Objektorientierung in Java  
am Beispiel des NAO

Verstehen und Anwenden von einfachen und fortgeschrittenen objektorientierten Programmierkonzepten.

Themen (Anfänger):

* Was ist ein Programm?
  + *Aus sequentieller Sicht*
* Was ist Syntax, Warum ist Syntax wichtig?
* Was ist ein einfacher Datentyp?
* Was ist eine Variable?
* Was sind Abfragen und Schleifen?
* Wie kommentiere ich Code?
* Was bedeutet Objektorientierung?
* Was sind Klassen, was sind Objekte?
  + *Zustände / Verhalten*
* Konstruktoren
* Was sind Attribute? *Zustände*
  + *Differenz zu lokalen Variablen*
* Was sind Methoden? *Verhalten*
* Namenskonvention und warum sie wichtig ist
  + Klassen, Attribute, Methoden
* Strings, Arrays

Themen (Fortgeschritten – in Komplexität steigend)

* Was sind Pakete?
* Namenskonvention und warum sie wichtig ist REVISITED
  + Pakete, Klassen, Attribute, Methoden
* Was ist Typecasting?
  + *Implizit, Explizit*
* Vererbung, this, super
* Overriding von Methoden
* Zugriffsmodifikatoren *default, public, protected, private*
* Warum Kapselung?
  + *Kontrollierbare Zustände durch Getter und Setter*
* Speicher-Management
* null
* Pass-by-value VS Pass-by-reference
* == VS .equals()
* Interfaces und Enums
* Die Schlüsselworte *static, final und abstract*
  + *Versch. Bedeutungen von final für Attribute, Klassen und Methoden*
* Generische Datentypen
* Exceptions, try-catch-finally, throw, throws