Unit Testing in Java

Johannes Link

With Contributions by Peter Fröhlich

Unit Testing in Java

How Tests Drive the Code

Johannes Link

With Contributions by Peter Fröhlich

MORGAN KAUFMANN PUBLISHERS

AN IMPRINT OF ELSEVIER SCIENCE

AMSTERDAM BOSTON LONDON NEW YORK
OXFORD PARIS SAN DIEGO SAN FRANCISCO
SINGAPORE SYDNEY TOKYO

Senior Editor	Tim Cox
Publishing Services Manager	Edward Wade
Editorial Coordinator	Stacie Pierce, Richard Camp
English translation	Angelika Shafir
Project Management	Matrix Productions, Inc.
Cover Design	Frances Baca
Cover Image	Photodisc Collection/Getty Images
Text Design	Rebecca Evans
Composition	Nancy Logan
Illustration	Dartmouth Publishing, Inc.
Copy Editor	Yoni Overton
Proofreader	Dan Young
Indexer	Edwin Durbin
Interior printer	The Maple-Vail Book Manufacturing Group
Cover printer	Phoenix Color Corporation

Morgan Kaufmann Publishers
An Imprint of Elsevier Science
340 Pine Street, Sixth Floor
San Francisco, CA 94104-3205
www.mkp.com

Title of German original: Unit Tests Mit Java

Printed in the United States of America

07 06 05 04 03 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data
Link, Johannes.
Unit testing in Java : how tests drive the code / by Johannes Link ; with contributions by Peter Fröhlich.
p. cm.
Includes bibliographical references and index.
ISBN 1-55860-868-0
1. Computer software—Testing. 2. Java (Computer program language)
I. Fröhlich, Peter. II. Title.
QA76.76.T48L55 2003
005.1'4—dc21
Morgan Kaufmann ISBN: 1-55860-868-0
dPunkt ISBN: 3-89864-150-3

This book is printed on acid-free paper.

Foreword

Erich Gamma

As a professional software developer, I want to develop software as fast as possible, as well as possible, and as stress-free as possible. Automated unit tests help to bring me closer to this goal. They are a small investment, which help me get confidence in the code I produce and maintain later on. When I don't have automated tests, I have to fall back on manual testing. However, manual tests cannot be automatically repeated. Consequently, the stress increases, particularly when they have to be done under time pressure, which isn't, of course, the exception. At the push of a button you can determine at any given time whether or not the last change impacts the fitness of your software. You can do this today, tomorrow, or any time in the future, regardless of whether or not a deadline is knocking at your door.

This book by Johannes Link and contributor Peter Fröhlich is a practical introduction to using automated unit tests and the test-first approach in your day-to-day software development. The automation framework used in the book is JUnit. It is a small and simple framework for creating and managing tests. However, more is needed for successful development with unit tests. In fact, a developer has to be familiar with many different testing techniques, in particular when unit tests have to be created in the context of databases or distributed applications based on application servers. This book sheds light on these problems and is a highly welcome contribution to the field of automated unit testing.

JUnit itself was also developed with automated unit tests and the test-first approach, as explained in this book. In fact, the techniques were even used under challenging conditions, such as while fighting jet lag or during

electric power failures in alpine cabins. Still, the techniques have proven themselves every time. I hope that, thanks to this book, you will become test infected and that in the future you will always be able to give a positive answer to the classical unit test control question, "Where are the unit tests?"

Erich Gamma
Co-author of JUnit
Technical Director, Object Technology International

Foreword

Frank Westphal

Do you remember your first programming experience? I don't mean the details of the computer or language used; I mean how did you feel?

I remember typing in a few statements from the programming handbook, eager to see the program run. It was amazing to watch how the code sprang to life. Within a few hours I had grown the book example into what seemed an impressive program. I had made additions here and there and after every change I would rerun the program to see how it was doing. In the evening I showed it to my parents. They could tell by the look in my eyes how proud I was.

How things have changed. I enjoy programming more than ever, but now and then I realize that some of the original fun is gone. It's in these moments that I reflect back on my first experience. Why can't programming always be like that?

Actually, it was in one of those moments that I came across the techniques described in this book. Automated tests and continuous refactoring applied in tandem brought me a bit closer to the beginning of my programming career. More often than not since then, I have been able to act like the only thing I have to do is write a few lines of code. But don't be fooled; these techniques are not only applicable to small programs. The larger the scale, the more valuable these techniques became to me.

I was glad when Johannes asked me to write a second foreword for two reasons. First, this book brings together the knowledge that a number of pioneering extreme programmers wish they had when they started applying test-first programming five years ago. If you follow down that route, you will invariably run into testing problems. Even though you are writing

your tests first, you will come to halt because you won't see how to test your code. That's natural. Actually, that's the perfect time to reflect. Or to pick up this book and read what Johannes tells us.

Second, I was going to write a book just like this one. However, when Johannes shared with me the first few chapters for review, I could see that it was well written and even covering a large suite of tools to support testing code that is usually hard, if not impossible, to test. I wish I had written this book. Therefore, writing an accompanying foreword is a great pleasure.

There is but one danger in reading this book. You might come away with the impression that it's all about techniques and tools. When in fact, it's all about you.

Test-infected programmers will tell you how the tests changed their relationship to the code. There is a certain fascination in seeing a few hundred tests passing and checking all the innards of your software. Indeed, sometimes you will find yourself pressing the run button a few more times, just for the extra kick that everything's working fine.

Frank Westphal
Independent trainer and consultant

Contents

Preface

When my German publisher told me that my book was going to be translated into English I was delighted: no work for lots of fame. This view was more than naive. Despite the translator's excellent job, the number of errors and omissions uncovered by the reviewers was quite a revelation.

So I had to take the challenge and rewrite parts of a book that I had already deemed finished and of high quality. What you hold in your hands now is maybe 70% translation and 30% new, updated, and improved material. And still I feel guilty for not having been able to seize all of the reviewers' suggestions.

As the book changed shape and went through heavy restructurings, my life did as well. Thanks to all who supported me in one process or another. These include the reviewers of both the German and the English versions: Frank Adler, Achim Bangert, Markus Barchfeld, Ekard Burger, Frank Cohen, Herbert Ehrlich, Eitan Farchi, Tammo Freese, Dierk König, Andreas Leidig, Erik Meade, Steve Metsker, Rainer Neumann, Christian Popp, Ilja Preuß, Stefan Roock, Michael Ruppert, Roland Sand, Martin Schneider, Thomas Singer, Andreas Schoolmann, Robert Wenner, Timothy Wall, and Frank Westphal; Angelika Shafir, who succeeded in translating not only the facts but also the spirit of the original text; Tim Cox and Stacie Pierce of Morgan Kaufmann Publishers, who value quality much higher than publication speed; Peter Fröhlich, co-author of the German version, who persevered through our discussions about language and style; all the people forming andrena objects, a more than suitable place to develop new ideas and to confront these ideas with reality; and Bettina and Jannek, who will hopefully help me fill many pages of our shared personal "book" with happiness and sadness.