

## **Prof. Dr.-Ing. Mathias Sporer**

erwarb nach einer Berufsausbildung zum Facharbeiter für Datenverarbeitung an der TU Dresden im Fernstudium den akademischen Grad des Diplom-Informatikers. Thema der Diplomarbeit war die ereignisorientierte diskrete Simulation technischer Prozesse. Diese Aktivitäten wurden an der TU Chemnitz fortgesetzt und führten zur Promotion auf dem Gebiet der Informatik mit dem Schwerpunkt der Datenbankunterstützung für den Entwurfsprozess eingebetteter Systeme. In der Industrie arbeitete Mathias Sporer auf dem Gebiet der System- und Anwendungsentwicklung für Mainframe- und Personal-Computer im Umfeld netzwerkorientierter und relationaler Datenbanksysteme. Nach Lehrtätigkeiten an der TU Chemnitz und der Staatlichen Studienakademie Glauchau erfolgte im Jahre 2010 die Berufung zum hauptamtlichen Dozenten für Technische Informatik und im Jahre 2017 zum Professor an der Berufsakademie Sachsen.

Die Forschungsinteressen liegen auf dem Gebiet der datenbankgestützten Modellierung und Simulation.

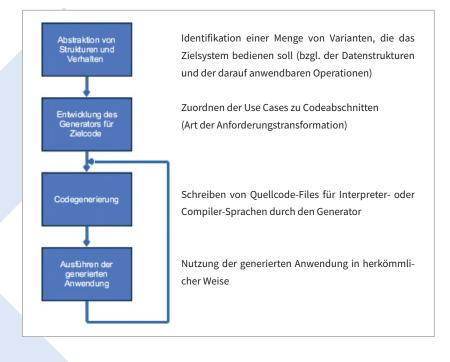
**KONTAKT:** Berufsakademie Sachsen | Staatliche Studienakademie Glauchau sporer@ba-glauchau.de

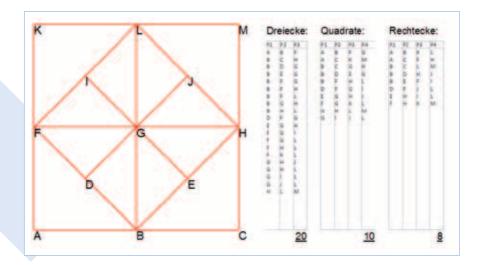
## Lehrmaterialien für Konzepte der generativen Programmierung

## Prof. Dr. Mathias Sporer

In einem von so starken Veränderungen geprägten Gebiet wie der Informatik erlangt die schnelle Übernahme wissenschaftlicher Erkenntnisse in die akademische Lehre immer größere Relevanz. Die Notwendigkeit, in kürzester Zeit komplexe Zusammenhänge vermitteln zu müssen, erfordert neue Formen der Lehre und insbesondere die Flexibilisierung von Unterrichtsmitteln. Dieses Ziel verfolgt das Projekt zur Entwicklung von Lehrmaterialien für die Informatik-Ausbildung.

Im Bereich der Software-Entwicklung standen Implementierungen zur generativen Programmierung im Mittelpunkt. Diese Technologie ermöglicht die effektive Entwicklung einer großen Anzahl strukturell ähnlicher Anwendungen durch Abstraktion der erwarteten Funktionalität. Die folgende Abbildung visualisiert die allgemeine Vorgehensweise sowie die Zielsetzungen der jeweiligen Abschnitte.





Am Beispiel von Datenbank-Anwendungen wurden diese Konzepte erprobt und ein Demonstrator entwickelt. Das graphentheoretische Beispiel 1 geht von einer gegebenen Punktmenge aus. Aus dieser sind ebene Figuren zu erzeugen, darzustellen und zu zählen.

Dazu erzeugt der Generator (in VBScript geschrieben) eine Microsoft Access-Datenbank mit Tabellen, Abfragen, Formularen, Berichten und Modulen.

Ausgangspunkt der weiteren Generierung ist die Tabelle der Punkte mit abstrakten Koordinaten. Diese bilden die Grundlage virtueller Tabellen, die – gesteuert von den Codeworten eines Binärcodes – jene Punkte identifizieren, die geometrische Objekte bilden können.

Selektionsbedingung ist z.B. bei Dreiecken die Dreiecksungleichung. Für die nicht in SQL verfügbaren Operationen erzeugt der Generator einen Modul mit VBA-Funktionen, die von der Datenbank-Engine aufgerufen werden. Ein Report-Objekt der Datenbank übernimmt die graphische Darstellung.

Ein weiterer Demonstrator generiert auf der Grundlage dieser Datenbank Zielcode für HTML (Aufbau der Web-Seite), JavaScript (Event-Handling) und SVG (Darstellung der Geometrie-Objekte). Somit kann der Nutzer die Figuren interaktiv über ihre Eckpunkte selektieren und die Graphik einfärben.

Die Prinzipien der generativen Programmierung wurden somit exemplarisch dargestellt und können von den Studierenden im Rahmen des eigenverantwortlichen Lernens bzw. in Praktika zum Software-Engineering auf andere Programmiersprachen übertragen werden. Anwendungsgebiete des generativen Ansatzes erschließen sich

- in der Web-Programmierung zur kundenspezifischen Oberflächengestaltung
- in der Low-Code-Entwicklung
- in der Cross-Compilierung für Spezialprozessoren
- in der Entwicklung von Transpilern

In allen Fällen ist besser wartbarer und in gewissen Grenzen formal prüfbarer Code das Ziel der Entwicklung. Den in der Anfangsphase höheren Aufwand für die Implementierung des Generators rechtfertigt die erhebliche Zeiteinsparung bei der späteren Nutzung desselben für eine größere Zahl ähnlicher Anwendungen. Der Ansatz ist sprachneutral und bereits mit Bordmitteln von Microsoft Office realisierbar.

## Literaturverzeichnis

Magazin "Kopf-fit", S & D Verlag, Geldern, Ausgabe 07/16, S. 42 Czarnecki, Krzysztof; Ulrich W. Eisenecker: Generative Programming: Methods, Tools, and Applications. Addison Wesley, 2000. Klar, Michael: Einfach generieren: Generative Programmierung verständlich und praxisnah. Hanser Fachbuchverlag, 2006.

