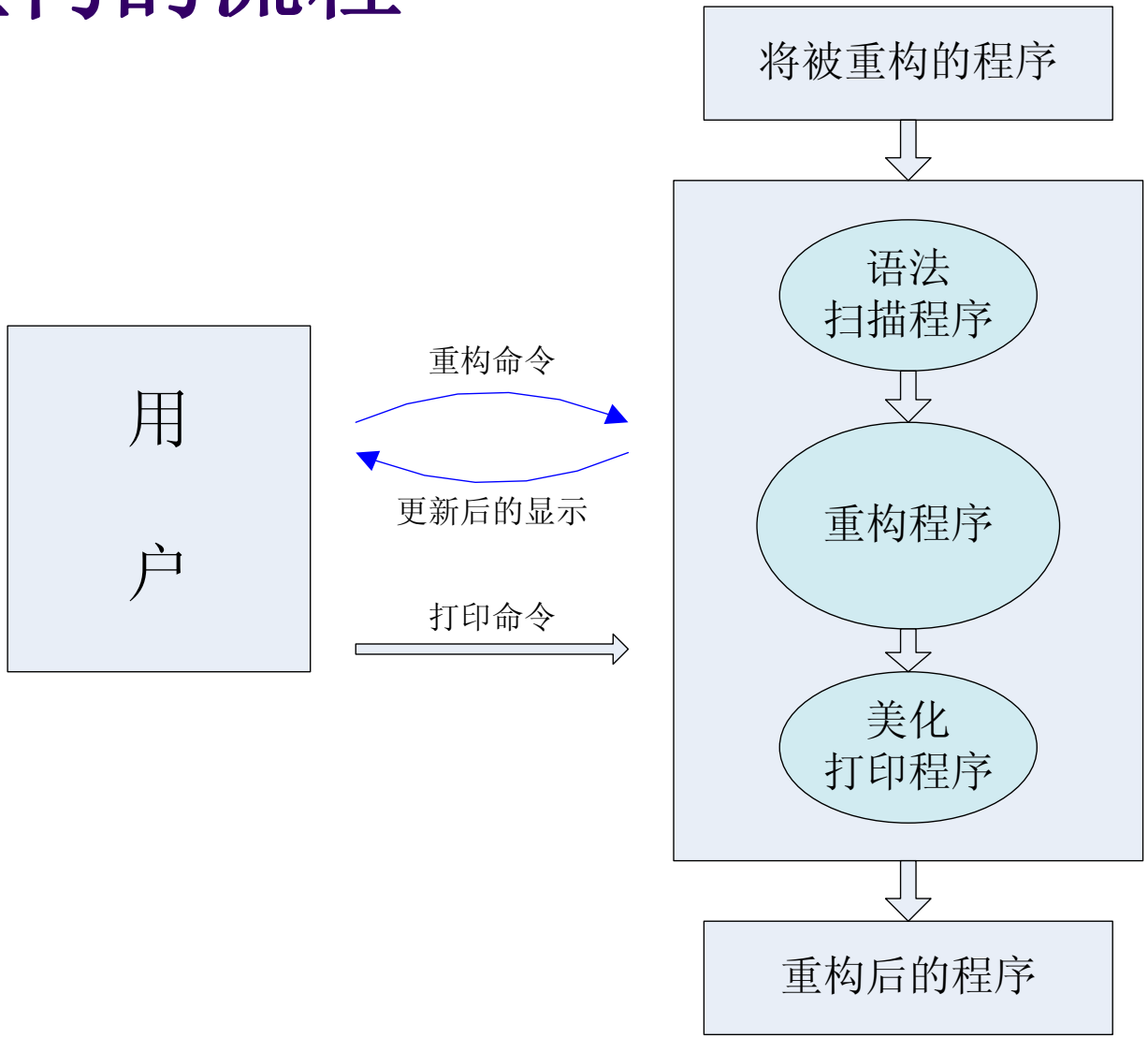
重构：是指在不改变软件任何功能的前提下对代码进行修改，调整其结构，提高其可理解性，降低其修改的成本。

为什么要重构



- 改进软件的设计；
- 使软件更容易被理解；
- 帮助尽早的发现Bugs；
- 提高软件开发速度。

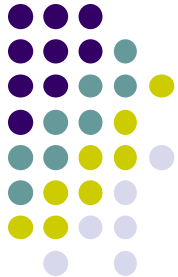
重构的流程



代码坏味

- 重复的代码（Duplicated Code)
- 过长的函数(Long Method)
- 过大类(Large Class)
- 过长的参数列(Long Parameter List)
- 发散式变化(Divergent Change)
- 霰弹式修改(Shotgun Surgery)
- 依恋情结(Feature Envy)
- 数据泥团(Data Clumps)
- 基本型别偏执(Primitive Obsession)
- Switch语句(Swtich Statements)
- 平行继承体系(Parallel Inheritance Hierarchies)
- 冗赘类(Lazy Class)
- 夸夸其谈未来性(Speculative Generality)
- 令人迷惑的暂时值域(Temporary Field)
- 过度遇合的消息链(Message Chains)
- 中间转手人(Middle Man)
- 狎昵关系(Inappropriate Intimacy)
- 异曲同工的类(Alternative Classes with Different Interfaces)
- 不完善的程序库类(Incomplete Library Class)
- 纯粹的数据类(Data Class)
- 被拒绝的遗赠(Refused Bequest)
- 过多的注释(Comments)

Eclipse提供的重构功能

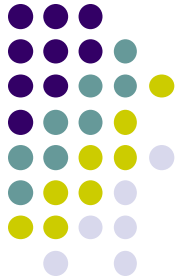


- 重构是软件开发过程中保证代码质量非常重要的手段，而手动进行重构代码的话，很容易引入一些低级错误（例如，单词拼写错误），从而导致浪费大量不必要的时间。Eclipse为重构提供了很强大的支持，很大程度上用户不必为重构的笔误而再烦恼。

重构功能分类

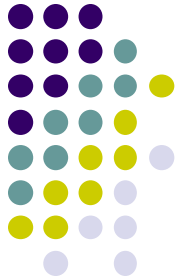
- 结构性重构
- 类级别重构
- 类内部重构

结构性重构



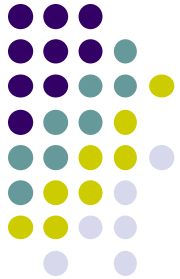
- Rename
- Move
- Change Method signature
- Convert Anonymous Class to Nested
- Convert Member Type to Top Level

类级别重构



- Push Down
- Push Up
- Extract Interface
- Use Supertype Where Possible

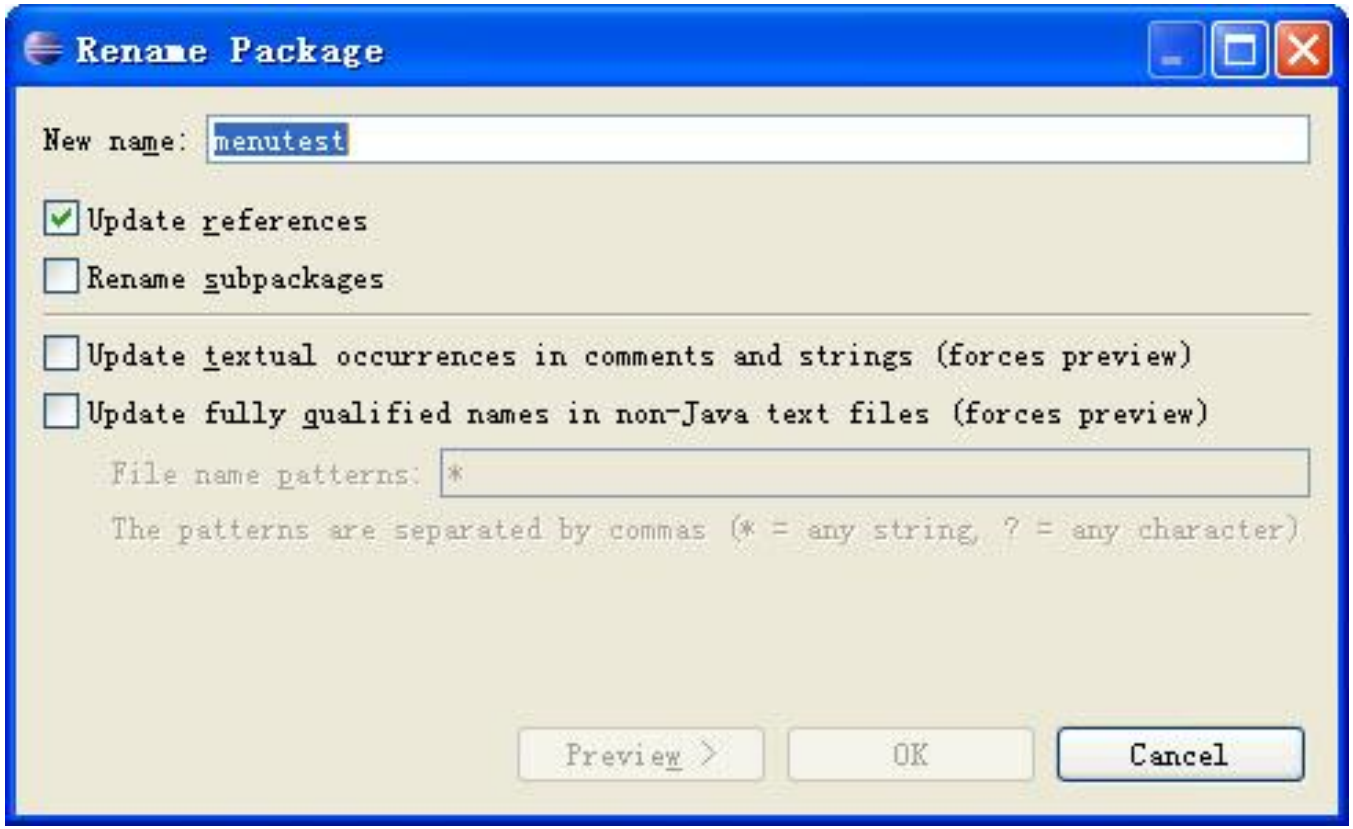
类内部重构



- Inline
- Extract Method
- Extract Local Variable
- Extract Constant
- Introduce Parameter
- Introduce Factory
- Convert Local Variable to Field
- Encapsulate Field

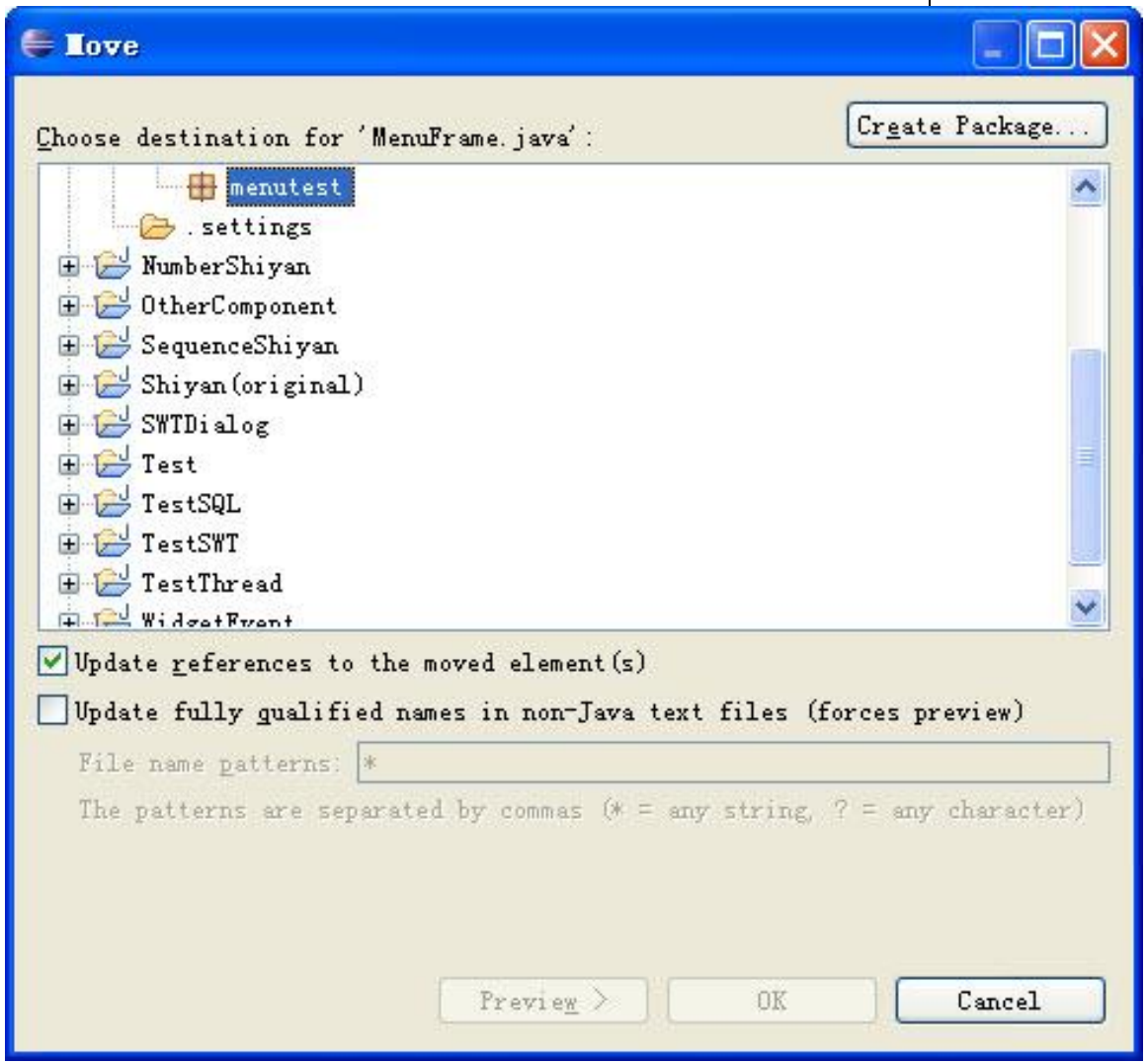
Rename

- Rename重构的功能就是重命名Java元素。（包括包名、文件名、类名、方法名、变量名等）



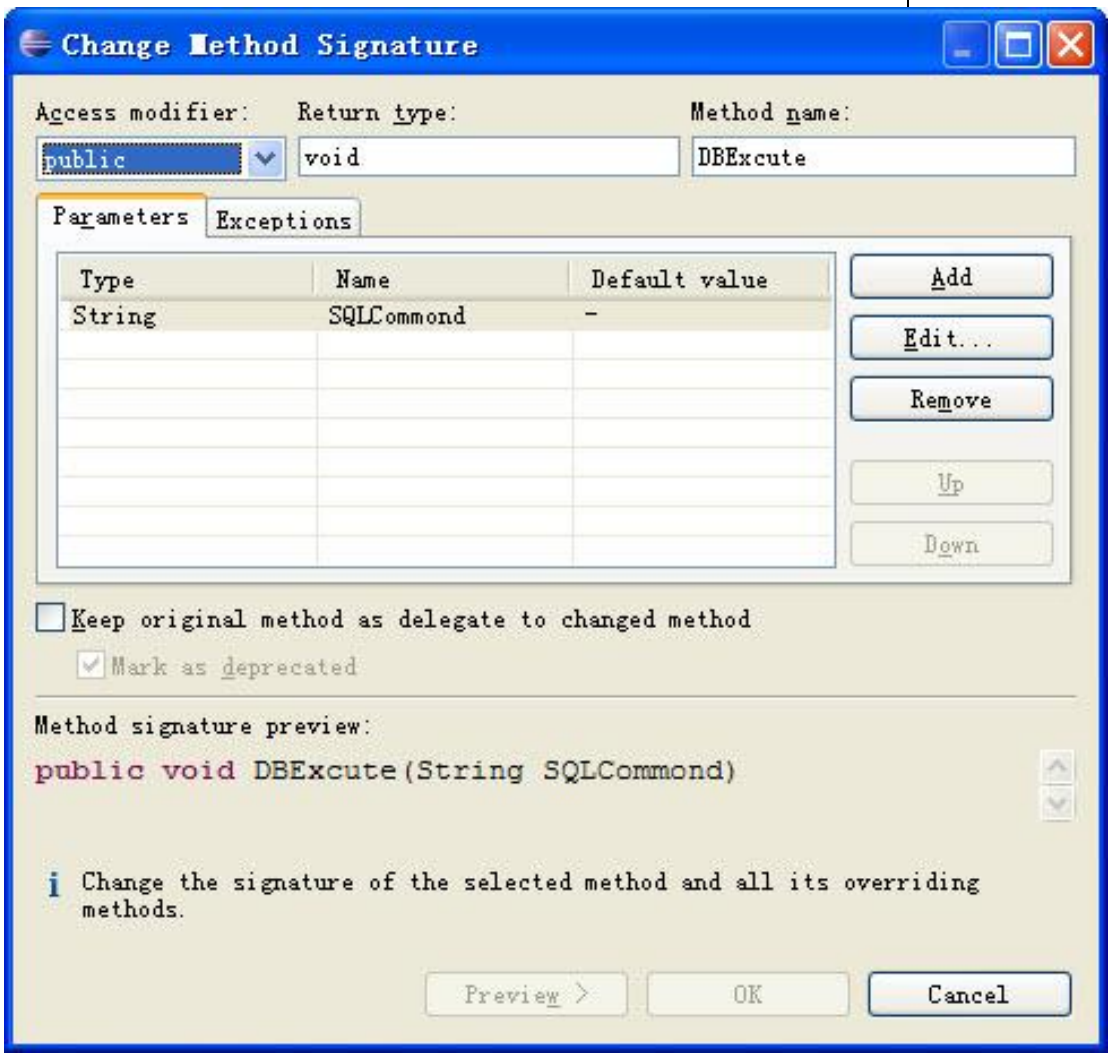
Move

- Move的重构功能可以把一个Java元素从一个地方移动到另一个地方，Move的重构主要用来移动一个类到不同的包下。



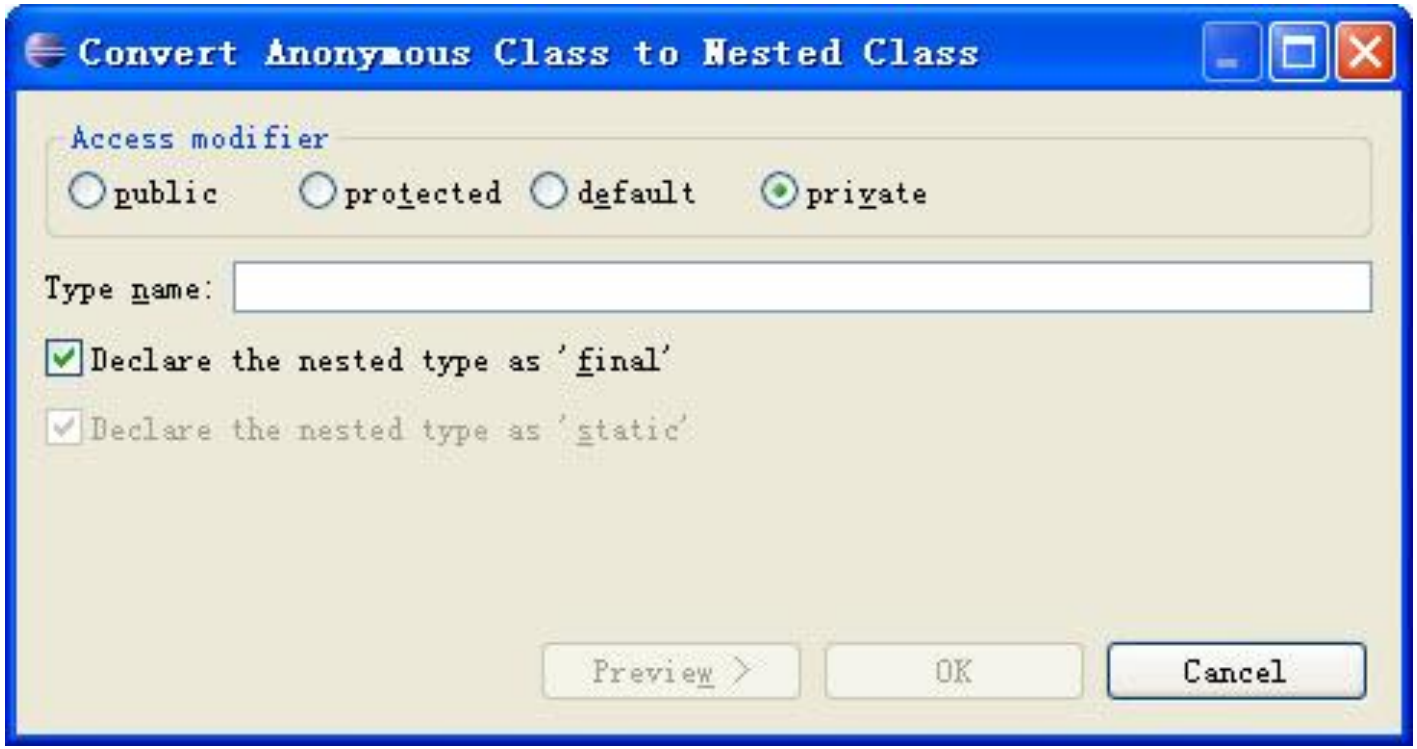
Change Method signature

- “Change Method Signature”重构的功能是改变方法的定义，例如改变方法的参数名称、类型和个数、返回值的类型，方法的可见性以及方法的名称等。

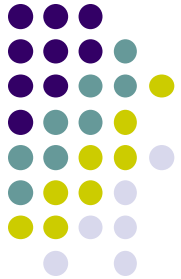


Convert Anonymous Class to Nested

- Convert Anonymous Class to Nested重构的功能是把匿名类改成内部类，这样同一个类的其它部分也可以共享此类了。



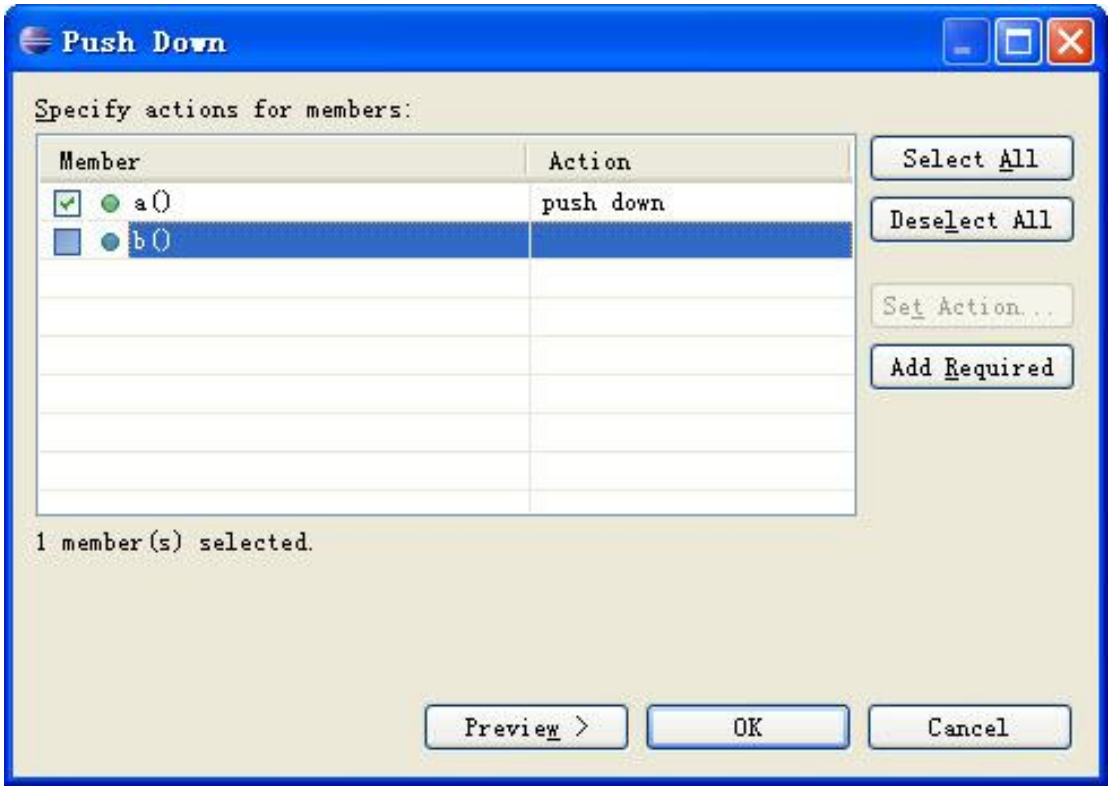
Convert Member Type to Top Level



- 通过Move Member Type to Top Level的重构方式，可以把内部类改成非内部类，并且重新创建一个新的文件，这样其它的类就可以共享此类。

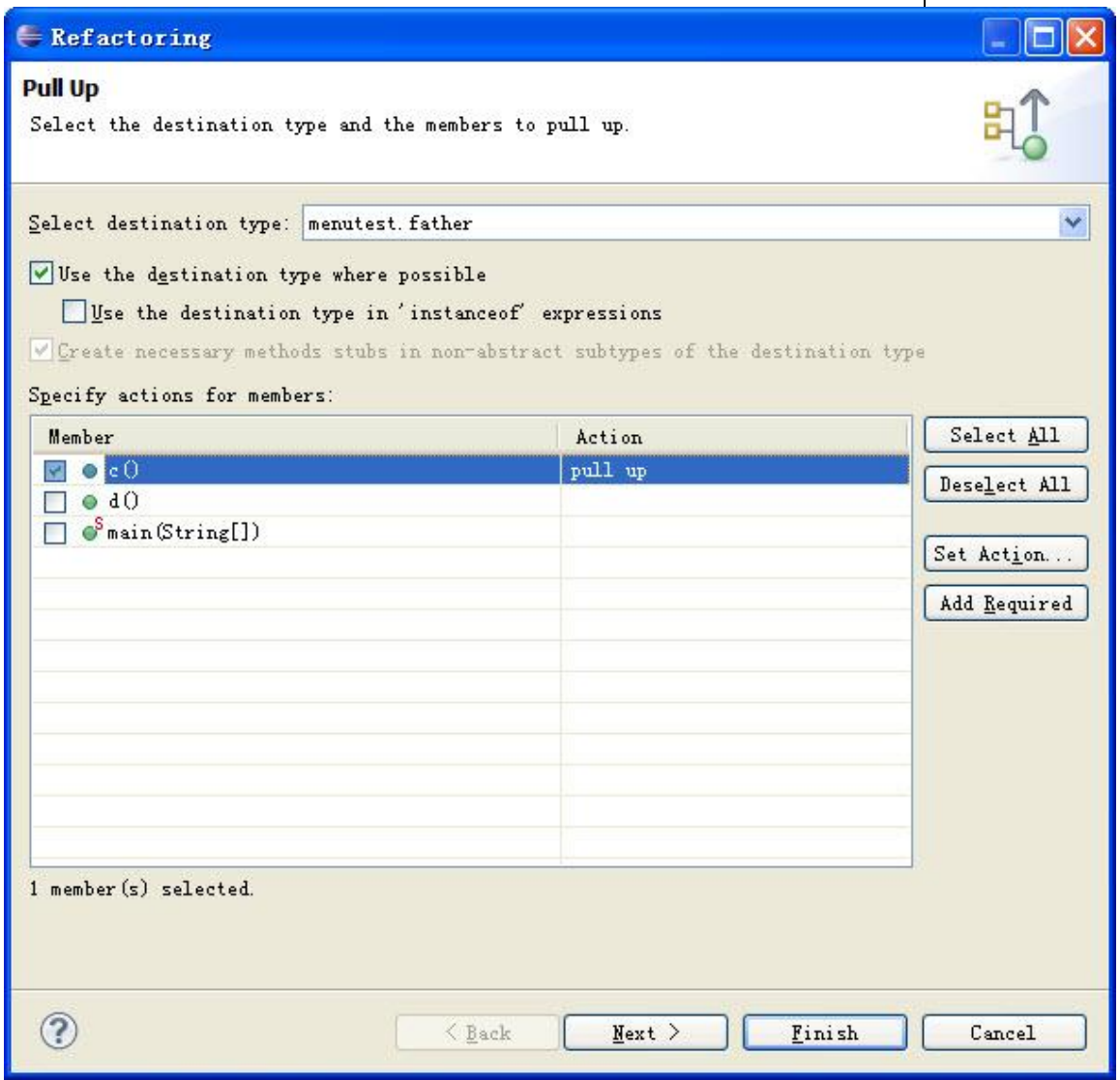
Push Down

- Push Down重构功能是把父类的方法和属性移动到所有的子类中，父类的方法可以选择性的保留抽象方法。



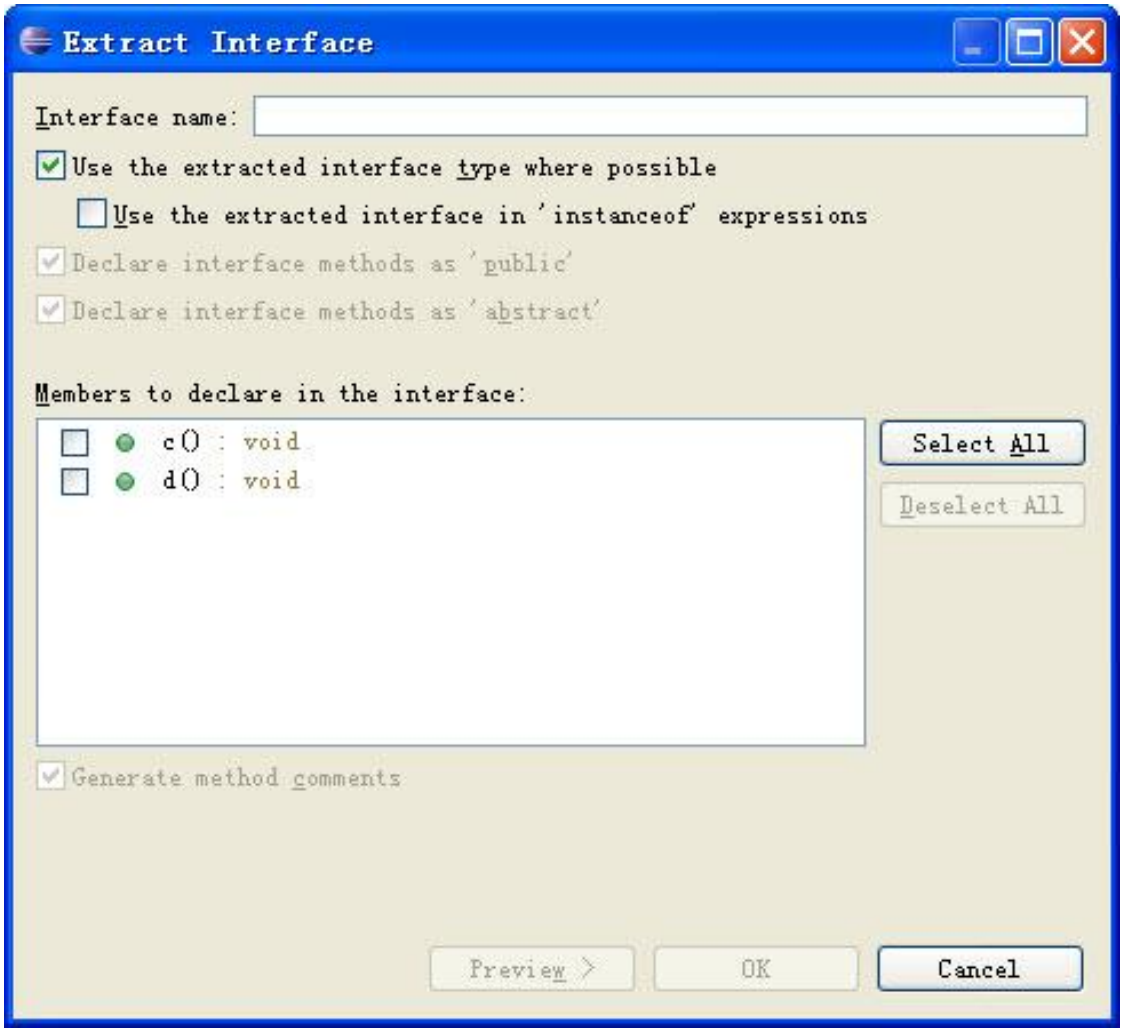
Pull Up

- Pull Up重构和Push Down重构正好相反，它的作用是把方法和属性移动到其父类中去。

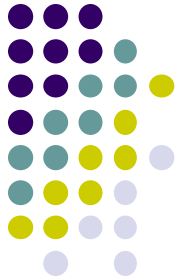


Extract Interface

- Extract Interface 重构能够从一个已存在的类中提取接口，它可以从某个类中选择方法，把这些方法提取到一个单独的接口中。



Use Supertype Where Possible



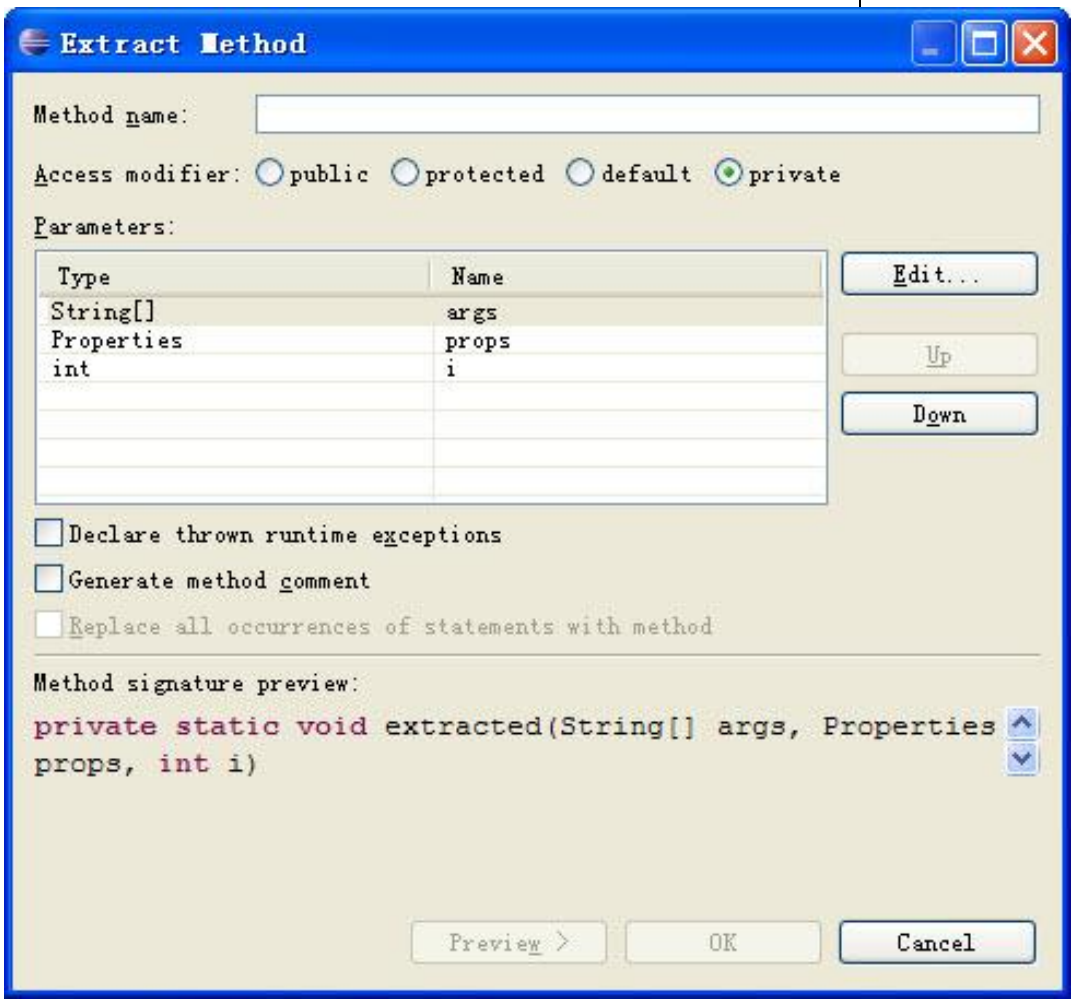
- Use Supertype Where Possible重构能够用某一个类的父类的类型替换当前类的类型。

Inline

- Inline重构能用函数的内容替换掉函数的引用，或者将类的内容转移到另一个类当中，并删除此类。

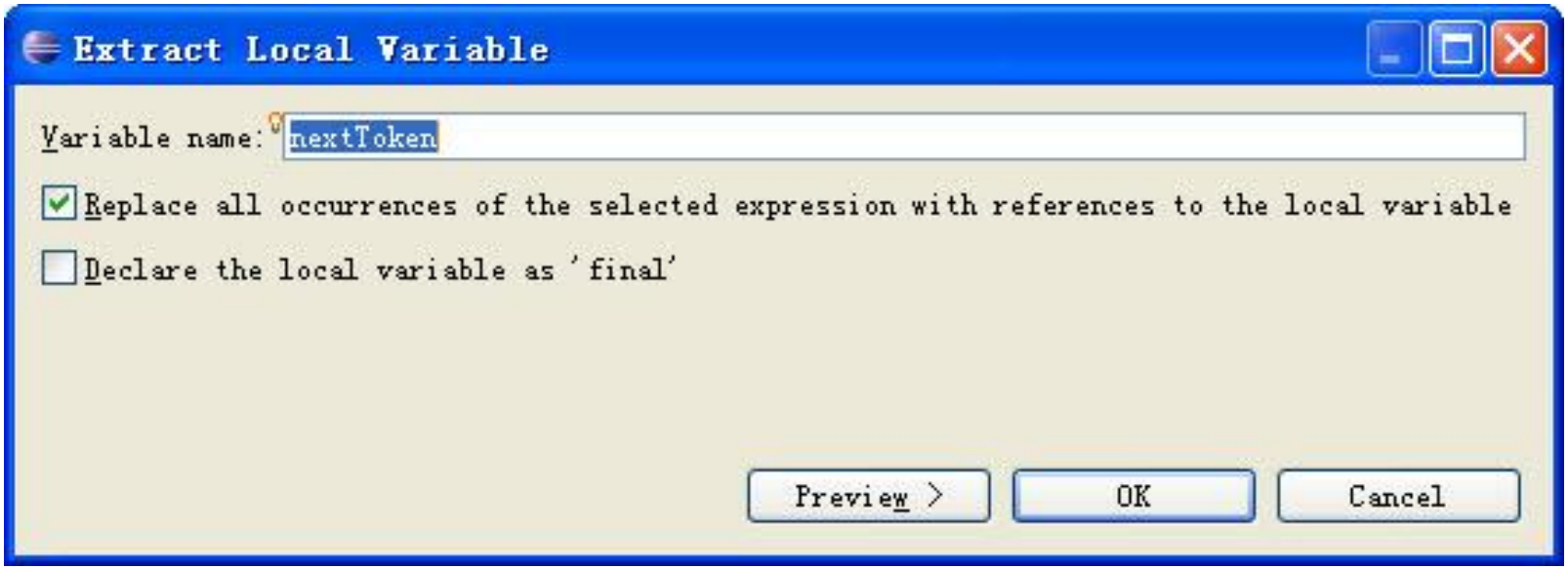
Extract Method

- Extract Method重构和Inline重构相反，它能够从冗长的方法中提取小的方法，把大的方法分解成多个小方法来实现，通过此重构能够使代码看上去更简单漂亮，也很大程度上提高代码的复用性。



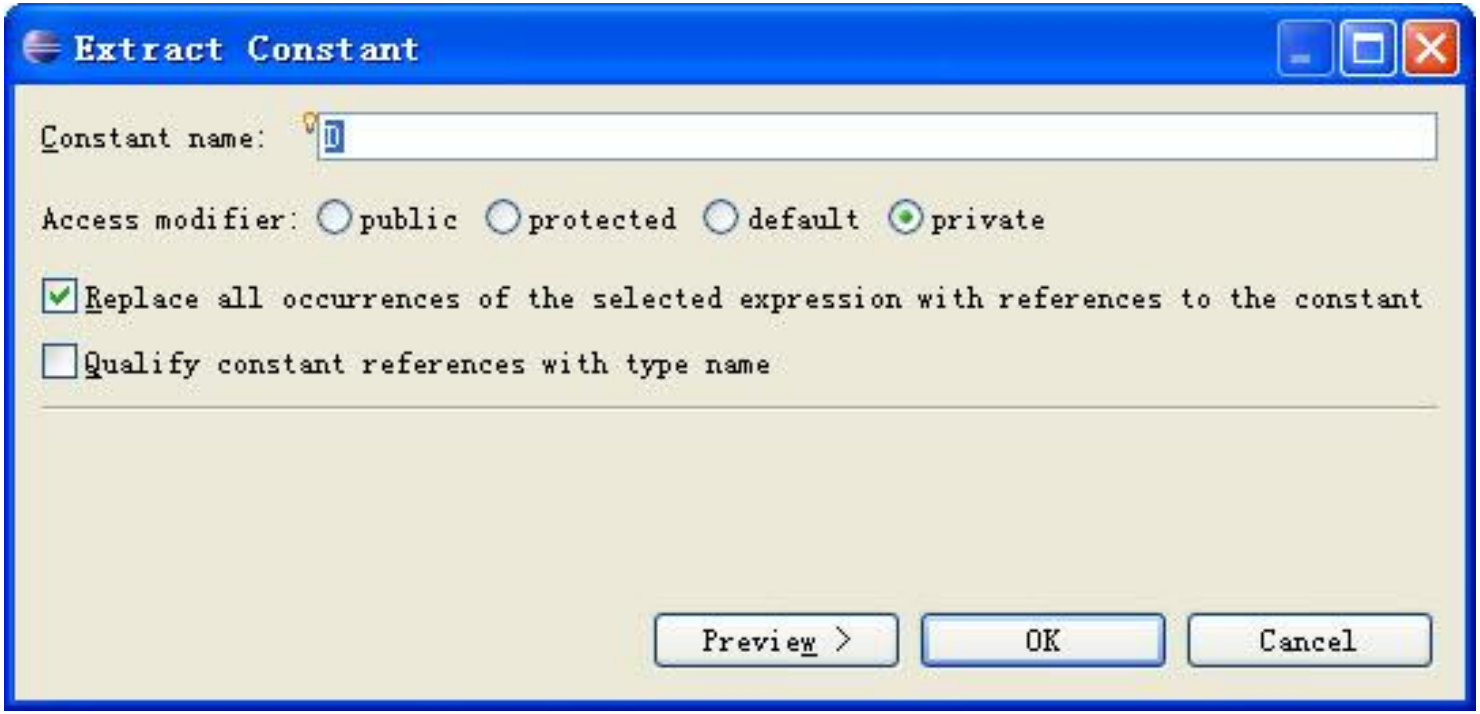
Extract Local Variable

- Extract Local Variable 重构取出一段被直接使用的表达式，然后将这个表达式首先赋值给一个局部变量。然后在原先使用那个表达式的地方使用这个变量。

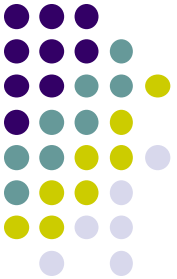


Extract Constant

- Extract Constant 与 Extract Local Variable 相似，但是您必须选择静态常量表达式，重构工具将会把它转换成静态的 final 常量。

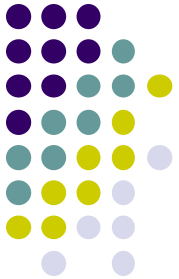


Introduce Parameter



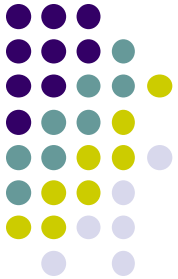
- Introduce Parameter重构可以通过函数中的表达式、变量或引用为函数添加新的参数，还能够自动更新引用此函数的其它位置的默认参数。

Introduce Factory



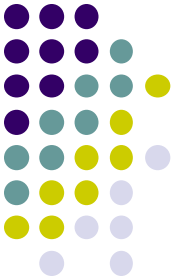
- “Introduce Factory”重构能够为类创建工厂方法。

Convert Local Variable to Field

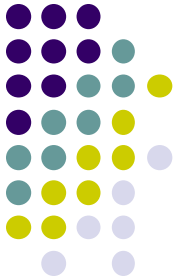


- Convert Local Variable to Field重构能够把局部的变量转换成类中的全局变量。

Encapsulate Field



- Encapsulate Field重构能够包装属性的可访问性，以及生成访问的方法。



结束语

Eclipse 提供的工具使重构变得简单，熟悉这些工具将有助于提高效率。敏捷开发方法采用迭代方式增加程序特性，因此需要依赖于重构技术来改变和扩展程序的设计。Eclipse 的重构工具可以在进行一般的代码修改时提供节约时间的方法，因此，花一些时间来熟悉这些工具，以便于在出现可以利用它们的情况时进行重构是值得的。

谢谢!