

# Objektorientierte Programmierung WS 2013/2014 - Aufgabenblatt 7

Prof. Dr. U. Kastens

Institut für Informatik, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn

Ausgabe: 21.01.2014

## Aufgabe 1 (Prototyp-basierte Vererbung)

Das Verzeichnis `blatt7/pwdgen` enthält ein HTML-Dokument, das JavaScript einsetzt um einen Passwort-Generator zu realisieren:



Lagern Sie die Methode `show` der `Passwort`-Objekte in ein neues Objekt aus, das Sie als Prototyp für den `Passwort`-Konstruktor verwenden.

## Aufgabe 2 (Erste Schritte in Scala)

Betrachten Sie das Scala-Programm in `blatt7/first`:

```
trait Log {
  def logMethod(s: String) {
    println("entering method " + s)
  }
  def logConstructor(s: String) {
    println("entering constructor " + s)
  }
}

class Person(ln : String, fn : String, s : Person) extends AnyRef {
  def lastName = ln
  def firstName = fn
  def spouse = s

  def this(ln : String, fn : String) = { this(ln, fn, null) }

  private def isMarried() = {spouse != null}

  def computeTax(income: Double) : Double = {
    if (isMarried()) income * 0.20 else income * 0.30
  }

  override def toString() = {
    "Hi, my name is " + firstName + " " + lastName +
    (if (spouse != null) " and this is my spouse, " + spouse.firstName + " "
     + spouse.lastName + "." else ".");
  }
}

object Willy {
  def main( a : Array[String]) {
    val p1 = new Person("Mann", "Willy", new Person("Mann", "Dora"))
  }
}
```

```

        println(p1)
        println(p1.computeTax(42000))
    }
}

```

Verwenden Sie die Scala-Sprachbeschreibung, um folgende Aufgaben zu lösen:

- Übersetzen Sie das Programm und führen Sie die gegebene main-Methode aus.
- Ergänzen Sie einen parameterlosen Konstruktor, der einen unverheirateten "Max Mustermann" erzeugt.
- Verwenden Sie `logMethod()` des Mixins `Log` um Log-Ausgabe für die Methoden der Klasse `Person` zu erzeugen.
- Was bedeutet `extends AnyRef`? Was müssen Sie in der Lösung c) ändern, wenn Sie `extends AnyRef` streichen?
- Verwenden Sie `logConstructor()` des Mixins `Log` um Log-Ausgabe für alle drei Konstruktoren der Klasse `Person` zu erzeugen.

### Aufgabe 3 (Case Classes in Scala)

- Erläutern Sie das Konzept der Case Classes in Scala.
- Das Scala-Programm in `blatt7/caseclass` nutzt Case Classes zur Modellierung ganzzahliger Terme:

```

abstract class Term
case class Num(x:Int) extends Term
case class Plus(x:Term, y:Term) extends Term
case class Times(x:Term, y:Term) extends Term
case class Minus(x:Term, y:Term) extends Term
case class Div(x:Term, y:Term) extends Term

object MainClass {
    def eval(t: Term) : Int = // ergänzen ...

    def main( a : Array[String]) {
        val t1 = Times(Num(10), Plus(Num(2), Div(Num(5), Num(2))))
        println (eval(t1));
    }
}

```

Ergänzen Sie die fehlende Implementierung der Methode `eval`.