

Lab Report Nr. 1

Lab1 hatte für uns viele kryptische Schichtungen. Mit dem Konzept von Git und vor allem dem Einrichten von Ruby on Rails und dessen Gems musste viel nachgefragt und nachgelesen werden. In der ersten Übung konnten wir Git relativ gut einrichten und haben herausgefunden, dass es für Git eine GUI gibt, welches vieles vereinfacht. Ruby on Rails machte uns allerdings mehr Sorgen. Das gesamte Konstrukt mit Ruby on Rails und dessen Gems auf unseren PC's zum laufen zu bringen und wie die Tests funktionierten hatte für uns Ausmaße die einer 5 stufigen Schnitzeljagd glichen. Da die Übungen aber diesmal zweiwöchentlich sind, konnten wir vieles in der zweiten Übung klären.

Für die Programmieraufgabe mussten wir durch eine relativ große Anpassungsphase. Die Syntax von Ruby war in unseren Augen ungewöhnlich. Da Ruby jedoch eine Script-Sprache ist, konnte man viel rumprobieren, ob etwas funktionierte oder nicht.

Und keine Semikolons setzen zu müssen fühlt sich an als ob Gott seine Musen auf unseren Laptops verteilt hätte.

Das Konzept von Getter und Setter war in ziemlich das selbe wie in Java

```
def name
  @name
end

def name=(value)
  @name = value
end
```

Jedoch gibt es vieles was in Ruby neu für uns war, wie die Intersektion von Arrays (es gab natürlich auch Methoden, die uns schon bekannt waren, wie z.B.: .split)

```
def commonHobbies(p2)
  @hobbies & p2.hobbies
end
```

Auch das Iterieren ist sehr ähnlich von dem was wir gelernt haben

```
for p1 in persons
```

Was uns aber ziemliche Probleme bereitet hat, war es einen Error zu raisen.

Hierfür mussten wir eine eigene Klasse erstellen und ein Array der Attribute Accessors erstellen um dieses mit den Initialize-Attributen zu vergleichen damit auch bei neuen Attribute Accessors kein Error geworfen wird.

```
def initialize(attributes = {})
  key_array = attributes.keys
  attr_array = Person.public_instance_methods(false).select {|m|
    Person.public_instance_method(m).arity.zero?
  }
  if (key_array - attr_array).size > 0
    raise UnknownAttributeError
  end
  attributes.each {|key, value| send "#{key}=", value}
end
class UnknownAttributeError < StandardError
end
```