



Die Regel für kurzen Code

Scrollen im Quelltext ist Mist. Besonders, wenn man an zwei weit entfernten Stellen quasi gleichzeitig etwas ändern muss.

Klar kann man das Fenster teilen und damit die beiden Stellen immer im Auge behalten. Aber diese Notwendigkeit zeigt ein wichtiges Feature des Quelltextes: Er ist zu lang. Feature ist hier ein Synonym für Bug, denn langer Quelltext ist weder Clean Code noch gut. Er ist schlecht, weil unübersichtlich. Die alles entscheidende Frage ist nur: Warum schreibe ich immer solchen Code?

Damit das künftig nicht mehr passiert, habe ich jetzt eine Regel entwickelt. Ja, ich will auch mal was Wichtiges zur Softwareentwicklung beitragen. Die Regel heißt: Rule of [Strg]+[Tab], gesprochen: Rule of Steuerung plus Tab.

Um von einer Stelle des Codes schnell zu einer anderen Stelle in einer anderen Datei wechseln zu können, ist die Tastenkombination [Strg]+[Tab] ideal. Das geht über die Tastatur schnell von der Hand und Maus und Scrollbar bleiben unberührt. Deshalb ist diese Variante des Kontextwechsels so schnell.

Scrollen hingegen bedarf der Maus und ist extrem ungenau. Das Finden einer Stelle dauert lang. Okay, na gut, man könnte noch mit [Strg]+[G] und Eingabe der Zeilennummer schnell hinspringen. Aber das ist viel aufwendiger als [Strg]+[Tab].

Deshalb habe ich jetzt in schweißtreibender Kleinarbeit die Rule of [Strg]+[Tab] entwickelt.

Und das lautet so: Teile deinen Code in kleine Pakete auf, damit du mit [Strg]+[Tab] schnell zwischen mehreren Stellen hin- und herspringen kannst. Pakete ist hier synonym für Dateien zu verstehen, und da idealerweise in jeder Datei eine Klasse, Komponente et cetera enthalten ist, bewirkt die Regel eine Verkürzung langer Quelltextorgien.

Erste Einsätze brachten gute Ergebnisse: Die Dateien werden tatsächlich kürzer. Ein Hin- und Herscrollen ist wesentlich seltener nötig.

Ein Problem ist allerdings noch nicht gelöst: Die Anzahl an Dateien in einem Projekt wächst dadurch so stark, dass man ohne Scrollen in der Projektmappe die gewünschte Datei nicht mehr findet. Mal sehen, welche Regel mir hierfür einfällt.

Viel Spaß mit der dotnetpro

Tilman Börner
Chefredakteur dotnetpro



Fabian Zankl

packt mithilfe von CoreRT eine Anwendung in eine einzige Datei, die nativ läuft. S. 58



Martin Gossen

zeigt, wie Sie GUI-Tests mit dem Windows Application Driver aufsetzen. S. 70



Christian Kiefer

sorgt mit dem Library Manager für Ordnung im Chaos der ASP.NET-Core-Projekte. S. 110