Refactoring

Code refactoring

Fowler says that refactoring is the

"*... process of changing a software system in such a way that it does not alter the external behavior of the code yet improves its internal structure.*"

Just cleaning up code.

Why Refactor?

- To improve the quality of the codebase
- Makes software easier to understand
- This in turn helps in finding bugs
- .. and in turn allows you to program faster in the end.

Code refactoring

- Contrary to idealized development strategy:
  - analysis and design
  - code
  - test
- At first, code is pretty good but as requirements change or new features are added, the code structure tends to atrophy. Refactoring is the process of fixing a bad or chaotic design.
- Amounts to moving methods around, creating new methods, adding or deleting classes, ...

tool: RefactorIT

Rename
Renames a method, field, type, package or prefix. Updates all references.

Move Class
Moves a class or interface into another package.

Encapsulate Field
Replaces direct field usage with corresponding accessor methods.

Extract Method
Analyzes the selected piece of code and extracts it into a separate method.

tool: RefactorIT

Extract Super-class/Interface

Extracts selected methods and fields into new superclass or interface.

Minimize Access Rights

Determines the minimal access modifiers for class fields and methods. Automatically changes selected modifiers.

Create Constructor

Creates a simple constructor on group of field declarations that initializes these fields.