

Arbeitsblatt zum Thema »Verzweigungen«

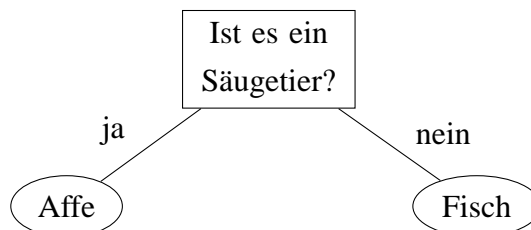
Mit Hilfe des Programms *Tiererraten* soll das Mobiltelefon in die Lage versetzt werden, durch Nachfragen herauszufinden, welches Tier sich die Nutzerin bzw. der Nutzer vorgestellt hat.

Aufgabe 1

- Wieviele verschiedene Tiere kann das Programm erraten, ohne dass Sie zu Beginn eine Datei laden?
- Nehmen wir an, Sie bekommen eine Datei zur Verfügung gestellt und bei einem Probelauf werden Ihnen acht Fragen gestellt. Wieviele Tiere kennt das Programm damit mindestens?

Aufgabe 2

Sie laden die Datei `simpleAnimaltree2.xml` bei Programmstart. Stellen Sie die enthaltenen Daten grafisch dar. Verwenden Sie das folgende Beispiel als Anregung.



Aufgabe 3

- Sie erweitern die Daten um (mindestens) eine weitere Tierart. Wie muss eine Frage formuliert sein, damit Sie sich sauber in das Spielkonzept einbettet?
- Wieviele Tierarten können höchstens erraten werden, wenn maximal 10 Fragen erlaubt sind?

Aufgabe 4

Sie wollen Tiere mit zwei, vier, sechs und acht Beinen unterscheiden. Als Antwortmöglichkeiten stehen aber nur *Ja* und *Nein* zur Verfügung. Wie gehen Sie vor?

Aufgabe 5

Erstellen Sie selbst ein einfaches Programm, welches ebenfalls Tierarten erraten kann. Dabei müssen die Fragen **nicht** innerhalb des Programms **erweiterbar** sein, sondern von Ihnen fest vorgegeben. Ihr Programm soll **mindestens vier Tiere unterscheiden** können.

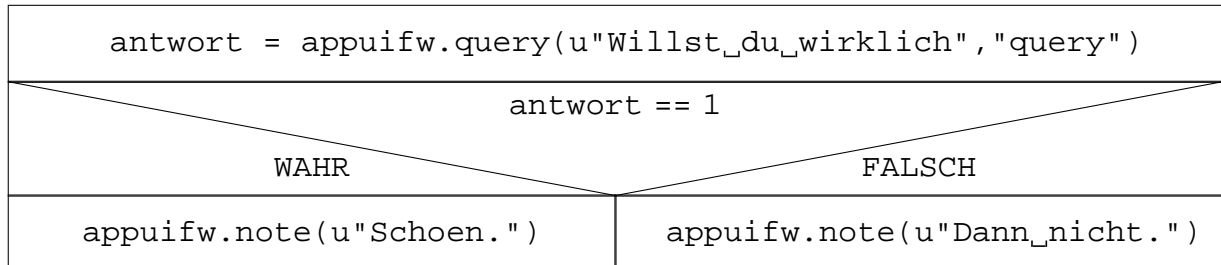
Das appuifw-Modul stellt eine Methode zur Verfügung, die von dem Nutzer bzw. der Nutzerin eine Bestätigung erfragt. Wird die Meldung bestätigt, so wird eine 1 zurückgegeben, wird stattdessen der Softkey zum Abbruch gedrückt, wird None zurückgegeben.

```
antwort = appuifw.query(u"Willst_du_wirklich?", "query")
```

Mit Hilfe einer Programmiersprache kann eine **Bedingung** überprüft werden. Ist die Bedingung erfüllt, reagiert das Programm anders, als wenn diese nicht erfüllt ist (**Verzweigung** des Programtablaufs).

```
if antwort == 1:
    appuifw.note(u"Schoen.")
else:
    appuifw.note(u"Dann_nicht.")
```

Stellen Sie vor der Programmierung die von Ihnen benötigten Verzweigungen mit Hilfe eines **Struktogramms** dar. Für das hier gezeigte Beispiel würde ein Struktogramm wie folgt aussehen:



Aufgabe 6

Die Frage nach der Anzahl der Beine kann – wie oben gezeigt – durch Entscheidungsfragen umgesetzt werden. Man kann diese aber auch als Zahl von der Benutzerin bzw. dem Benutzer erfragen. Die Methode query aus der letzten Aufgabe bekommt dazu einen anderen Parameter.

```
antwort = appuifw.query(u"Wieviele_Beine?", "number")
```

Stellen Sie in einem Struktogramm dar, wie ein Programm auf unterschiedliche Eingaben der Benutzerin bzw. des Benutzers reagieren soll. Setzen Sie diese Ausschnitt dann innerhalb Ihres Programms der vorherigen Aufgabe um.