Table of Contents

Refactoring

Code refactoring is the process of restructuring existing computer code—changing the factoring—without changing its external behavior.

*Snippet from Wikipedia: **Code refactoring***

> In computer programming and software design, **code refactoring** is the process of restructuring existing computer code—changing the *factoring*—without changing its external behavior. Refactoring is intended to improve the design, structure, and/or implementation of the software (its *non-functional* attributes), while preserving its functionality. Potential advantages of refactoring may include improved code readability and reduced complexity; these can improve the source code's maintainability and create a simpler, cleaner, or more expressive internal architecture or object model to improve extensibility. Another potential goal for refactoring is improved performance; software engineers face an ongoing challenge to write programs that perform faster or use less memory.
>
> Typically, refactoring applies a series of standardized basic *micro-refactorings*, each of which is (usually) a tiny change in a computer program's source code that either preserves the behavior of the software, or at least does not modify its conformance to functional requirements. Many development environments provide automated support for performing the mechanical aspects of these basic refactorings. If done well, code refactoring may help software developers discover and fix hidden or dormant bugs or vulnerabilities in the system by simplifying the underlying logic and eliminating unnecessary levels of complexity. If done poorly, it may fail the requirement that external functionality not be changed, and may thus introduce new bugs.
>
> > By continuously improving the design of code, we make it easier and easier to work with. This is in sharp contrast to what typically happens: little refactoring and a great deal of attention paid to expediently add new features. If you get into the hygienic habit of refactoring continuously, you'll find that it is easier to extend and maintain code.

Creative Commons Attribution-Share Alike 4.0

Related:

- Technical Debt

External links:

- https://refactoring.com/

kb, programming

From:
https://almbok.com/ - **ALMBoK.com**

Permanent link:
https://almbok.com/kb/refactoring

Last update: **2022/08/10 06:33**