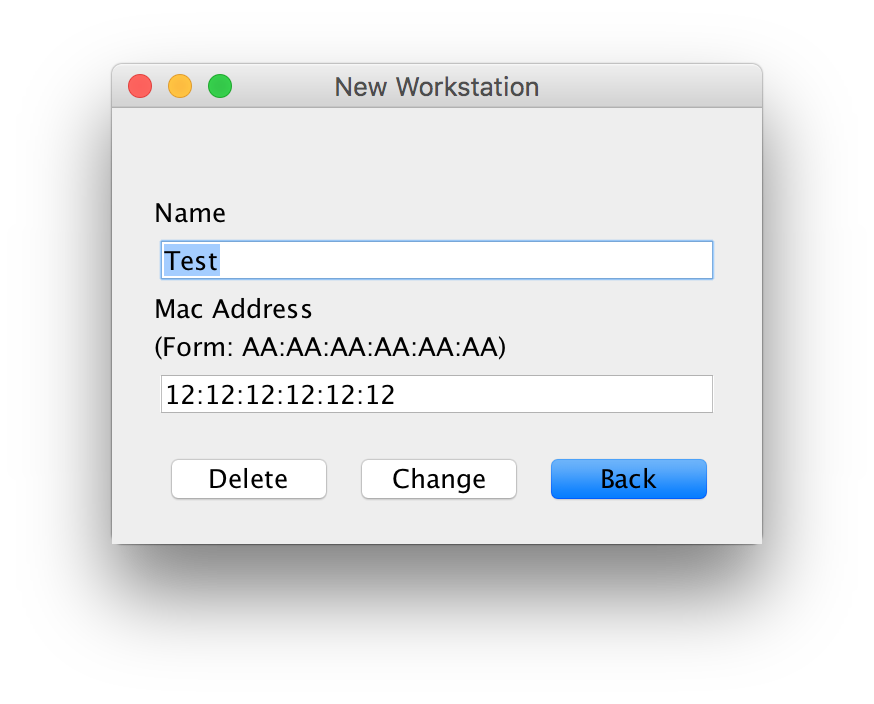
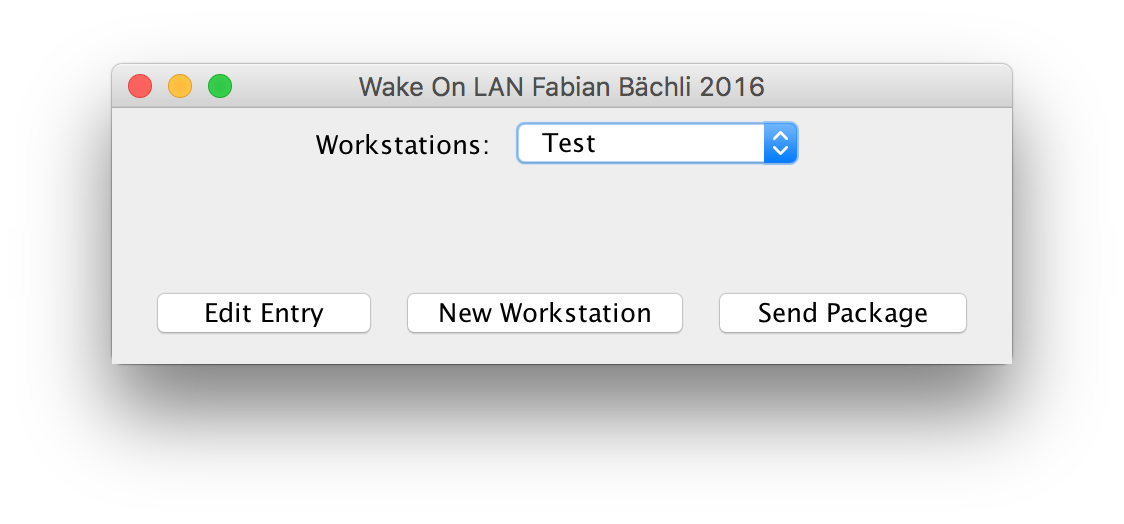
# Projektdokumentation M226

## Einführung

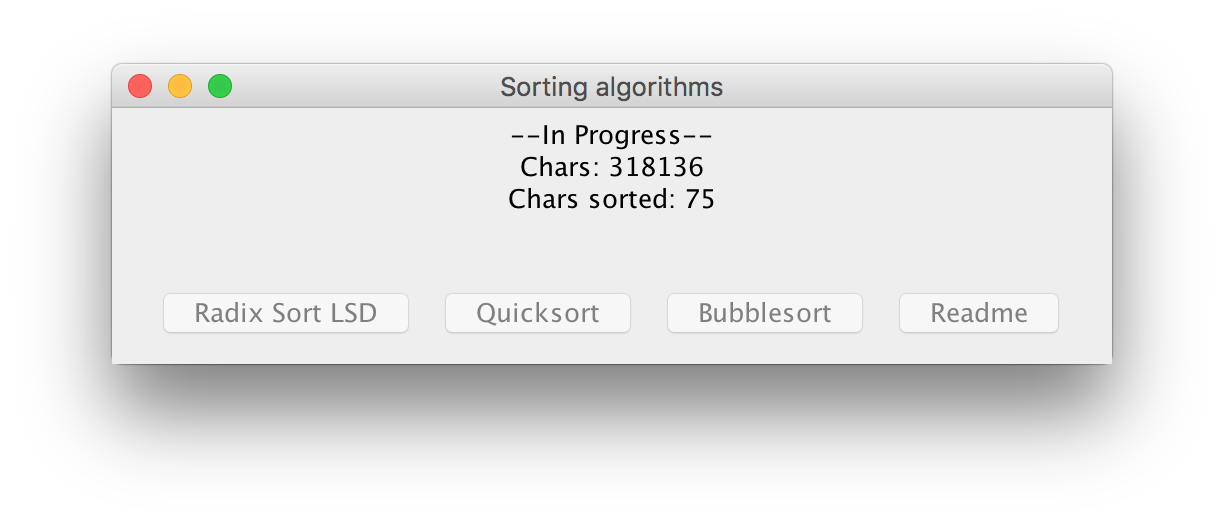
Vor diesem Modul programmierte ich schon mit Java. Ein Projekt, welches ich bereits realisiert habe ist ein Programm, mit welchem Wake On LAN Pakete an MAC Adresse senden konnte. Das Programm konnte Arbeitsstationen speichern, sodass man die MAC Adresse nur einmal eingeben musste. Dazu musste ich einen DOM Parser (XML File Parser) anwenden. Ausserdem hatte das Programm eine grafische Oberfläche. Das Endprodukt sah so aus:



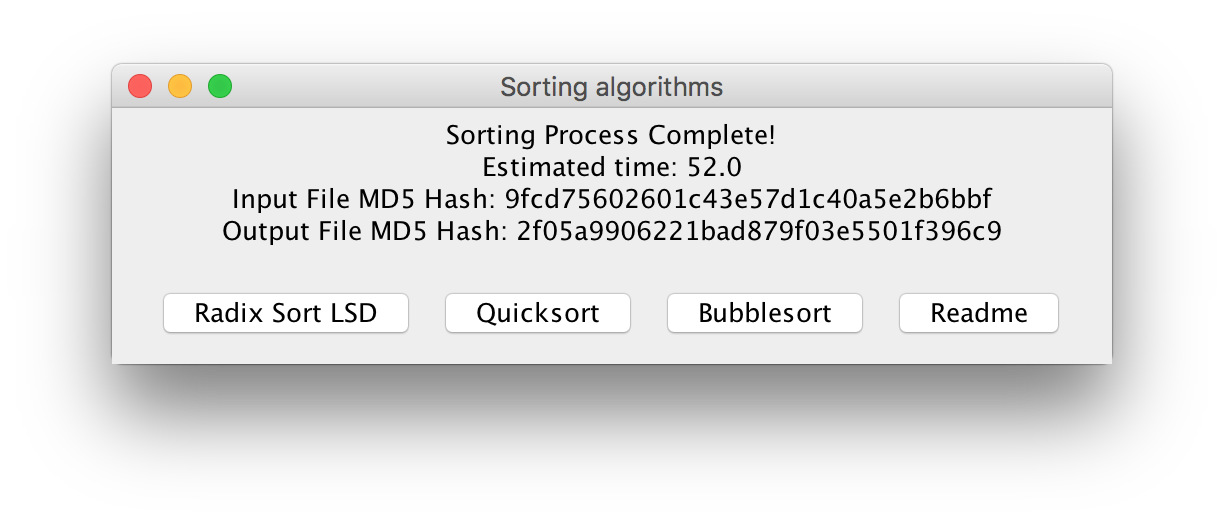
Ich hatte also schon zumindest gewisse Grundkenntnisse

## 1. Sortieralgorithmus

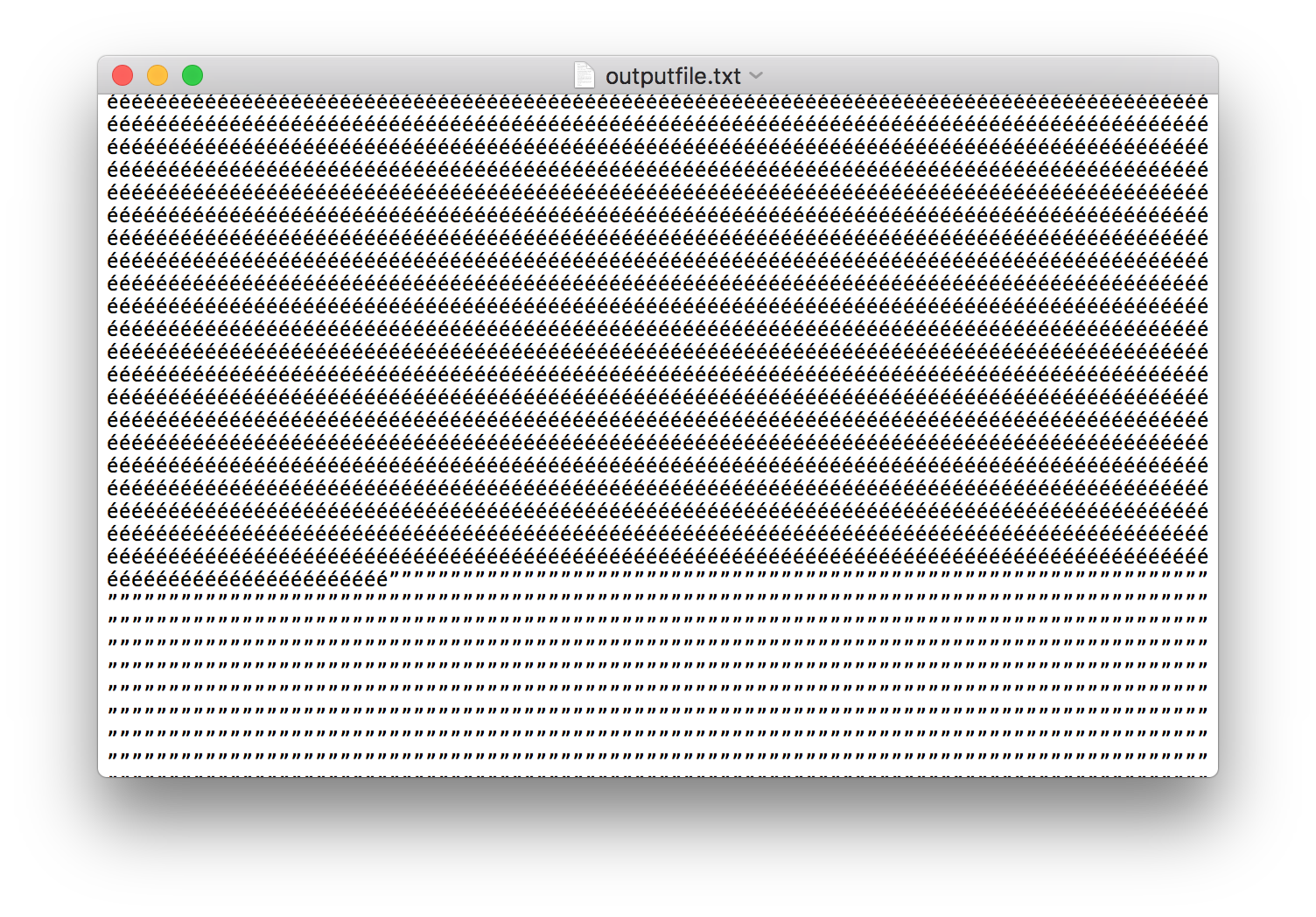
Als Erstes wagte ich mich an Etwas, was mir noch einigermassen bekannt war. Ich schrieb ein Programm, welches ein Inputfile parst und die Charakteren nach ihren Unicode Values sortiert. Ich implementierte die Algorithmen Bubblesort, Radix Sort Least Significant Digit und den Quicksort. Den sortierten Output schreibe ich in ein Outputfile.txt. Es wird jeweils die Zeit gemessen, welche der jeweilige Algorithmus braucht, um den gesamten Input zu sortierten. Am Ende wird ein MD5 Hash des Outputfiles erstellt, um zu validieren, ob die Algorithmen jeweils auf denselben Output kommen. Beim Bubblesort konnte ich ausserdem ausgeben, wie viele Charakteren schon sortiert sind. Dies ist bei den anderen Algorithmen leider nicht möglich, weil diese nicht linear vorgehen. Hier das Programm, während es den Bubblesort am Inputfile anwendet:



Am Ende eines Sortiervorgangs:



Und das Outputfile:



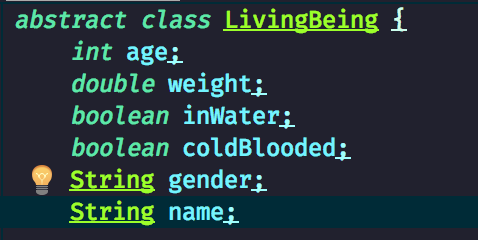
### Gelerntes

Was ich in dieser Aufgabe lernte, waren sicher zum einen, die neuen Algorithmen (Den Radix Sort und den Quicksort kannte ich davor noch nicht) zum anderen, dass es andere Wege gibt, ein GUI zu erstellen, als Java Swing. Ich erstellte dieses GUI nämlich in Java FX. Abgesehen davor barg diese Aufgabe nicht allzu viel Neues und ich fühlte mich eigentlich ziemlich sicher die ganze Zeit.

Ein Ort, an dem ich Probleme hatte, war das schnelle Updaten, des Labels, welches den Fortschritt des Bubblesort zeigt. Dies war nicht möglich, bis ich einen separaten Thread auf der Methode anwendete. Ein Problem, welches mir neu war.

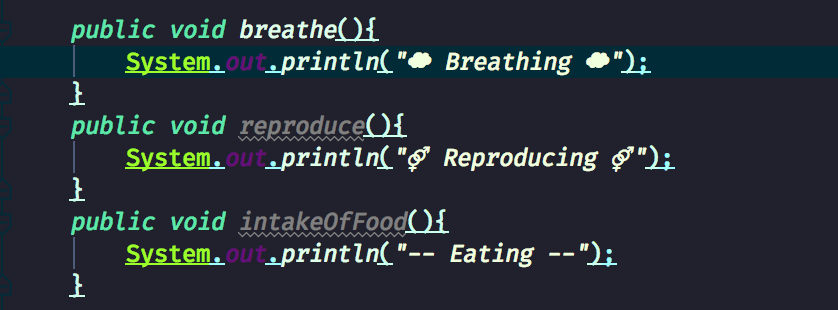
## 2. Diverse Klassen

Als Zweites machte ich ein Programm, welches Vererbung beinhalten sollte. Ich entschied mich dazu für eine Konstellation, in welcher es die Superclass LivingBeing gab. Alle Attribute von LivingBeing, sollten in den erweiternden Klassen danach auch vorkommen:



Zu jedem Attribut kam ausserdem ein Getter und Setter.

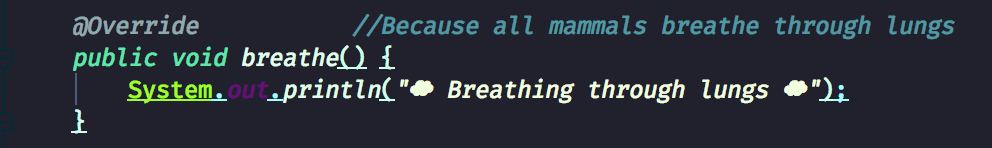
Ausserdem definierte ich Methoden, welche alle anderen, erweiternden Klassen von LivingBeing benutzen sollen können:



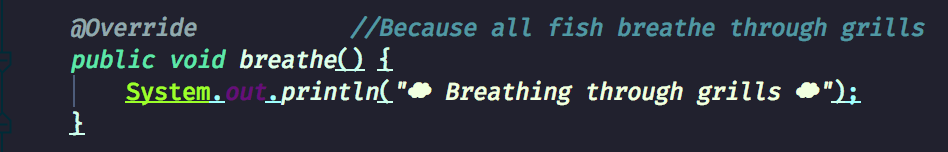
Danach kreierte ich die erweiternden Klassen von LivingBeing namens Mammal und Fish.

Ich überschrieb die Methode breathe().

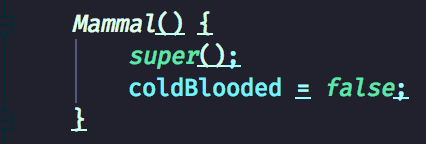
Bei Mammal:

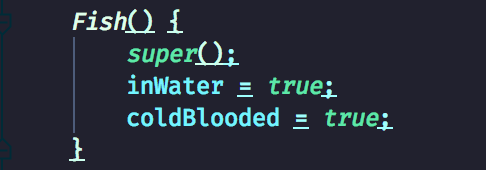


Und bei Fish:



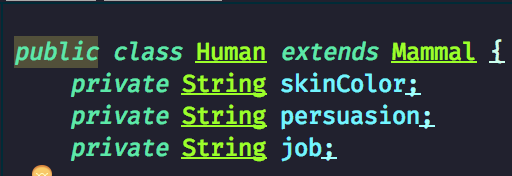
Ausserdem erstellte ich für beide Klassen Konstruktoren, welche die Attribute, die immer gleichbleiben, setzten:



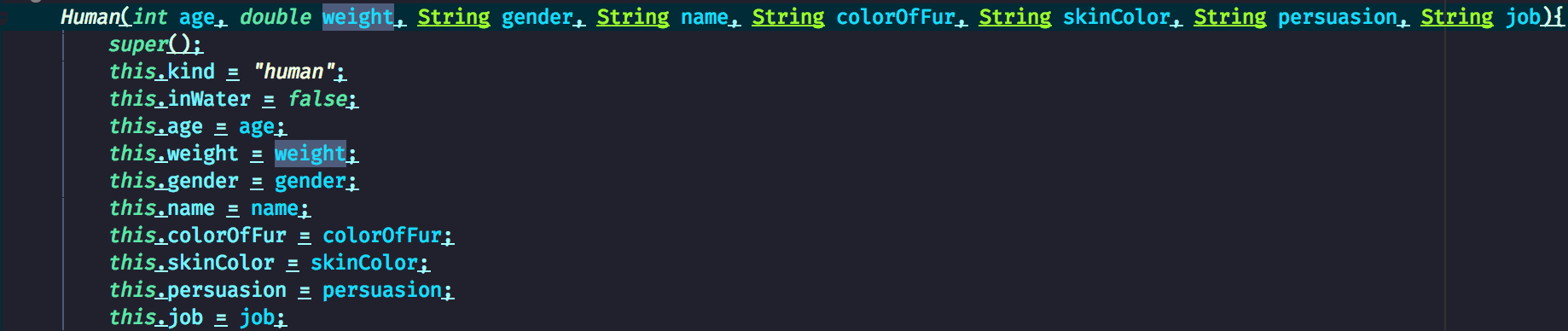


Bei beiden Klassen kamen jeweils für Fische und Säugetiere spezifisch Attribute hinzu. Z.B bei Fish, colorOfScale (Schuppen). Mammal und Fish sind immer noch abstrakt.

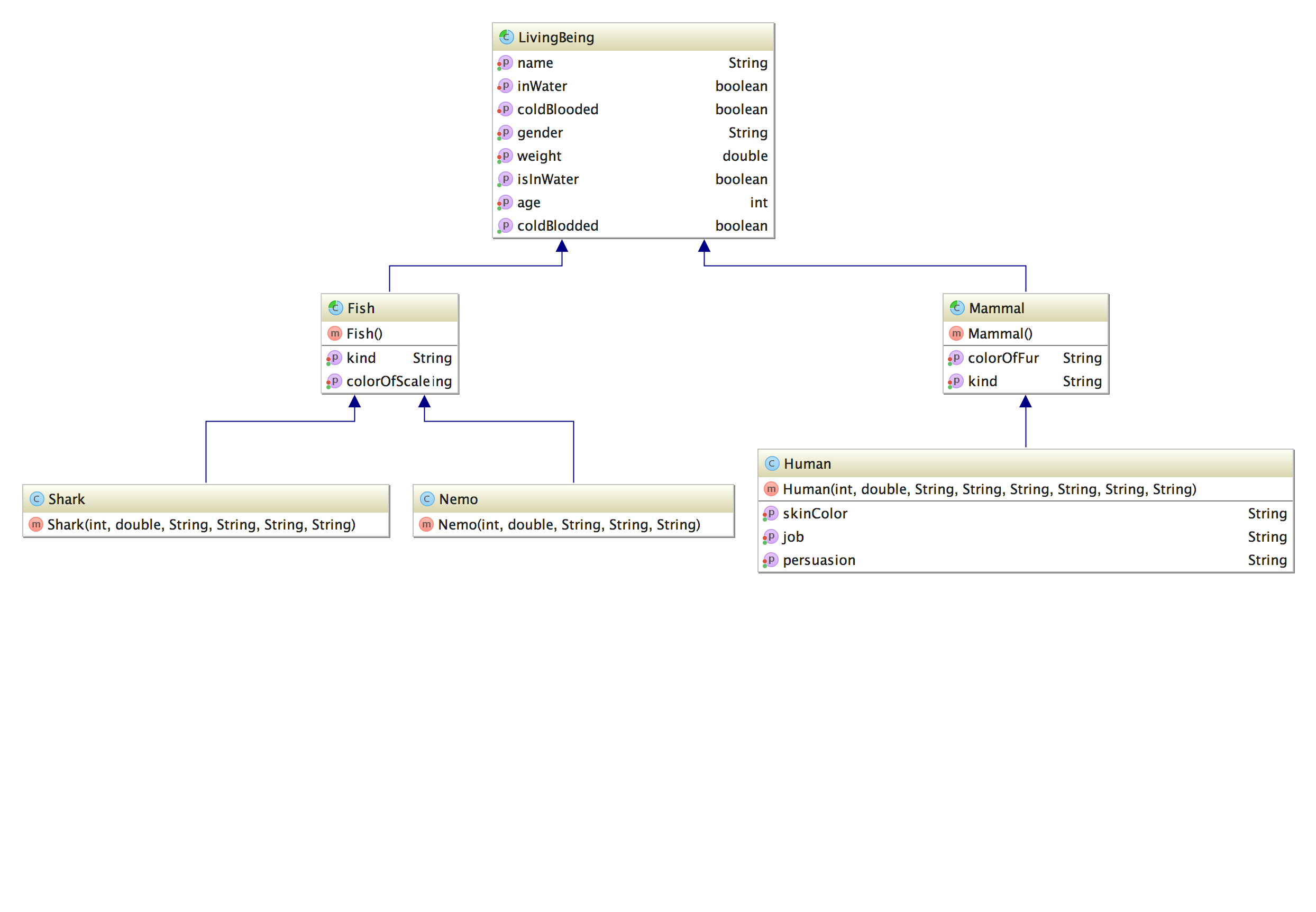
Als erweiternde Klasse von Mammal kreierte ich Human, welche Mensch-spezifische Attribute hat:



Und einen Konstruktor:



Später kamen zu der Klasse Fish noch Nemo und Shark hinzu. Nemo hatte die nemo-spezfische Methode „findDory“ und Shark hatte „electroPerception“ (Haie können elektromagnetische Felder spüren). Hier ist das schlussendliche Klassendiagram:

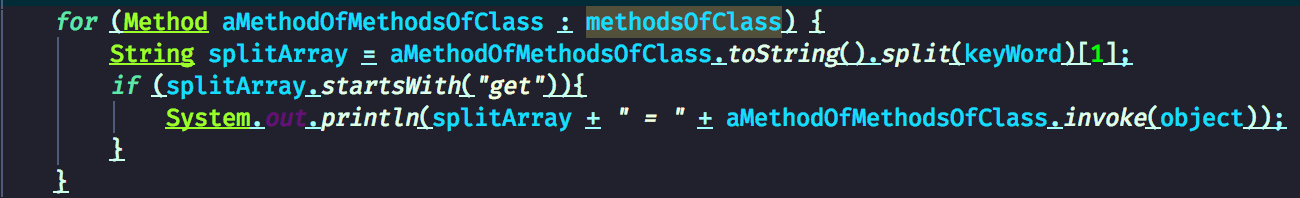


Als letztes und für mich kompliziertestes in diesem Task, erstellte ich eine Controller Klasse. Diese Klasse sollte alle Attribute von einem Objekt in die Konsole schreiben und das Alter von einem Objekt um eine gegebene Zahl erhöhen können.

Bei der Methode, welche alle Attribute eines Objekts zurückgeben sollte ging ich so vor:

## ../Desktop/Screen%20Shot%202016-10-03%20at%2007.21.03.png

Mit isinstanceof schaue ich, welcher Klasse das Objekt angehört. Danach fülle ich alle Methoden der jeweiligen Klasse in ein Array namens methodsOfClass ab und setzte das Key Word.



Als nächstes, gehe ich alle Methoden des Arrays durch und schaue, ob die Methode mit „get“ beginnt. Danach führe ich die Methode mit invoke auf dem Objekt aus.

Bei der Methode, welche das Alter von einem Objekt erhöhen kann, bin ich ähnlich vorgegangen, nur dass ich bei dieser Methode zuerst das Alter des Objekts rausfinden muss hier der Code:

