

Refactoring

auch große Umbauten meistern

- Weniger ist mehr
- Bauplan
- Die Öffentlichkeit
- Keine Schulden
- Fragen

Weniger ist mehr

Refactoring mit API-Änderungen neigt dazu, immer größer zu werden. Deshalb ist beim Refactoring Disziplin gefragt. Hier ein paar grundlegende Tips:

- Ein Refactoring nach dem anderen, nicht mehrere auf einmal
- Zerlege ein großes Refactoring in kleine Schritte
- Lieber unbenutzter neuer Code als kaputter alter Code



Bauplan



- Mach Dir vorher einen Plan
- Definiere einzelne (eincheckbare) Schritte
- Checke oft ein, Subversion ist Dein Freund
- Breche alte APIs möglichst spät
- Spreche Dich ggf. mit anderen ab, um Konflikte zu vermeiden
- Trau Dich, jeder darf es

Die Öffentlichkeit

- Vorsicht bei öffentlichen APIs!
- Public ist öffentlich, protected aber auch
- Nicht alles sollte public sein
- Zunächst deprecated markieren statt löschen
- Versionierung an API-Änderungen anpassen
- Im Notfall lieber eine Exception werfen, aber kompi



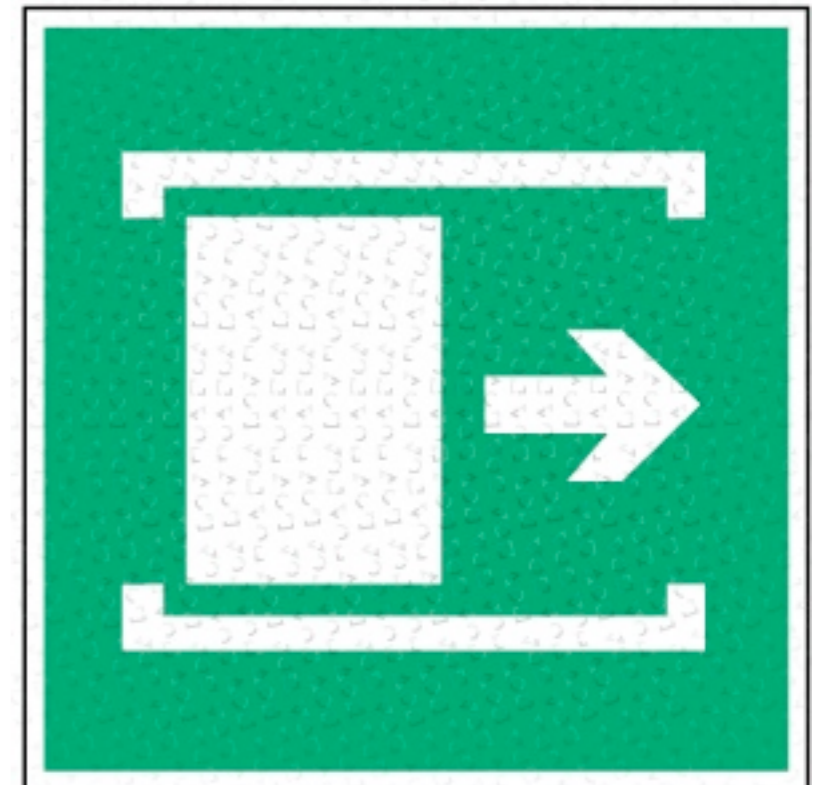
Implementierung austauschen

- Neue Implementierung zunächst zusätzlich zur alten aber unbenutzt
- Spring-Konfiguration fertigstellen
- Austauschen der Implementierung durch Umbenennung des Beans
- Alte Implementierung löschen



Feld verschieben

- Neues Feld anlegen
- Neues Feld befüllen, altes Feld als deprecated markieren
- Verwendungen des alten durch das des neuen Feldes ersetzen
- Altes Feld nicht mehr setzen
- Altes Feld entfernen



Schnittstellen erweitern

- Gilt für Interfaces und abstrakte Klassen
- Existierender Code geht potenziell kaputt!
- Neue Methode: besser neues abgeleitetes Interface mit neue Methode, an benötigten Stellen neues Interface verwenden und implementieren
- Methode löschen: zuerst neues (Super)Interface, dann Verwendungen und Implementierungen des alten entfernen
- Methode umbenennen: wie neue und dann alte löschen
- Signatur ändern: neue Methode, alte als deprecated markieren und bei Implementierung wenn möglich auf neue Signatur verweisen

keine Schulden machen

- Refactoring bis zum Ende durchführen
- Alte, unbenutzte Implementierungen, Methoden und Felder am Ende löschen
- Deprecated nur bei öffentlichem API stehen lassen, sonst Benutzung und Klasse/Methode entfernen



Fragen

