

Übungen zur Entwicklung mit Rekursion:

- *summe-Erarbeitung.py*
- *sortieren-durch-Einfuegen.py* (Kohärenz ansprechen! - schrittweise!) und zum Vergleich
- *Sortieren-durch-Einfuegen-iterativ.py* zeigen
- ~~Angebot: Präsentationen aus OO zu Listen, Tupel, Strings usw~~
- Hinweis auf die beiden iterativen Versionen von *fuelle*

nachholen

- Warum ist *fuelle* so einfach?
 - Einsatz von *fuelle-greedy-rekursiv-tiefe-liste-aufrufprotokoll.py* ,
- Software zeigen mit *Binaerbaum_vierstufig.graph*
- dazu Präsentation **P02-1 Binaerbaum_Container_80.pdf**
- ~~Eingehen auf den Stack: Problem Endrekursion wird nicht erkannt!~~
- ~~Rekursionsproblem~~ bei Python zeigen mit Vergleich *rekursionsproblem.py* und *keinRekursionsproblem.sem*
- ~~Programmversion (und Begriff) zu flacher Liste ansehen, zwei Varianten~~
 - Fassungsvermögen extern **nachholen**
 - Fassungsvermögen am Kopf der flachen Liste

Weiter mit *fuelle TS* (z.T. Wiederholung)

- Testen des Programms mit Containergröße 100 führt auf Erweiterung: Besprechung der (beim letztenmal erarbeiteten) Lösung zum Programm *fuelle-tiefensuche-tiefe-Liste.py*
- Programmierläuterung in Präsentation **P02 Rucksack packen mit TS-Py.pdf**
- Testen der Lösung mit *fuelle-ts-mit-Aufrufprotokoll.py*
- dazu Präsentation **P02-2 Binaerbaum_Container_100.pdf**
- Hinweise auf
 - 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
- Text einsetzen **KI-02-a Stack bei Tiefensuche.pdf** dazu gehört *fuelle-tiefensuche-aufrufprotokoll-wirdZuVoll.py*
- Hinweis auf Text: **KI-02-b Aufrufprotokoll_Tiefensuche.pdf**

Fuelle viele

- Auf Vorschlag eines Kollegen entwickeln wir gemeinsam eine Version mit zwei Containern *fuelle-tiefensuche-zwei-Kurs.py* auf der Basis von *fuelle-tiefensuche-ganz-kurz.py* mit Austesten des Ablaufs durch Anwendung von `input(...)`.
- Erste Überlegungen der KuK zum Berechnungsaufwand
- Schlussbesprechung mit der Ankündigung der Erweiterung auf mehrere (*beliebig viele, also Liste einsetzen*) zu füllende Container