

Terminplanung Termin 1

[19.02.26]

- Hinweis auf Text **KI-00 Messung von Intelligenz.pdf** (nachholen)
- Hinweis auf GI Artikel „Ohne Umwege“ von 09'24 zu Dijkstra (nachholen)
- Werkzeug **Python** Vergleich mit Scheme
 - Link python.org
 - Ziel: Auch bei Python das „Funktionale Programmierparadigma“ kennen lernen sowie die Probleme bei dessen Anwendung
 - In diesem Kurs also Wiederholungsstrukturen Rekursion und Iteration

Kursplanung und Bücher Hinweis

- Präsentation **P01 Kurs Intelligente Suchverfahren.pdf** bis Folie 6
- sehr viel Material in Büchern zur KI, auch aktuell zu kaufen
- speziell:
 - Volker Turau: Algorithmische Graphentheorie
<https://www.ti5.tuHH.de/staff/turau/graph/>
 - Peter Gritzmam und René Brandenberg: Das Geheimnis des kürzesten Weges
 - Gallenbacher: Abenteuer Informatik
 - Ziegenbalk: Algorithmen von Hammurapi bis Gödel
- Bücher Python:
 - Ernesti, Kaiser: Python 3

Was heißt funktionale Programmierung ?

- Präsentation **P00-c Funktionen-bei-Python.pdf**

Einstieg in die Arbeit

Das einfache Rucksackproblem

- Präsentation **P-KI-Suchprobleme-Präsentation.pdf**
 - Hinweis auf Präsentation **P01a-KI Vereinfachung nutzen.pdf**
 - Hinweis auf Text **KI-01-a Vereinfachung nutzen.pdf**
 - Hinweis auf Arbeitsblätter **KI-00-d Arbeitsblatt schrittweise Erarbeitung.pdf** **KI-01-b1 Stuecke zum Ausschneiden.pdf** und **KI-01-b2 Stuecke groessengerecht.pdf** (nachholen)
- Programm entwickeln nach dem Konzept der schrittweisen Entwicklung dazu Präsentation **P01b fuelle-schrittweise-Erarbeitung.pdf** (ist tiefe Liste)
 - Projekte im Ordner *fueller-schrittweise-tiefe-Liste* ,
Beginn mit *fueller-Schritt-0.py* ,
 - weiter mit den nachfolgenden Programmen (*im Kurs mit der Kursversion fueller-kurs-26.py umgesetzt*)
 - dabei spezielles Eingehen auf die Version ohne elif und else
- ~~die einzelnen Schritte noch einmal erläutern mit der Präsentation **P02-1 Binaerbaum Container 80.pdf** (nachholen)~~
- Programm-Variante mit flacher Liste ansehen (nachholen)
- Warum ist fuelle so einfach?
 - Einsatz von *fueller-greedy-rekursiv-tiefe-liste-aufrufprotokoll.py* (*im Kurs mit der Kursversion fueller-kurs-26.py umgesetzt*),
 - Eingehen auf den Stack: Problem Endrekursion wird nicht erkannt!

Was fehlt?

- Testen des Programms mit Containergröße 100 (*außerdem ein Teilnehmerinnen-vorschlag [20,20,20,30] mit dem gleichen Ziel*) zeigt Misserfolg – warum?
- Erarbeitung der Lösung zu Containergröße 100 ergibt das Programm *fueller-Schritt-5-ts.py* (im Kurs mit der Kursversion *fueller-kurs-26.py* umgesetzt)
- Präsentation **P02 Rucksack packen mit TS-Py.pdf**
- Testen der Lösung mit *fueller-ts-mit-Aufrufprotokoll.py*
- dazu Präsentation **P02-2 Binaerbaum_Container_100.pdf** (nachholen)
- noch „einkürzen“ zu *fueller-tiefensuche-ganz-kurz.py*

Verallgemeinerung des Vorgehens bei der Tiefensuche mit den beiden (*altes Design*) Präsentation

- Präsentation **P01-e Tiefensuche mit backtracking Programm.pdf** mit der Auflistung der Verzweigungen und
- Präsentation **P01-d Tiefensuche-mit-backtracking.pdf** mit der schematischen Darstellung des Suchgraphen
- Hinweis auf Text **KI-02-a Stack bei Tiefensuche.pdf** (nachholen)
- Hinweis auf Text **KI-02-b Aufrufprotokoll_Tiefensuche.pdf** (nachholen)

Allgemein Python

- Übungen zur Entwicklung mit slices:
 - *summe-Erarbeitung.py*
dazu Präsentation **P00-e Summe rekursiv mit Python.pdf**
 - ~~*sortieren-durch-Einfuegen.py*~~
dazu Präsentation **P00-f Sortieren durch Einfuegen rekursiv.pdf**
(Kohärenz „jede ... nur für eine Aufgabe...“ ansprechen! – schrittweise!)
und zum Vergleich
 - ~~*Sortieren durch Einfuegen iterativ.py*~~ zeigen
 - Angebot: Präsentationen aus OO zu Listen, Tupel, Strings usw

weitere Übungen ????

- ~~Übungen (Präsentationen in „Aufgaben zu Grundlagen“) zu Operationen auf Listen:~~
 - ~~Länge einer Liste~~
 - ~~Einfügen eines Elements in eine Liste an verschiedenen Positionen~~
 - ~~Entfernen eines Elements aus einer Liste~~
 - Umkehrliste