

# Programmierparadigmen

Programmiersprachen gehören zu den wichtigsten Werkzeugen der Computernutzung. Es wurden und werden viele Programmiersprachen entwickelt, die meist auf besondere Aspekte hin ausgerichtet und optimiert sind. Deshalb kann es oft von großem Vorteil sein, ein bestimmtes Problem mit einer dafür besonders geeigneten Programmiersprache zu lösen.

Darüber hinaus ist mit Programmiersprachen stets ein bestimmtes Denk- bzw. Problemlösungsparadigma verbunden. Darum ist die Kenntnis von Programmiersprachen im Hinblick auf die Verbesserung der Problemlösungskompetenz sehr nützlich.

Üblicherweise kennen Computerfachleute nur wenige Programmiersprachen und schöpfen daher das mögliche Lösungspotential nur in geringem Ausmaß aus.

Ziel dieses Seminars ist die Beschäftigung mit einer Programmiersprache (auch praktisch) und mit dem damit verbundenen Programmierparadigma, um die Möglichkeiten von weniger bekannten Ansätzen zu zeigen, und damit den Horizont zu erweitern.

Dazu sollen Sie sich beispielsweise mit folgenden Fragen auseinandersetzen:

- Wie sieht die Syntax dieser Sprache aus?
- Wie sieht eine typische Programmierumgebung für diese Sprache aus?
- Worin unterscheidet sich diese Sprache von verbreiteten wie C++, Java oder C#
- Für welches Paradigma ist diese Sprache am besten geeignet?
- Wodurch unterstützt diese Sprache gerade dieses Paradigma?
- Warum ist es in dieser Sprache nicht möglich, in einem anderen Paradigma (suchen Sie ein geeignetes) zu arbeiten?
- Wie sieht ein nicht triviales Codebeispiel aus, das die hinter der Sprache liegende Idee gut zum Ausdruck bringt (aber nicht länger als eine Folie ist)?
- Wie beurteilen Sie diese Sprache im Bezug auf ihren allgemeinen Nutzen (bzw. Mangel desselben)?

Folgende Programmiersprachen vertreten jeweils eines (oder mehrere) der größeren Paradigmen. Bearbeiten Sie die obigen Fragen in Gruppen für eine Sprache zu zweit oder zu dritt:

1. D Programming Language  
<http://www.digitalmars.com/d/2.0/index.html>
2. Erlang  
<http://www.erlang.org>
3. Smalltalk  
<http://www.smalltalk.org>, <http://www.squeak.org>
4. SELF  
<http://research.sun.com/self/>, <http://gliebe.de/self/index.html>
5. Scheme (und Common Lisp)  
<http://www.schemers.org>, <http://www.lisp.org/HyperSpec/FrontMatter/index.html>
6. Ruby  
<http://www.ruby-lang.org/en/>
7. Forth (und Factor)  
<http://forth.org>, <http://factorcode.org/>
8. Prolog  
<http://pauillac.inria.fr/~deransar/prolog/docs.html>
9. Haskell  
<http://www.haskell.org>
10. OCaml  
<http://caml.inria.fr/ocaml/index.en.html>