

# Preface

The challenges in today's software development are diverse and go far beyond implementation tasks. They range from requirement specification over system design and implementation to maintenance and further adaptation of the software—to name just a few phases in the software life cycle. In all of these phases of the software development process, many people with different backgrounds and experiences are usually involved. These people need a common language for efficient communication. Obviously, such a language should be as precise as possible without the ambiguities of a natural language. For this purpose, modeling languages have emerged. They are used to create sketches and blueprints for software systems, which in turn serve as a basis for the implementation or even automatic generation of executable code. In the area of object-oriented software development, the Unified Modeling Language (UML) was able to prevail. Of course, to use the language correctly and efficiently, it is necessary to understand the concepts offered by UML. Since 2006, we have offered the course “Object-Oriented Modeling” at the Vienna University of Technology. This course is mandatory for computer science and business informatics students in their first year. Overall, we have up to 1,000 students per year who attend our course. To deal with such a huge number of students while keeping high quality standards, much effort has been spent on the preparation of such a course. This includes the overall organization, course material, and e-learning support. Parts of the course design have been presented at the Educators' Symposium of the MODELS conference [8, 9, 10, 11, 7, 46]. We teach the basics of object-oriented modeling by means of UML.

In particular, we teach

- class and object diagrams,
- sequence diagrams,
- state machine diagrams,
- activity diagrams, and
- use case diagrams

as well as their interrelations. For this purpose, we introduce the syntax (the notation of the language elements), the semantics (the meaning of the language elements), and the pragmatics (how to use the language elements) of these UML diagrams. They cover the most essential concepts of object-oriented modeling and are used in many different stages of the software development process. The course is designed for students who already know the basic concepts of object-oriented programming languages such as Java or C#, but still have no practical experience in software engineering. Based on our comprehensive experience in teaching UML, we wrote the book *UML@Classroom*. In this book, we address both readers who wish to learn UML in a compact but nevertheless precise manner and teachers whom we want to provide with inspiration for their own course exercises with our extensive example repertoire. We teach UML as close to the standard as possible and illustrate all concepts using intuitive examples. The book is complemented by a website, which contains a complete set of slides to teach the contents of the book as well as teaching videos and e-learning material (<http://www.uml.ac.at/>).

Software modeling is a very young field of computer science. It experienced an incredible growth within the last two decades. Today, the usage of models goes far beyond pure documentation. Techniques from the area of modeling continually replace conventional programming. Models are far more than just pictures, and modeling is far more than just drawing. With our book *UML@Classroom*, we want to provide a solid foundation and deep understanding of the most important object-oriented modeling concepts. We aim for rising interest and enthusiasm for this exciting and extremely important field of computer science. *UML@Classroom* is a textbook, which explicitly addresses beginners and people with little or no modeling experience. It introduces basic concepts in a very precise manner, while abstaining from the interpretation of rare special cases. *UML@Classroom* is kept very compact in order to allow the reader to focus on the most commonly used concepts of UML. Much emphasis is spent on illustrative examples breathing life into the theory we present.



<http://www.springer.com/978-3-319-12741-5>

UML @ Classroom

An Introduction to Object-Oriented Modeling

Seidl, M.; Scholz, M.; Huemer, C.; Kappel, G.

2015, XII, 206 p., Softcover

ISBN: 978-3-319-12741-5