

# Kurs: Perl objektorientiert

© T. Birnthaler, OSTC GmbH ([www.ostc.de](http://www.ostc.de))

(Version 1.2 vom 19.6.2005)

**Dauer:** 2 Tage

**Motivation:** Perl ist eine portable und leistungsfähige Skript-Sprache, die weltweit von sehr vielen Entwicklern für allgemeine Programmentwicklung, für Web-Applikationen und in der Systemadministration eingesetzt wird. Man könnte Perl auch als "Linux/UNIX in a Box" bezeichnen, d.h. man erhält im Rahmen einer Programmiersprache die Vielzahl an Möglichkeiten und Hilfsmitteln des Linux/UNIX-Betriebssystems zur Verfügung gestellt. D.h. bei Linux/UNIX-Vorkenntnissen werden Ihnen viele Sprachelemente und -eigenschaften von Perl bekannt vorkommen, umgekehrt vereinfacht die Kenntnis der Programmiersprache Perl den Umgang mit dem Linux/UNIX-Betriebssystem.

**Ziel:** Mittels vieler praktischer Übungen lernen Sie die fortgeschrittenen Elemente von Perl gründlich kennen und erstellen einfache sowie auch komplexere Programme (Skripte), die gründlich ausgetestet werden. Der Kurs legt viel Wert auf sorgfältig und klar geschriebene Programme und gibt auch Tipps, um trotz der (zunächst) etwas kryptisch erscheinenden Perl-Syntax übersichtlichen Programmcode zu formulieren.

**Hinweis:** Der Kurs ist für Linux/UNIX- und für MS-Windows-Systeme geeignet, als Programmierumgebung wird die Kommandozeile und ein beliebiger Editor verwendet.

**Inhalt:**

- Referenzen
  - Was sind Referenzen?
  - Erzeugen und Auflösen von Referenzen
  - Referenzen auf Skalare
  - Referenzen auf Arrays und Hashes
  - Anonyme Arrays und Hashes
  - Autovivification ("Selbstbelebung")

- Komplexe Datenstrukturen
  - Arrays von Arrays
  - Arrays von Hashes
  - Hashes von Arrays
  - Hashes von Hashes
  - Einsatz von "grep" und "map"
  - Data::Dumper: Dumpen von Datenstrukturen
  - Storable: Speichern/Einlesen von Datenstrukturen
  
- Referenzen auf Subroutinen
  - Anonyme Subroutinen
  - Callbacks
  - Closures
  - Statische lokale Variablen
  
- Module und Pakete (Namensräume)
  - Konzept der Modularisierung in Perl
  - Exportbeschränkung
    - \* @EXPORT
    - \* @EXPORT\_OK
  - Sichtbarkeit von Funktionen und Variablen ("Scope")
  - Schreiben eigener Module
  
- Objektorientierung ("OO")
  - Was ist Objektorientierung?
    - \* Objekte, Klassen, Methoden
    - \* Konstruktor, Destruktor
  - Einsatz objektorientierter Module
  - Erstellen eigener objektorientierter Module
    - \* bless
    - \* Instanz-Variablen und -Methoden
    - \* Polymorphie
    - \* Vererbung ("Inheritance", @ISA, SUPER)

- \* Getter- und Setter-Methoden
- \* Klassen-Variablen und -Methoden
- \* Mehrfachvererbung ("Multiple Inheritance")
- OO-Module
- Delegation mit UNIVERSAL
- Delegation mit AUTOLOAD
- Sonstiges
  - Debugging und Testing
  - Funktions-Prototypen
  - Programmdokumentation (POD)

### **Schulungsunterlagen:**

- Skript
- Lösungsblätter zu allen Übungen

### **Voraussetzungen:**

- Muss: Allgemeine Computer-Kenntnisse (Tastatur, Maus, Grafische Oberfläche, Editor)
- Muss: Kenntnis einer beliebigen Programmiersprache
- Muss: Kenntnis des Kurses "Perl-Einführung"
- Optional: Kenntnis des Kurses "Perl-Aufbau"
- Optional: Kenntnis einer Integrierten Entwicklungsumgebung (IDE)
- Optional: Linux/UNIX-Kenntnisse
- Optional: Shell-Programmierung-Kenntnisse