

Practical Refactoring II: Patterns and Practices

خلاصه :

مارتین فولر در مقدمه‌ی کتاب (Refactoring: Improving the Design of Existing Code (2nd Edition)) بازنویسی (Refactoring) را این طور تشریح می‌کند:

"بازنویسی (Refactoring) تکنیک کنترل شده‌ای برای بهبود طراحی کدهای موجود است. ماهیت آن اعمال یکسری تغییر شکل‌های کوچک در کد با حفظ رفتار قبل است، که هر یک از آنها «به‌حدی کوچک هستند که ارزش انجام دادن داشته باشند». بنابر این تجمیع تاثیر هر یک از این تغییر شکل‌ها کاملاً مهم و قابل توجه است. با انجام این کار در قدم‌های کوچک، ریسک ایجاد خطاها را کاهش می‌دهید، از هم فروپاشی سیستم در هنگامی که بازسازی انجام می‌شود، اجتناب می‌کنید و فرصت می‌یابید تا به تدریج بازسازی سیستم را در یک بازه‌ی زمانی طولانی‌تر انجام دهید."

این دوره به‌گونه‌ای طراحی شده است که شرکت‌کنندگان در آن به ازای هر تکنیک بازسازی، مشکل را شناسایی می‌کنند، راه حل آن را می‌فهند، چرایی آن را درک می‌کنند، با مزیت(های) آن آشنا می‌شوند، موانع انجام آن را در می‌یابند، زمان انجام آن را فرا می‌گیرند، چگونگی انجام آن در کد را یاد می‌گیرند و سایر تکنیک‌های مشابه با آن را می‌بینند.

همچنین شرکت‌کنندگان در آن به ازای بوهای کد (Code Smells) با علامت‌ها و نشانه‌های آن آشنا می‌شوند، دلایل ایجاد آن را می‌فهمند، تکنیک‌های بازنویسی آن را فرا می‌گیرند و بازدهی و کارایی کد را بعد از بازنویسی می‌بینند.

مدت دوره: ۲۴ ساعت

دوره در قالب جلسه‌های ۴ ساعته برگزار خواهد گردید. مدت کل دوره ۲۴ ساعت یا ۶ جلسه است.

پیش نیاز:

شرکت‌کنندگان این دوره باید با یکی از زبانهای برنامه نویسی #C، JAVA یا PHP آشنایی داشته باشند.

مخاطب:

- برنامه‌نویسان
- دانش‌جویان و دانش‌آموختگان رشته‌های نرم‌افزار

اهداف دوره:

مهمترین اهداف این دوره عبارتند از:

- آشنایی با انواع بدهی فنی
- آشنایی با انواع بوهای بد در کد
- آشنایی با انواع تکنیک‌های بازسازی کد
- توانایی تشخیص کد تمیز از کد کثیف
- توانایی انتخاب زمان مناسب برای بازسازی کد
- توانایی انجام بازسازی کد به صورت عملی

این دوره شامل:

- ۳۳۷ صفحه جزوه به زبان JAVA، C# و PHP
- ۶۶ تکنیک بازسازی کد
- ۲۱ دسته بو در کد بد
- آموزش از طریق ارائه‌ی مثال قبل از بازسازی و بعد از بازسازی کد در JAVA، C# و یا PHP

سرفصل دوره:

Introduction:

- What is refactoring?
- How to write clean Code?
- How to identify technical debt?
- When to refactor?
- How to refactor?

Code Smells:

- Bloaters
- Object-Orientation Abusers
- Change Preventers
- Dispensables
- Couplers
- Other Smells

Code Refactoring Techniques:

- Composing Methods
- Moving Features between Objects
- Organizing Data
- Simplifying Conditional Expressions
- Simplifying Method Calls
- Dealing with Generalization

منابع دوره:

- 1) Refactoring: Improving the Design of Existing Code (2nd Edition), Martin Fowler
- 2) Clean Code: A Handbook of Agile Software Craftsmanship, Robert C. Martin
- 3) Refactoring to Patterns, Joshua Kerievsky



