

Vorwort

Zugegeben, es hat eine Weile gedauert, bis ich die Möglichkeiten der Objektorientierung in MATLAB® schätzen lernte. Im Jahr 2007 schrieb ich ein Lehrbuch, das sich mit dem „Programmieren mit MATLAB“ beschäftigt. Damals standen wir am Fachbereich Maschinenbau und Produktion der HAW Hamburg vor einem Problem: Unsere Studenten waren nicht recht zu motivieren, eine Programmiersprache zu erlernen. Wir „zwangen“ sie, sich längere Zeit mit Compiler, Linker und unverständlichen Fehlermeldungen herumzuquälen, um am Schluss als Ergebnis die Zahl 42 auf dem Bildschirm zu sehen. Ingenieure kommen inzwischen immer weniger mit Programmiersprachen wie C++ in Kontakt, ob in der Industrie oder im weiteren Verlauf des Studiums. Anders liegt die Situation beim Programm MATLAB. Dessen Funktionalität wird bei uns intensiv genutzt, beispielsweise zum Lösen von Differentialgleichungen oder zum Ansteuern von Robotern. Und besonders die grafischen Möglichkeiten von MATLAB fördern das Verständnis für viele technische Anwendungen. Deshalb habe ich damals, zusammen mit Kollegen, unser Bachelor-Modul „Angewandte Informatik“ auf die in MATLAB integrierte Programmiersprache umgestellt. Aus dem Skript zu dieser Vorlesung entstand das oben erwähnte Lehrbuch.

An die objektorientierte Programmierung (OOP) dachte ich dabei nur am Rande. Zwar hatte MATLAB auch damals bereits eingeschränkte OOP-Funktionalität, die ich in meinem Buch beschrieb. Die Vorgehensweise war jedoch noch sehr umständlich. Der große Schritt kam im Jahr 2008 mit der Einführung der *classdef*-Datei, die ähnlich wie in anderen OOP-Sprachen die Struktur einer Klasse festlegt. Durch die Definition von Handle-Klassen erhielt MATLAB außerdem eine Art von Referenzen.

Trotzdem, eine Zeit lang blieb ich noch skeptisch, ob man mit MATLAB wirklich vernünftig objektorientiert programmieren kann. Im Vergleich zu anderen OOP-Sprachen fehlten doch ein paar Dinge, beispielsweise die Möglichkeit, mehrere Methoden mit demselben Namen, aber unterschiedlicher Signatur zu deklarieren. Dies ist in MATLAB nicht vorgesehen. Aber hierfür – und auch für andere fehlende Bereiche – kann man sich recht einfach einen vernünftigen Workaround bauen.

Je länger ich mich mit der OOP-Funktionalität in MATLAB beschäftigte, desto mehr hat sie mich überzeugt. Und ich hoffe, dass ich auch Sie in diesem Buch dafür begeistern kann.

Nun zu Ihnen: Was erwarte ich von meinem Leser?

Idealerweise sollten Sie bereits ein wenig programmiert haben, nicht notwendigerweise in MATLAB. Zwar stelle ich Ihnen in diesem Buch, in den Kapiteln 1 und 2, die Oberfläche und die wichtigsten Sprachelemente von MATLAB vor – soweit es zum Verständnis der folgenden Abschnitte notwendig ist. Das ist jedoch nur eine knappe Zusammenfassung, eher gedacht für Umsteiger von anderen Programmiersprachen wie C++. Einem absoluten

Anfänger würde ich als Einstieg eher mein Buch „Programmieren mit MATLAB“ empfehlen – ein Lehrbuch, in dem Sie auch viele Beispiele und Übungen finden.

Doch zurück zum jetzigen Buch: Kapitel 3 beschreibt den objektorientierten Ansatz von MATLAB. Hier werden die zentralen Begriffe eingeführt, wie Klassen, Eigenschaften, Methoden, Datenkapselung, Vererbung, Polymorphie, Handle-Klassen etc.

In Kapitel 4 finden Sie längere Anwendungen aus der Physik und dem Maschinenbau. Dies soll den Umgang mit dem Erlernten vertiefen und weitere Tipps für ein strukturiertes Vorgehen geben.

Zum Abschluss liefert Kapitel 5 eine Befehlsreferenz und den Vergleich mit anderen OOP-Sprachen.

Dieses Buch ist aber kein Referenz-Handbuch für MATLAB. Die MATLAB-Funktionen werden oft nur so weit vorgestellt, wie es für die aktuelle Aufgabenstellung nötig ist. Für eine vollständige Definition der Funktionen sei auf die MATLAB-Hilfe verwiesen.

Die Idee, ein zweites Buch zu MATLAB zu schreiben, entstand während einer Unterredung mit Frau Franziska Jacob, M. A., meiner Ansprechpartnerin beim Fachbuchverlag Leipzig im Carl Hanser Verlag. Vielen Dank für ihr Engagement, mit dem sie das Projekt im Verlag durchsetzte. Dank auch an Frau Dipl.-Ing. (FH) Franziska Kaufmann, die mir beim Layout zur Seite stand. Dank an alle Kollegen, die mich zu diesem Projekt ermutigten und mir hilfreiche Tipps gaben, speziell Prof. Dr. rer. nat. Ivo Nowak, Prof. Dr. rer. nat. Thorsten Struckmann und Prof. Dr.-Ing. Jürgen Dankert. Und einen besonderen Dank an Elfriede Neubauer, die mir bei der stilistischen Überarbeitung eine große Hilfe war.

Die im Buch beschriebenen und abgebildeten Abläufe beziehen sich auf die Bedienoberfläche der Version MATLAB 2015a. Andere MATLAB-Versionen präsentieren sich dem Anwender zum Teil mit einer leicht abgewandelten Oberfläche. Lassen Sie sich deshalb nicht verwirren. Die vorgestellten Programme wurden mit verschiedenen Versionen getestet. Erweiterungen und die Lösungen der Aufgaben finden Sie auf meiner Homepage

www.Stein-Ulrich.de/Matlab/

Ich wünsche den Lesern, dass Ihnen das Programmieren auch Spaß macht und dass Ihnen möglichst viel vom hier präsentierten Stoff bei Problemlösungen nützt. Und nicht verdrängen oder vergessen: Informatik kann auch Schaden anrichten. Deshalb sollte jeder, der programmiert, sich überlegen, ob er sein Tun verantworten kann und will.

Hamburg, im August 2015

Ulrich Stein