Termin 13 (Stichworte)

Wiederholung und Nachholen

- Problem mit Modul pickle bei Einbau der Füllfarbe, daraus folgt die Idee:

Daten lesbar speichern

- ***Arbeitsplan-18 Daten lesbar speichern.pdf***
- Vorbereitung durch *Allgemein-Oeffnen-Speichern-mit-Dialog.zip*
- Präsentation ***OO-Python-P08-d Daten lesbar speichern.pdf***
- Projekt: *Raumplaner-Daten-lesbar* (in 08)
  (basiert auf *Raumplaner-Gui-Maus-und-Fuellfarbe*)

Bearbeitung:

- Die Daten sollen in lesbarer Form, also textlich gespeichert werden
- Dazu muss Moebel sie bereitstellen (*als Tupel gewählt*)
- RaumplanerModell sammelt sie in einer Liste und
  - speichert sie in dieser Form ab (allerdings als ein einzelner String)
  - stellt die Objekte aus der Liste nach dem Lesen wieder her
- Projekt *Raumplaner-Daten-lesbar-Kurs-zwei-Schritte.zip*
- Austesten des Lesens der Daten mit einem Texteditor und des Bearbeitens.

MVC

Theorie

- Trennung der Modellkomponente von ihrer Darstellung ist grundlegendes Konzept:
  Model-**V**iew-**C**ontroller – Konzept;
  Präsentationen***OO-Python-P09-a0-Modellklasse vs Controller.pdf*** und
  OO-Python-P09 MVC.pdf

Bearbeitung:

- Präsentation ***OO-Python-P09-a Einbau-Controller.pdf*** (*noch nötig zu zeigen?*)
- ~~Projekt: *Raumplaner-Controller-Name_und_Entkopplung*~~ Bitte noch ansehen!
  ~~(basiert auf *Raumplaner-Daten-lesbar*)~~
  - ~~Die Klasse RaumplanerModell wird zu RaumplanerController umbenannt~~
  - ~~Die Methode für die Verarbeitung der Koordinaten zur Position kommt in die Zeichenflaeche~~
  - ~~Gui und Controller bekommen das Zeichenflaeche-Objekt von der App zugewiesen~~
  - ~~u.a. dadurch Entkopplung der Controllerklasse von wx~~

~~Abwägung zum Ziel: **MVC konsequent umsetzen** (*??? sinnvoll ???*)~~

- ~~Zeichenflaeche hält nicht mehr die Objekte selbst~~
- ~~Alle Anforderungen gehen über den Controller~~

Das Problem mit der Füllfarbe

- Projekt *Raumplaner-Controller-Fuellfarbe.zip*
  (basiert auf *Raumplaner-Controller-Name_und_Entkopplung*)
- Eine Lösung:
  - GibFigur wird ersetzt durch GibFiguren
  - Die Klasse Grafikfenster wird überarbeitet:
    sie holt sich (in Draw) die Figuren von der Kontrollerklasse
  - Alle konkreten Moebelklassen müssen eine Liste von Figuren zurück geben

    - Alternativentwurf: Tupel von Figuren
- Eine zweite Lösung:
  Raumplaner-Daten-lesbar-Figur-neu.zip
    - In GibFigur() werden mehrere Pfade eingesetzt
      Lösung ist einfach bei nebeneinander liegenden Flächen
    - Das Beispiel der Badewanne zeigt, dass die Umsetzung sehr viel schwieriger ist bei geschachtelten Teilfiguren.
- ~~Diskutieren: alternativer Entwurf Teilfiguren mit unterschiedlicher Farbe und Füllfarbe ermöglichen. Die Folge: Tupel aus (<figur>,<farbe>,<fuellFarbe>) wird nicht in der Controllerklasse gebildet!~~

Alternativentwurf mit einer Oberfläche

- Projekt *Raumplaner-eine-Oberflaeche.zip*
  (basiert auf *Raumplaner-Controller-Fuellfarbe*)
    - Hinweis: Konfigurationsdialog zwingend notwendig
- wichtige weitere Änderung: die Klasse Moebel führt selbst kein Update durch, das übernimmt die RaumplanerController-Klasse
- grafikfenster.py enthält nur noch die Klasse Zeichenflaeche
  Umbenennung in zeichenflaeche.py
- Gui enthält nur noch Menue-Elemente und holt sich als panel die Zeichenflaeche
  dazu muss sie in Bearbeiten die entsprechenden Methoden anbieten

~~Grundriss?~~

~~eventuell zeigen:~~
~~- Raumplaner-mit-Grundriss (nicht mit Hintergrundbild!)~~
~~- Überarbeitung von 2025: Grundriss kann nicht angeklickt werden~~
~~- Projekt *Raumplaner-mit-Grundriss.zip*~~

Die nicht „geschafften“ Abschnitte sind durchgestrichen, sind aber sinnvolle Ergänzungen.