

Restructuring Code: From “Push” To “Pull”

Proposal for a Practitioner Workshop

Nicole Rauch

Andreas Leidig

Summary

This workshop is intended to introduce the participants to a systematic and structured way of performing large restructurings in object-oriented code. Refactorings in general are nowadays an integral part of a software developer’s toolbox. Nonetheless, it is still rather difficult to accomplish large restructurings by applying these refactorings because each of them performs only small changes.

This workshop provides a guideline that supports the developer in conducting such restructurings in a directed way by applying a series of small incremental refactorings. We will present a way of reshaping existing code. After a short introduction, the participants will work in pairs on an example Java codebase where they will perform the restructuring steps themselves.

Session Contents

Developers often experience code that is both hard to read and to track. At the same time, they strive to write “clean code” that follows certain principles like single responsibility, encapsulation and coherence. While most developers know these principles by name, they often find it hard to establish them, especially in legacy code. They feel the need to restructure parts of the application, but they have no idea how they could perform such a large-scale restructuring, and sometimes they might not even have an idea of what the resulting code should look like. One can find lots of information on how to perform small refactorings in legacy code, but there is almost no guidance for performing large restructurings and redesigns.

This workshop intends to give a concrete example of how a large restructuring can be performed in a systematic, structured and reproducible way, taking small and safe steps. If this strategy is carried out consequently, the resulting code follows the abovementioned principles and becomes object-oriented, with small methods in objects communicating with each other, in one word: the code has become “clean code”.

Our strategy addresses a specific coding pattern which we have found in many applications: Fields in objects are written from multiple locations in the code. This has various effects: The value of a field at a given time is not clear, and tracking back all the methods that write the field is difficult. Also, it is hard to tell how and when the results have been determined because there is no unique location in the code where the transformation from the input values to the individual output values takes place. We call this way of coding “Push” because the code actively pushes its results into some object’s fields. These results are determined in a procedural and sequential way. The resulting code is rather fragile

and tends to be buggy. Changing and extending the code is difficult and often turns into a “Shotgun Surgery”.

Instead of using the “Push” style, the output values can be calculated on-demand when retrieving them. This way, the algorithms for each value are clearly visible and very descriptive. We call this style “Pull” because the values are pulled by other parts of the code. The resulting code has proven to be readable, understandable, testable, changeable and extensible. The participants can directly experience these advantages by implementing “Pull” style code themselves.

The session will start with a short presentation (15-30 mins) of the goal and the background of the restructuring. After that, we will introduce our example codebase which uses the “Push” style to implement the following requirement: From a sequence of bank account transactions, determine the balance of the account at the end of each month as well as its monthly average. We will conduct a hands-on session where the participants will use this codebase to perform the restructuring themselves in pairs, guided by a sequence of small refactoring and restructuring steps (some automatable, some not). Each of these steps will be introduced by a short presentation (2-5 mins), and we will support the participants in conducting them. A short debriefing will conclude the workshop.

Organisational Information

Proposed Length

We propose this workshop to take half a day.

Intended Audience and Expected Benefits

This workshop is intended for software developers who know Java and who are interested in expanding their knowledge with regard to object-oriented software development.

We expect the following benefits for the participants: The participants will learn how to perform a large restructuring that leads to “clean” object-oriented code. They will get to know the characteristics of the “Push” style, and they will learn to recognize code that follows it. They will also get to know the “Pull” style and how it helps them to improve their code according to the widely accepted clean code principles.

The most important takeaway for the participants will be the sequence of concrete refactoring steps that allows them to perform large restructurings by breaking them into small systematic transformations.

Benefits for the workshop organizers are that they will learn which steps of the restructuring are easy to understand and follow and which are unintuitive for the participants. This way, the organizers can improve the restructuring process as well as their workshop.

Expected Outcomes

All shown and used material and especially the initial example code as well as the code after each transformation step will be available from Github.

Number of Participants

We can conduct this workshop with up to 24 people.

Room Setup

We need tables and chairs for the participants. The participants should be seated in pairs. There should be one computer per pair, containing a Java IDE. There should additionally be one keyboard and mouse per pair. A beamer is required as well. A whiteboard and/or a flipchart are welcomed.

Organizer Information

Nicole Rauch (Primary Organizer)

Nicole Rauch is a software developer with a solid background in compiler construction and formal methods. Over the last two years, she worked on the restructuring of large legacy code applications with different technologies. Alongside her occupation as developer, she took part in organizing some self-organized conferences related to agile coaching.

Andreas Leidig (Additional Organizer)

Andreas Leidig has broad experience in object oriented software development and agile practices. He started programming years ago with Smalltalk and worked as an agile coach for a few years. Nowadays he is developing enterprise software products.

Experience and Background

Nicole and Andreas co-operate at work and in user group activities. Their passion for producing great software led them to organize a conference on Software Craftsmanship, known as SoCraTes. Subsequently they initiated Softwerkskammer, a Germany-wide user community for software craftsmanship, and Nicole founded the regional Softwerkskammer branch in Karlsruhe.

They are both very active members of this community and perform regular activities. They speak at conferences several times a year, often together.

Both are experienced workshop facilitators. They both ran (often together) TDD workshops as well as (Legacy) Code Retreats. They also facilitated numerous coding dojos focusing on subjects such as TDD, refactoring or legacy code.

Prior Presentations of this Workshop

The idea for this workshop arose from a large restructuring which we performed at work with a huge Java Enterprise application. We transformed about 1500 LOC which led to twelve times as many classes and four times as many methods while reducing the total number of statements to about 60 % as well as lowering the class complexity to 5 % and the method complexity to 50 % of their original values. During the last few months this code had to be changed and extended many times. It has shown to be very robust and flexible. The runtime performance even improved and further optimizations could be applied in isolation. It was also easy to increase the unit test coverage.

So far, we gave one talk at Entwicklertag Karlsruhe 2012 where we presented a first sketch of the steps for the restructuring from “Push” to “Pull”. This is our first version of a workshop in which the participants themselves perform the steps; we have not run it in public before.