

Wegeproblem kürzester Weg

- Schritt zur Programmierung bedeutet bei komplexen Problemen Modularisierung:
 - Datenstruktur entwickeln
 - Entscheidung Knotenliste oder Kantenliste für unsere Entwicklung: Kantenliste
- Präsentation **P03-d1 Dijkstra Programm Modularisierung-Kanten.pdf** (->Grafik)
- Präsentation **P03-c2 Dijkstra Funktionale Analyse_Kanten.pdf** (→ Text)

Zur Wiederholung (überarbeitet mit neuem Material)

- Aufgabenfolien (im Programmeordner)
 - Aufgabe **Prioritaetswarteschlange.pdf**, dazu
 - *prioritaetswarteschlange_Erarbeitung.scm* und
 - *prioritaetswarteschlange_Kurs.scm* als Ergebnis
 - Aufgabe , dazu
 - *aktualisiere-besuchliste_Erarbeitung.scm* und
 - *aktualisiere-besuchliste_Kurs.scm*

neu

- Aufgabenfolie
 - Aufgabe **wege.pdf**, dazu
 - *wege_Erarbeitung.scm*
 - Erarbeitung mit dem Kurs ergibt *wege_Kurs.scm*
- Kantenliste in *gallenbacher-graph-kanten.scm* mit Zugriffsfunktion (*nur ansehen!*)
- Vergleich mit Knotenliste → Assoziationsliste ansprechen, dazu Datei *gallenbacher-graph-knoten.scm*
- Präsentation **P03-e Gallenbacher-Graph-Entwicklung.pdf** (*ist korrigiert*)
- Testen des Programms *dijkstra-kanten.scm* mit verschiedenen Start- und Zielorten.
 - Bewertung der Qualität
 - Einordnung von Dijkstra: Präsentation **P03-g Dijkstra Typisierung.pdf**
- Hinweis auf Material zu Dijkstra-Kanten-kleiner-Graph.zip
→ Präsentation **P03-e3 Dijkstra-Schritte kleiner-Graph**
- Präsentation **P03 Wegeprobleme.pdf**

Minimal spanning tree

- Hinweis: Zwei Algorithmen (**Prim** und **Kruskal**), im Material zu Anforderungen nur Kruskal
- Präsentation **P04-1 Algorithmus von Kruskal.pdf**