

Dokumentation zur Aufgabe „Baywatch“

Die Lösungsidee:

Um an eine vollständige Liste mit Start im Norden zu gelangen, muss man die zwei Listen mit einander abgleichen. Man „dreht“ dabei die Liste mit unbekanntem Startpunkt solange, bis die bereits bekannten Felder der Liste mit Start im Norden, an der selben Stelle auch auf der vollständigen Liste mit unbekanntem Startpunkt liegen.

Die Umsetzung mit Quelltext:

Die Umsetzung der Lösungsidee erfolgt in meiner Bearbeitung in der Programmiersprache Ruby. Das Programm beziehungsweise Script („baywatch.rb“) liegt im Ordner „Junioraufgabe 2“ und stellt zeitgleich Quelltext und ausführbare Datei dar. Ich habe das Script mit der Ruby Version 2.5.3-1-x64 in der CMD von Windows 10 getestet.

```
puts "Gebe den Namen einer .txt-Datei mit Landzungenlisten ein (z.B. baywatch6.txt)"
file = gets.chomp
lists = File.open(file).read.force_encoding("UTF-8").split("\n")
list1 = lists[0].split(" ")
list2 = lists[1].split(" ")
```

Zunächst werden die Listen eingelesen. Die Einträge der zwei Listen werden in die Arrays *list1* und *list2* geteilt. *list1* beinhaltet dabei die vollständige Liste mit unbekanntem Startpunkt und *list2* die unvollständige Liste mit Start im Norden.

```
i = 0
a = 0
searching = true
finalList = ""
while i < list1.length && searching do
  matches = true
  while a < list1.length do
    if not list1[a] == list2[a]
      if not list2[a] == "?"
        matches = false
      end
    end
    finalList = finalList + list1[a].to_s + " "
    a = a + 1
  end
  a = 0
end
```

In diesem Code Abschnitt gibt es folgende Variablen:

i ist ein Integer, welcher in einer while Schleife Hochgezählt wird, bis er so groß ist, wie die Anzahl an verschiedenen Inselabschnitten, welche sich im Array *list[0]* befinden. Da nach einer Lösung gesucht wird, bei der zwei Listen übereinander gelegt werden müssen, kann es nicht mehr Möglichkeiten zum Vergleichen der Listen geben, als es Inselabschnitte gibt. Folglich kann die Suche nach der Lösung abgebrochen werden, wenn die Anzahl der Versuche mit der Anzahl der Inselabschnitte übereinstimmt.

a ist ein Integer, der benutzt wird um in der zweiten while Schleife zu gucken, ob jede der Angaben aus der vollständigen Liste mit denen aus der unvollständigen Liste übereinstimmt, wobei die unbekannten Felder nicht kontrolliert werden, indem sie durch die zweite if Verzweigung ausgeschlossen werden.

Am Anfang der ersten while Schleife wird der Boolean *matches* auf true gesetzt. Anschließend werden in der zweiten while Schleife die zwei Listen darauf kontrolliert, ob sie an jeder Stelle übereinstimmen. Wenn nun eine Stelle nicht übereinstimmt, die in der unvollständigen Liste bekannt ist - also kein ? ist - so wird der Boolean *matches* auf false gesetzt. Sollte am Ende der zweiten while Schleife also der Boolean immer noch auf true stehen, so stimmen die beiden Listen überein und man hat die vollständige Liste mit Beginn im Norden gefunden.

Der Boolean *searching* steht auf true, bis die gesuchte Liste gefunden wurde. So kann die erste Schleife beendet werden, welche nur weiterläuft, wenn *searching* auf true steht.

Der String *finalList* beinhaltet am Ende der zweiten while Schleife den Inhalt des Arrays *list1*.

Dieser wird in jedem durchlauf der ersten while Schleife um eine Position verschoben.

