|  |  |  |
| --- | --- | --- |
| **Composing Methods** | | |
| **Refactoring Name** | **Detectable** | **Why?** |
| Extract Method |  |  |
| Inline Method |  |  |
| Inline Temp |  |  |
| Replace Temp with query |  |  |
| Introduce Explaining variable |  |  |
| Split temporary variable |  |  |
| Remove assignments to parameters |  |  |
| Replace Method with method object |  |  |
| Substitue algorithm |  |  |

|  |  |  |
| --- | --- | --- |
| **Moving Features between objects** | | |
| **Refactoring Name** | **Detectable** | **Why?** |
| Move method |  |  |
| Move fiels |  |  |
| Extract class |  |  |
| Inline class |  |  |
| Hide delegate |  |  |
| Remove middle man |  |  |
| Introduce foreigns method |  |  |
| Introduce local extension |  |  |

|  |  |  |
| --- | --- | --- |
| **Organizing data** | | |
| **Refactoring Name** | **Detectable** | **Why?** |
| Self encapsulate field |  |  |
| Replace data value with object |  |  |
| Change value to reference |  |  |
| Change reference to value |  |  |
| Replace array with object |  |  |
| Duplicate observed data |  |  |
| Change unidirectional association to bidirectional |  |  |
| Change bidirectional association to unidirectional |  |  |
| Replace magic number with symbolic constant |  |  |
| Encapsulate field |  |  |
| Encapsulate collection |  |  |
| Replace record with data class |  |  |
| Replace type code with class |  |  |
| Replace type code with subclass |  |  |
| Replace type code with state/strategy |  |  |
| Replace subclass with fields |  |  |

|  |  |  |
| --- | --- | --- |
| **Simplifying conditional expressions** | | |
| **Refactoring Name** | **Detectable** | **Why?** |
| Decompose conditional |  |  |
| Consolidate conditional expression |  |  |
| Consolidate duplicate conditional fragments |  |  |
| Remove control flag |  |  |
| Replace nested conditional with guard clauses |  |  |
| Replace conditional with polymorphism |  |  |
| Introduce null object |  |  |
| Introduce assertion |  |  |

|  |  |  |
| --- | --- | --- |
| **Making method calls simpler** | | |
| **Refactoring Name** | **Detectable** | **Why?** |
| Rename method |  |  |
| Add parameters |  |  |
| Remove parameter |  |  |
| Separate query from modifier |  |  |
| Parameterize method |  |  |
| Replace parameter with explicit methods |  |  |
| Preserve whole object |  |  |
| Replace parameter with method |  |  |
| Introduce parameter object |  |  |
| Remove setting method |  |  |
| Hide method |  |  |
| Replace constructor with factory method |  |  |
| Encapsulate downcast |  |  |
| Replace error code with exception |  |  |
| Replace exception with test |  |  |

|  |  |  |
| --- | --- | --- |
| **Dealing with generalization** | | |
| **Refactoring Name** | **Detectable** | **Why?** |
| Pull up method |  |  |
| Pull up constructor body |  |  |
| Push down method |  |  |
| Push down field |  |  |
| Extract subclass |  |  |
| Extract superclass |  |  |
| Extract superclass |  |  |
| Extract interface |  |  |
| Collapse hierarchy |  |  |
| Form template method |  |  |
| Replace inheritance with delegation |  |  |
| Replace delegation with inheritance |  |  |
| **Big refactorings** | | |
| **Refactoring Name** | **Detectable** | **Why?** |
| Tease apart inheritance |  |  |
| Convert procedural design to objects |  |  |
| Separate domain from presentation |  |  |
| Extract hierarchy |  |  |