

Aufgabe 1

Grafische Benutzeroberfläche zur manuellen Segmentierung in Schichtbild-Datensätzen

Manuelle Segmentierungen von medizinischen Bildern sind notwendige Voraussetzungen, um den Erfolg von (lernbasierten bzw. überwachten) Bildverarbeitungsalgorithmen zur automatisierten Auswertung von medizinischen Bilddaten zu ermöglichen. Vor allem zur Bereitstellung gelabelter Daten durch medizinische Experten sind solche Segmentierungsarbeiten von entscheidender Bedeutung, um damit z.B. Verfahren der künstlichen Intelligenz (welche Lerndaten erfordern) zu entwickeln und zu testen.

In dieser Praktikumsaufgabe sollen mit einer GUI manuelle Segmentierungen für medizinische MRT-Schichtdatensätze durch einen Anwender realisiert werden. Dabei soll der Anwender die Umrisse von Organen schichtweise eingeben können und diese Ergebnisse über „binäre Masken“ (Binärbilder) von den Organbereichen schichtweise abspeichern können. Der Anwender soll weiterhin schichtweise Korrekturen an den bereits segmentierten Organbereichen vornehmen bzw. Zwischenergebnisse verfeinern können.

Hinweis 1: für diese Aufgabe benötigen Sie keine medizinischen bzw. anatomischen Kenntnisse über abdominale Organe. Zumindest fließt das Unwissen über die korrekten Organbereiche nicht in die Benotung ein! Die Beispielbilder zeigen die Organbereiche nur zur Illustration.

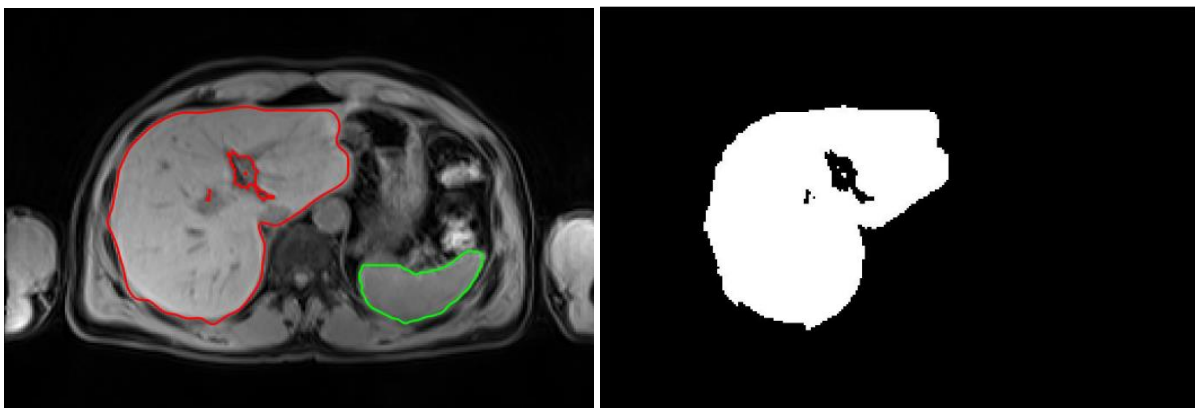
Hinweis 2: die Magnetresonanz-Bilder sind keine Originalbilder von Probanden, sondern wurden mit einem speziellen Entrauschungsverfahren geglättet. Somit wurde das Bildrauschen gegenüber dem Originalbild reduziert, was allerdings kleinere Bilddetails verwischen kann.

Die GUI soll folgende Funktionalitäten vorweisen und dabei möglichst fehlerfrei mit dem Nutzer interagieren:

- Auswahl von Schichtbilddatensätzen verschiedener Probanden, die jeweils in eigenen Ordnern abgespeichert sind
- Einlesen aller Schichtbilder eines Probanden: es existieren von jedem Probanden 64 Schichtbilder, die alle Grauwertbilder sind (keine Farbe!)
- Die Schichtbilder sollen nacheinander in einem Bereich der GUI angezeigt werden. Dabei soll der Benutzer die Möglichkeit haben, die Bilder sowohl in aufsteigender als auch in absteigender Reihenfolge der Schichtnummern durchzugehen.
- Die Probandennummer und die aktuelle Schichtnummer sollen in der GUI angezeigt werden.
- Der GUI-Nutzer soll eine Kontur auf dem angezeigten MR-Schichtbild eingeben können, damit er den Umriss eines Organs in der aktuell angezeigten Schicht manuell segmentieren kann.
- Die Kontur soll geschlossen sein und dabei das Organsegment in dieser Schicht mit einer Farbkontur darstellen.
- Das Organsegment kann Bereiche bzw. Löcher enthalten, die nicht zum Organ gehören.

- Der segmentierte Organbereich soll für jede Schicht in einem Binärbild abgespeichert werden. Das Binärbild entspricht einer binären Maske, die eine 1 an den Stellen des segmentierten Bereichs enthält und eine 0 für Bereiche, die nicht zum Organgewebe gehören (s.u.).
- Es sollte ein Binärbild für jede Schicht und für jedes Organ in speziellen Ordnern abgespeichert werden, der idealerweise als Name die entsprechende Probandennummer enthält.
- Die Konturen stellen somit die Grenzen der 1er-Bereiche zu den 0er-Bereichen in der binären Maske dar.
- Der Benutzer soll Möglichkeiten erhalten, Die Organgrenzen zu verändern bzw. zu verfeinern, indem Organbereiche hinzugefügt und entfernt werden können. Das Hinzufügen/Entfernen kann mit logischen Verknüpfungen auf den binären Masken erreicht werden:
 - o Beim Entfernen einer Region werden alle Pixel innerhalb der eingegebenen Region mit Nullen versehen und die Werte (0 oder 1) der anderen Bereiche in die Ergebnismaske übernommen.
 - o Beim Hinzufügen einer Region werden alle Pixel innerhalb der eingegebenen Region mit Einsen versehen und die Werte (0 oder 1) der anderen Bereiche in die Ergebnismaske übernommen.
- Der Anwender soll 4 verschiedene Organe eingeben können, deren Organbereiche mit unterschiedlichen Farben angezeigt werden sollen.
- Die 4 Organe sind: Leber, Milz, Galle, Nieren (linke und rechte)
- Die Daten von 4 Probanden liegen in vier unterschiedlichen Magnetresonanz-Gewichtungen vor. Das Rauschen in den Bilddaten wurde mit speziellen Bildverarbeitungsverfahren reduziert (entrauscht). Die Bilddaten der 4 Gewichtungen liegen ordnerweise vor (Gewichtung_42..45). Der Anwender soll sich die Bilder in den 4 verschiedenen Gewichtungen anzeigen lassen können, wobei er auch in jeder Schicht zwischen den 4 verschiedenen Gewichtungen beliebig wechseln können soll.

Links: Beispielbild für die farblich unterschiedlichen Konturen 2 verschiedener Organe (Leber: rot, Milz: grün). Rechts: Beispiel für eine binäre Maske (hier nur Lebergewebe)



Für diese Aufgabe ist das Aneignen von Programmierwissen (bzgl. Python) aus den folgenden Links hilfreich:

findcontours:

<https://www.delftstack.com/de/howto/python/opencv-find-contours/>

<https://www.programcreek.com/python/example/70440/cv2.findContours>

<https://medium.com/analytics-vidhya/opencv-findcontours-detailed-guide-692ee19eeb18>

ginput:

<https://www.geeksforgeeks.org/matplotlib-pyplot-ginput-in-python/>

Finden von Punkten innerhalb von Polygonen:

pip install Shapely

https://autogis-site.readthedocs.io/en/latest/notebooks/L3/02_point-in-polygon.html

<https://automating-gis-processes.github.io/CSC18/lessons/L4/point-in-polygon.html>

Aufgabe 2

Grafische Benutzeroberfläche zur Auswahl von Schiffsreisen

Es soll eine grafische Benutzeroberfläche zur Abfrage und Auswahl bestehender Schiffsreisen erstellt werden, die im Anschluss für eine personenbezogene Buchung weiter genutzt werden können. Die bestehenden Schiffsreisen sind in einer Excel-Datei (Schiffsreisen.xlsx) mit ihren wesentlichen Daten zusammengefasst. Die GUI soll die einzelnen Daten aus der Excel-Datei selektiv nach den Bedürfnissen des Anwenders abrufen können. Die GUI soll diejenigen Reisen herausuchen, die den Abfragekombinationen des Anwenders (= Auswahlkategorien) entsprechen: Z.B. alle Schiffsreisen auf der **Ostsee**, mit mindestens **8 Übernachtungen**, wobei die Städte „**Stockholm, Helsinki, ...**“ besucht werden und die einen bestimmten Schiffstyp (**A-F**) haben. Die Auswahl soll in einer Liste auf der GUI angezeigt werden. Dabei sollen die wesentlichen Informationen in der Liste dargestellt werden. Der Benutzer kann zu den einzelnen Kategorien Auswahlen treffen, allerdings auch keine Angabe/Bedingungen zu einzelnen Auswahlkategorien machen dürfen. Der Anwender soll folgende Abfragen einstellen können:

- Den Ort des Meeres („Meerart“: Ostsee, Nordsee, Mittelmeer)
- Die Anzahl an Übernachtungen +/-2 Übernachtungen. Wenn der Anwender 5 Übernachtungen auswählt, dann sollen Reisen mit 3-7 Übernachtungen angezeigt werden.
- Der Nutzer soll keine, eine oder mehrere Städte, die er als Schiffsreisender besuchen möchte, auswählen können. Bei Bedarf soll der Anwender ein Bild für eine Stadt, die er auswählt, separat angezeigt bekommen.
- Dabei ist zu beachten, dass belegte Auswahlkategorien die Auswahl der anderen Auswahlkategorien limitieren können. Z.B.: wenn ein spezielles Meer wie die Nordsee ausgewählt wurde, dann sollen keine Städte aus dem Mittelmeer mehr auswählbar sein.
- Der Schiffstyp (A-F) soll auswählbar sein. Dabei soll – bei Bedarf- der Schiffstyp als Bild angezeigt werden können.
- Für die meisten Auswahlkategorien sind Bilder hinterlegt, die auf Verlangen des GUI-Nutzers angezeigt werden können. Die Bilder sollen dabei möglichst in einheitlicher Größe angezeigt werden (Googlen nach geeigneten resize()-Methoden für Bilder).
- Bilder sollen dem Nutzer für folgende Auswahlkategorien angezeigt werden:
 - o Besuchte Städte: es können mehrere Bilder von den besuchten Städten ausgewählt und entsprechend angezeigt werden.
 - o Der Schiffstyp: es sind Beispielbilder für die Schiffstypen A-F in den Daten bereitgestellt.
 - o Die Kabinenart: es sind Bilder für die 3 Kabinenarten hinterlegt.

Die vom Nutzer vorausgewählten Reisen sollen ähnlich wie in der Excel-Datei in einer Art Liste erscheinen. Dabei soll die Reisennummer, Meerart, Anzahl Übernachtungen, etc....der Reihe nach angezeigt werden. Falls mehrere Reisen auf die Nutzerauswahl zutreffen, sollen mehrere Reisen untereinander erscheinen. Der Nutzer soll sich dann für eine Reise aus der Liste entscheiden können. Danach sollen die anderen Reisen aus der Liste verschwinden.

Wenn der Nutzer sich für eine Reise entschieden hat, soll er auch hier bei einem Klick auf eine der 3 angezeigten Preiskategorien (Innen-, Außen-, bzw. Balkonkabine) ein Bild von der Kabine angezeigt bekommen. Zudem soll der Nutzer eine Kabinenart für die ausgewiesene Reise mit einer Preiskategorie auswählen können. Nachdem die Kabinenart vom Nutzer festgelegt wurde, hat er sich für eine Reise entschieden. Dem Nutzer sollen dann alle besuchten Städte der Reise, der Schiffstyp und die Kabinenart gemeinsam mit dem zu zahlenden Endbetrag entweder in einem Bereich innerhalb der GUI oder in einem Extra-Fenster (Titel: „gekauft Produkt“) angezeigt werden. Dabei sollen die Bilder übersichtlich angeordnet werden und die Aufteilung sich nach der Anzahl der anzuzeigenden Bilder orientieren. Diese finale Anzeige dient als bildbasierte Darstellung des gekauften Produkts durch den Nutzer.

Beim GUI-Bereich bzw. im Extra-Fenster („gekauft Produkt“) soll mit einem entsprechenden GUI-Steuerelement einstellbar sein, ob die besuchten Städte auf einmal erscheinen oder nacheinander einzeln angezeigt werden sollen. Wenn die besuchten Städte auf einmal erscheinen, dann sollen sie entsprechend ihrer Anzahl neben- bzw. untereinander übersichtlich angezeigt werden. Falls die besuchten Städte nacheinander erscheinen sollen, sollen sie einzeln angezeigt werden und mit entsprechenden GUI-Steuerelementen (z.B. mit Vorwärts- und Rückwärts-Buttons) durchiteriert werden können.

Zum Schluss soll ihm ein Dialog angezeigt werden, der ihn nach seinem Namen (Vor- und Zuname) Adresse und Bankdaten fragt. Diese Daten sollen (quasi als Buchung für die ausgewählte Reise) in einer Textdatei abgespeichert werden.

Beispielauswahl besuchte Städte

Bergen



Göteborg



Trondheim



Tromsø



Beispielauswahl Schiffstyp (links) und Kabinenart (rechts)



Aufgabe 3

Grafische Benutzeroberfläche für einen Wochenbehandlungsplan für Zahnärzte und deren Patienten

Es soll eine GUI zur Zusammenstellung von zahnärztlichen Behandlungszeiten und -kosten für jeden Patienten konzipiert werden. Die Zusammenstellung soll für eine Kalenderwoche (Montag - Freitag) exemplarisch erstellt werden. Dabei soll der Patient Möglichkeiten erhalten, um u.a. die Anzahl seiner Behandlungen für diese (eine) Woche selbst festzulegen, um weitere Behandlungen z.B. auf spätere Wochen zu verlegen. Es muss kein genauer Terminkalender erstellt werden, allerdings müssen die Behandlungszeiten in das Zeitkontingent der 3 Zahnärzte passen, wobei jeder Zahnarzt 6 Stunden am Tag operiert.

Für diese grafische Benutzeroberfläche gibt es 2 verschiedene Rollen: Zahnarzt und Patient, wobei sich jeder unter seiner ID (initiales Passwort) anmelden kann und somit die Perspektive aus Zahnarztsicht bzw. Patientensicht auf die GUI automatisch eingestellt wird. Ein Patient erhält eine Übersicht über seine anstehenden Behandlungen und kann Anzahl der zu behandelnden Zähne und die Wertigkeit des Füllmaterials auswählen. Ein Zahnarzt kann sich die für jeden Wochentag eingetragenen Patienten, die er behandeln soll, anzeigen lassen. In der Datei Patienten_Zahnärzte_Kosten.xlsx sind die aktuellen Stammpatienten sowie die Zahnärzte der Zahnarztpraxis abgelegt. Zudem sind dort die Kosten und die Dauer von standardmäßigen Zahnbehandlungen hinterlegt.

Jeder Patient und jeder Zahnarzt haben eine eindeutige ID, die gleichzeitig auch ihre initialen Passwörter darstellen. Idealerweise werden diese Daten aus der Excel-Datei eingelesen,

können aber für die ersten Entwicklungsschritte per copy&paste in den Programmtext übernommen werden.

Rolle: Patient

Funktionalitäten/Programmabläufe als Patient: In der GUI soll es eine Zugangsmöglichkeit für jeden Patienten geben.

- Ein Patient meldet sich mit seiner ID, die gleichzeitig sein Passwort darstellt, am System an.
- Es soll eine Möglichkeit geben, dass der Patient ein neues Passwort eingibt und dieses im System hinterlegt wird. Der Benutzer kann, muss aber sein Passwort/ID nicht ändern.
- Der aktuelle Patient soll eine Kosten- und Zeitanzeige erhalten, in der sowohl die Kosten für seine Behandlungen als auch die Behandlungszeiten summiert werden. Dabei soll der Patient die Möglichkeit bekommen, die Anzahl der zu operierenden Zähne selbst einzustellen (um z.B. die restlichen Behandlungen in einem Folgemonat durchzuführen) und sich zwischen den 3 Möglichkeiten für ein Füllmaterial (normal, höherwertig, höchstwertig) zu entscheiden. Je nach eingestellter Anzahl der Zähne, dem Füllmaterial und der Krankenkasse soll dem Patienten ein Endbetrag angezeigt werden. Die anteilmäßige Kostenübernahme der gesetzlichen und privaten Krankenkassen sind in der Excel-Datei für jede Behandlungsart und Wertigkeit des Füllmaterials aufgeführt.
- Die Auswahlmöglichkeiten von Zahnanzahl und Füllmaterial sollen für den aktuell angemeldeten Patienten in der GUI sichtbar sein und seine Auswahl mit einem Bestätigungsbutton abgeschlossen werden können.
- Zum Beispiel: Frau Meyer wählt anstatt 2 Zähne (mit Karies) nur einen Zahn aus, den sie in dieser Woche behandeln lassen möchte. Sie wählt als gesetzlich Versicherte das höchstwertigste Füllmaterial (250€) für diesen Zahn aus. Somit bezahlt Sie $1 \times 250€ \times 0.5$ (Anteil als gesetzlich Versicherte) = 125€ und belegt $1 \times 0,25$ Stunden im Zeitkontingent von Dr. Kraft (der nur gesetzlich Versicherte in der Praxis behandelt). Nach einem Klick auf ein Bestätigungsbutton wird die Behandlung von Frau Meyer für diese Woche angenommen. Allerdings muss noch geprüft werden, ob diese Behandlung an einen Tag im Zeitkontingent von Dr. Kraft hineinpasst. Wenn das der Fall ist, wird Frau Meyer in das Zeitkontingent von Dr. Kraft an einem passenden Tag fest eingeordnet und die Anfrage ist abgeschlossen. Somit kann sich ein anderer Patient einloggen.
- Wenn ein Patient einem Zahnarzt zugeordnet wurde, wird das Zeitkontingent des Zahnarztes am ausgewählten Wochentag entsprechend erniedrigt. Es reicht mit Zeitkontingenten zu arbeiten anstatt die Zeitplanung der Zahnärzte detailliert anzuzeigen. Es kann jedoch auch ein detaillierter Wochenplan (mit 6 Stunden

Arbeitszeit pro Zahnarzt) angezeigt werden, bei dem sich ein Patient für eine Behandlungszeit eintragen kann. Dieser detaillierte Wochenplan der Zahnärzte ist allerdings optional.

- Jeder eingeloggte Patient soll eine Übersicht erhalten, an welchem Wochentag er für eine Behandlung eingeteilt wurde. Falls im vorigen Punkt mit einem detaillierten Wochenplan für Zahnärzte gearbeitet wird, soll der Patient auch die Uhrzeit am Wochentag seiner Behandlung angezeigt bekommen.
- Wenn die Behandlungen zeitlich länger werden, sollen die Behandlungen der restlichen Zähne auf andere Tage verteilt werden können.
- Falls die Behandlungen nicht mehr in das Zeitkontingent des zuständigen Zahnarztes hineinpassen, soll der Patient darauf hingewiesen werden, die Anzahl der zu operierenden Zähne zu reduzieren.
- Falls das Zeitkontingent eines Zahnarztes ausgefüllt, soll der Patient darauf hingewiesen werden, dass eine Behandlung seiner Zähne in der aktuellen Woche nicht möglich ist und seine angefragte Behandlung nicht in ein Zeitkontingent eines Zahnarztes übernommen werden.

Rolle: Zahnarzt

Wenn sich ein Zahnarzt in die GUI einloggt, dann:

- soll er ebenfalls die Möglichkeit haben, sein Passwort zu ändern.
- sich das von Patienten ausgefüllte Zeitkontingent von jedem Wochentag anzeigen zu lassen. Dabei sollen die Patientennamen mit Ihren Behandlungen und der gebuchten Zeit für den Zahnarzt in der GUI visualisiert werden.
- Optional kann auch mit einem detaillierten Wochenplan der einzelnen Zahnärzte gearbeitet werden.

Add-on: Anstatt nur die Zeitkontingente der Zahnärzte zu berücksichtigen soll ein konkreter Wochenplan für jeden Zahnarzt angezeigt werden. Die Behandlungszeiten der Zahnärzte sind in der Excel-Datei nun nachträglich hinzugefügt worden.

Aufgabe 4

Grafische Benutzeroberfläche für den Onlinekauf von landwirtschaftlichen Fahrzeugen und mobilen Nutzgeräten

Es sollen Käufe von Traktoren mit einer grafischen Benutzeroberfläche durchgeführt werden. Die zugrundeliegenden Informationen stehen in der Excel-Datei „mobile Arbeitsmaschinen Landwirtschaft.xlsx“. Ein Nutzer hat ein bestimmtes Budget zur Verfügung und soll anhand von Preisbereich, Km/h-Bereich, PS-Bereich und Baujahr-Bereich Traktoren auswählen können. Der Online-Verkäufer kann die Traktoren, deren Lagerbestand auf 0 geschrumpft ist nachbestellen oder Traktoren anderer Marken hinzukaufen. Nutzer und der online-Verkäufer sollen sich mit ihren Namen und Passwörtern in der GUI anmelden können, dazu sollen entsprechende Dialoge für die Passwordeingabe in der GUI realisiert werden. Vergeben Sie dazu initiale Passwörter für jeden Nutzer. Die Passwörter können der Einfachheit halber in einer Textdatei abgelegt werden.

Es sollen zunächst 5 Nutzer und der online-Verkäufer angelegt werden. Jeder Nutzer (jede Nutzerin) hat einen Namen und ein Startbudget:

Nutzer 1: Oskar, Budget: 750.000 €

Nutzer 2: Benni, Budget: 1.250.000 €

Nutzerin 3: Daniela, Budget: 2.250.000 €

Nutzer 4: Horst, Budget: 3.750.000 €

Nutzerin 5 Sieglinde, Budget: 5.000.000 €

Die Nutzer sollen sich unter ihrem Namen einloggen können und ihr Name und Budget sollen in der GUI angezeigt werden.

Zudem soll der Online-Verkäufer „Klaus“ sich mit seinem Namen und einem Passwort einloggen können. „Klaus“ hat ein Startbudget von 10.000.000 €.

(Add-on: Wenn noch Zeit ist, soll ein Dialog für die Aufnahme neuer Nutzer in die GUI integriert werden, bei dem Namen, Budget und initiales Passwort abgefragt werden.)

Ein Nutzer muss sich aus dem Online-Handel ausloggen (z.B. durch einen Ausloggen-Button), bevor ein anderer Nutzer oder der online-Verkäufer sich anmelden kann.

Ein Nutzer erhält die Perspektive des Käufers und kann seine Traktorensuche nach 4 verschiedenen Kriterien regeln:

- PS, Km/h, Preisbereich und Baujahr. Für diese Suchkriterien sollen geeignete Eingabemöglichkeiten mit geeigneten Auswahlbereichen (z.B. Preisbereich in 10.000€ Abstufungen, km/h-Bereich +/- 5, ...) zur Verfügung gestellt werden.
- Es soll eine Liste von ausgewählten Traktoren angezeigt werden, wobei der erste Eintrag ein kleines Bild des Traktors ist. Die anderen Kennzahlen (PS, km/h, Baujahr) sollen folgen sowie der Preis deutlich angezeigt werden.

- *(Add-on: Das angezeigte Bild soll bei Bedarf größer werden und der Nutzer soll in das Bild hineinzoomen können.)*
- Der eingeloggte Nutzer soll einen der Traktoren aus der Liste zum Kauf auswählen können. Der ausgewählte Traktor soll dann in einen Warenkorb abgelegt werden können.
- Abhängig von der Firma des ausgewählten Traktors soll der Nutzer sich kompatible Zusatzgeräte anzeigen lassen können. Aus der Liste der kompatiblen Zusatzgeräte soll der Nutzer sich wiederum einen oder mehrere Zusatzgeräte zum Kauf aussuchen können.
- Auch bei den Zusatzgeräten sollen kleine Bilder am Anfang der Liste angezeigt werden.
- Nachdem der Nutzer seine Auswahl getroffen hat und quasi „zum Kauf“ geht, soll der Endpreis aller Warenkorbprodukte (ausgewählte Traktoren und deren Zusatzprodukte) angezeigt werden.
- Überschreitet der errechnete Endpreis das Budget des Nutzers soll der Kauf nicht möglich werden und der Nutzer auf sein zu geringes Budget für den gewählten Kauf hingewiesen werden.
- Kann sich der Nutzer die Produkte im Warenkorb leisten, wird der Endpreis von seinem Budget abgezogen und dieses aktualisiert. Zudem werden die Anzahlen der eingekauften Produkte im Lager des Onlineanbieters entsprechend reduziert.

Perspektive des Online-Verkäufers

Der Online-Verkäufer kann, nachdem er sich eingeloggt hat, Traktoren und Zusatzgeräte, der schon im Bestand liegenden Marken sowie Traktoren anderer Marken (s. neue Modelle in Excel-Datei) einkaufen. Dabei muss er die Anzahl des Zukaufs bestimmen können und die eingekauften Produkte werden um ihre neue Anzahl im Lager aktualisiert. Auch der Käufer verfügt über ein Budget, das bei einem Verkauf an einen Nutzer erhöht bzw. einem Neukauf zur Lagerauffüllung reduziert wird. Gehen Sie der Einfachheit davon aus, dass der Verkäufer einen Einkaufspreis von 65% des Verkaufspreises hat.



Aufgabe 5

Grafische Benutzeroberfläche für ein „Schiffe versenken“-Spiel

Das bekannte Spiel „Schiffe versenken“ soll mit einer grafischen Benutzeroberfläche simuliert werden. Dabei wird das altbekannte Spiel aus:

https://de.wikipedia.org/wiki/Schiffe_versenken

leicht verändert und auf mehr als 2 Spieler ausgeweitet.

Vor Spielbeginn

- Vor Spielbeginn soll die Anzahl an Spielern ausgewählt werden können, jedoch soll die Spieleranzahl zunächst im Bereich 2-4 begrenzt werden.
- Die Spielfelder, auf denen die Schiffe der Spieler positioniert werden können, sollen für jeden Spieler auf der GUI angezeigt werden.
- Die Spielfelder sollen eine quadratische Größe von 10x10 haben und können als eine Tabelle oder sonstige Matrixform in der GUI erscheinen. (*Add-on: optional soll die Matrixgröße der Felder in sinnvollen Unter- und Obergrenzen auswählbar sein*)
- Die Felder sollen von 1-10 auf der vertikalen Achse sowie A-J auf der horizontalen Achse nummeriert und angesteuert werden (*bzw. entsprechend bei optionalen Variante mit variablen Matrixgrößen*). Die beiden Achsen sollen am linken bzw. oberen Spielfeldrand angezeigt werden.
- Farbauswahl für jeden Spieler: jeder Spieler soll sich seine Spielfarbe aus einem Menü oder Farbmenü auswählen können.
- Der Spieler soll die Positionen seiner Schiffe geeignet eingeben können. (Im echten Spielverlauf müssen dann die anderen Spieler wegsehen, da das Programm nur so die Schiffspositionen aller Spieler erhalten kann).
- Die Schiffe dürfen nur vertikal oder horizontal positioniert werden.
- Das GUI-Programm soll warnen und keine Schiffspositionen zulassen, die über den Spielfeldrand hinausgehen bzw. nicht vertikal oder horizontal ausgerichtet sind.
- Jeder Spieler hat sein eigenes Spielfeld, so dass sich Schiffspositionen von anderen Spielern an denselben Spielfeldkoordinaten nicht überlagern.
- Die Anzahl der Schiffe soll für alle Spieler gleich sein aber generell eingestellt werden können. Der Auswahlbereich soll zwischen 1 und 5 Schiffen liegen.
- Für jedes Schiff soll zudem die Schiffsgröße (minimal 1, maximal 5 Felder) eingestellt werden können. Z.B. angenommen alle Spieler verfügen über 2 Schiffe (Schiff A und B). Dann soll sowohl für Schiff A als auch für Schiff B deren Schiffsgröße über eine Nutzerinteraktion auswählbar sein. Die ausgewählte Schiffskonstellation (Schiffsanzahl und Anzahl belegter Felder) soll dann für jeden Spieler gleich sein.

Spielphase

In einer Runde kommen die Spieler nacheinander dran und sollen in jedem Spielzug 2 Aktionen ausführen können:

1. Sich einen Spieler auswählen, auf den sie einen Schuss abfeuern möchten. Bei 2 Spielern soll nur auf den direkten Gegenspieler abgefeuert werden können und die Abfrage zur Spielerauswahl (auf den ein Schuss abgefeuert werden soll) in der GUI schon automatisch nicht erscheinen (oder zumindest ausgeblendet sein).
2. Die Koordinaten für den Schuss sollen über eine geeignete Nutzerinteraktion ausgewählt werden können. Idealerweise soll das Koordinatensystem des ausgewählten Gegners in einem Extra-Dialog erscheinen und die Beschusskoordinaten mit der Maus auswählbar sein.

Nachdem die Beschusskoordinaten ausgewählt wurden, wird eine Rakete auf den ausgewählten Spieler an der eingestellten Koordinate abgefeuert. Erfolgt ein Treffer, dann soll das Feld entsprechend markiert werden (z.B. mit einem X). Erfolgte kein Treffer, dann wird dies ebenfalls in der Matrix kenntlich gemacht (z.B. mit einer 0).

Zudem sollen 4 Geräuscharten als Audiodaten in den Spielablauf eingebunden werden:

- Abschussraketenstart
- Wassereinschlag (kein Treffer)
- Teiltreffer
- Finaler Versenkungsschuss eines Schiffes

Nach jedem abgegebenen Schuss sollen einer der entsprechenden Geräusche ertönen. Vorschläge für verschiedenen Geräusche werden als mp3-Format auf der Moodle-Seite mitgeliefert. Es können auch eigene Geräusche recherchiert oder konstruiert werden.

Nachdem der letzte Spieler an der Reihe war, beginnt wieder Spieler Nr.1 mit seinen Aktionen. Die Bedingungen für das Spielende sollen vom Benutzer eingegeben werden können. Das Spiel kann grundsätzlich solange durchgeführt werden bis:

1. Nur noch ein Spieler Schiffe oder Schiffsteile besitzt. Dieser Spieler ist dann der Spielsieger.
2. Ein Spieler keine Schiffe mehr hat
3. Eine Anzahl an Spielrunden erreicht wurde.

Im Falle von 2. Und 3. muss die erreichte Punktzahl über den Sieger entscheiden:

- Ein Treffer auf ein Einerschiff ist 5 Punkte Wert
- Ein Treffer auf ein Zweierschiff ist 4 Punkte Wert
- Ein Treffer auf ein Dreierschiff ist 3 Punkte Wert
- Ein Treffer auf ein Viererschiff ist 2 Punkte Wert
- Ein Treffer auf ein Fünferschiff ist 1 Punkt Wert

Der Sieger soll mit seiner erreichten Trefferpunktzahl angezeigt werden.

Das Spiel Schiffe versenken existiert in verschiedenen Ausführungen auch im Internet. Einen Eindruck vom Ablauf bietet z.B.:

<https://www.spiele-kostenlos-online.de/brettspiele/brett-spiele/3d-schiffe-versenken/>

Beispielübersicht für eine Schiffe-versenken-Spiel mit 4 Spielern:

Spieler 1

	A	B	C	D	E	F	G	H	I	J
1										
2		0								
3		X	0		0			0		
4		X								
5		0								
6						0				
7										
8							0		0	
9										
10	0									

Spieler 2

	A	B	C	D	E	F	G	H	I	J
1										
2		0							0	
3						0				
4				0						
5										
6							0			
7		0								
8								0		X
9										X
10									0	

Spieler 3

	A	B	C	D	E	F	G	H	I	J
1										
2	0		0							
3			0							
4				0						
5						0				
6										
7										
8						0		0		
9										
10	0		0				0			

Spieler 4

	A	B	C	D	E	F	G	H	I	J
1		0								
2		X								
3	0	X	0							
4		X						0		
5		X								
6		X								
7										
8									0	
9										
10		0								

Aufgabe 6

Grafische Benutzeroberfläche zur Verwaltung von Medikamentenkäufen zwischen Patienten, Apotheken und Medikamentengroßhandel

Es soll eine graphische Benutzeroberfläche für verschiedene Akteure erstellt werden, die bei der Versorgung mit Medikamenten von Bedeutung sind. Diese Akteure sind: Patienten, Ärzte, Apotheken und Medikamentengroßhandel. Dabei soll die GUI ermöglichen, die verschiedenen Handlungen dieser Akteure in ihren Rollen anzuzeigen bzw. deren Daten zu verarbeiten.

In der Datei Apotheke_Patienten_Ärzte.xlsx sind die Daten für diese GUI-Programmieraufgabe enthalten. In der Tabelle Patienten sind mehrere Patienten enthalten, die mit ihrer ID, Art der Krankenversicherung, Symptomatik sowie einem individuellen Maximalbudget Medikamentenkauf aufgelistet sind.

Rolle: Patient

Ein Patient soll sich mit der GUI anmelden können und sein initiales Passwort (ID) bei Bedarf verändern können. Die Passwörter der einzelnen Akteure können in einer Textdatei abgespeichert werden. Wenn ein Name eines unbekannten Patienten eingegeben wird, soll die GUI mit einem entsprechenden Hinweis reagieren. Der/die Patient(in) kann sich für seine/ihre Beschwerden einen der 4 Ärzte aussuchen oder ohne Arztbesuch bei einer Apotheke einkaufen.

Wenn er/sie ohne Arztbesuch ein Medikament kauft, kann er/sie nur nicht verschreibungspflichtige Medikamente erwerben. Entscheidet sich ein Patient für einen Arzt, dann soll der dauerhaft an diesen Hausarzt gebunden sein und keinen anderen Hausarzt aufsuchen können(!) Bei der Arztbehandlung wird mit einem Zufallsgenerator unter Einbeziehung der Verschreibungswahrscheinlichkeit des Hausarztes entschieden, ob der Hausarzt dem Patienten ein Medikament im konkreten Fall verschreibt. Wenn der Patient für seine Symptome kein Rezept erhält, kann dieser nur versuchen, ein Medikament für seine Symptome ohne Verschreibungspflicht bei einer Apotheke zu kaufen.

Der Patient soll sich die Preise aller zur Verfügung stehenden Medikamente für seine Symptome anzeigen lassen können. Dabei soll er eine Übersicht erhalten, bei welcher Apotheke welcher Preis aktuell veranschlagt wird (falls diese Apotheke das Medikament auf Lager hat). Er kann sich dann für ein Produkt entscheiden: dabei wird die Anzahl im Lager der Apotheke entsprechend um eine Mengeneinheit erniedrigt. Zudem wird dem Patienten ein Geldbetrag abgezogen. Dieser Geldbetrag wird folgendermaßen ermittelt:

- Im Falle einer privaten Krankenversicherung werden 70% des Preises und bei einer gesetzlichen (oder freiwillig gesetzlichen) Krankenversicherung werden 25% des Preises ermäßigt.
- Wenn das Rezept nicht verschreibungspflichtig ist, werden zusätzlich noch 20% auf den bereits durch den Krankenkassensatz erniedrigten Preis erlassen, wenn dem Patienten vom Arzt eine Verschreibung gewährt wurde.

Wenn das Rezept verschreibungspflichtig ist, kann der Patient es generell nur kaufen, wenn er eine Verschreibung durch seinen Hausarzt erhalten hat. Die GUI soll darauf hinweisen, wenn das Budget des Patienten für den ausgewählten Medikamentenkauf nicht ausreicht.

Wenn 10 Käufe durch einen oder mehrere Patienten durchgeführt wurden, soll mit einem Zufallsgenerator eine Betrag zwischen 1 und 50 € ermittelt werden. Dieser Zufallsgenerator soll für jeden Patienten angewendet werden, um die Patientenbudgets zufällig zu erhöhen.

Rolle: Apotheke

Eine Apotheke (bzw. ein Apothekenbesitzer) soll sich ebenfalls mit ihrer ID einloggen können und diese ID bei Bedarf verändern. Eine Apotheke kann aus dem Großhandel neue Medikamente hinzukaufen, um den Lagerbestand zu erhöhen. Zudem soll eine Apotheke entscheiden können, ob sie neue Medikamente in einer bestimmten Anzahl hinzukauft. Eine Apotheke hat ebenfalls ein Budget. Wenn dieses Budget aufgebraucht ist, kann die Apotheke erst wieder neue Medikamente aus dem Großhandel kaufen, wenn sie durch neue Verkäufe an Patienten ihre Budget aufgefüllt hat. Die Apotheke kauft zu Großhandelspreisen beim Großhandel ein. Wenn im Großhandel das von der Apotheke bestellte Medikament nicht mehr zur Verfügung steht, dann kann es von der Apotheke nicht gekauft werden.

Zudem soll die GUI eine Funktionalität bereitstellen, so dass bei Bedarf der gesamte Bestand der Apotheke sowie das aktuelle Budget angezeigt wird.

Rolle: neue Apotheke

Es soll in der GUI die Möglichkeit geben auch neue Apotheken aufzunehmen. Dabei soll der Bestand an bereits vorhandenen Medikamenten über geeignete GUI-Steuerelemente abgefragt werden und somit die neue Apotheke mit seinem Medikamentenbestand ins GUI-System aufgenommen werden können. Dabei sollen auch Name der Apotheke und das aktuelle Budget eingegeben werden können, so dass die neue Apotheke auch wieder Medikamente beim Großhandel kaufen kann oder an einen Patienten verkaufen kann.

Aufgabe 7

Grafische Benutzeroberfläche zur Aktienanalyse für Kunden eines Aktienfonds

Sie sind ein Startup-Unternehmen im Finanzdienstleistungsbereich und möchten ihren potentiellen Kunden eine grafische Benutzeroberfläche zum Zusammenstellen für ein Aktienportfolio anbieten. Ein Teil Ihrer Kunden schwört auf den Index deutscher Mittelstandfirmen (MDAX) und möchte sich nur aus diesen Aktien ein Portfolio zusammenstellen können.

Die 5 in der Excel-Datei aufgelisteten Kunden sind Interessenten ausschließlich für den MDAX und sollen sich mit Passwort und ID in die Benutzeroberfläche einloggen können. Passwort und ID sollen auch vom Kunden bei Bedarf wieder geändert werden können.

In der Excel-Datei MDAX_60 sind 60 Unternehmen mit einigen generellen Informationen (wie z.B. Branche, Firmensitz) sowie finanziellen bzw. anlagerelevanten Informationen (wie z.B. KGV, ISIN, Dividendenrendite).

Die folgenden Teilaufgaben stellen die wesentlichen Programmierleistungen für diese Aufgabe dar:

Teilaufgabe 1: Erstellung einer Auswahlmöglichkeiten zur Anlageinformation des Kunden

Der eingeloggte Kunde soll mehrere Auswahlmöglichkeiten haben, um sich eine für ihn interessante Unterauswahl in der GUI (oder in einem extra Fenster) anzeigen zu lassen. Zu Beginn sollen die in der Excel-Datei enthaltenen Informationen zur Übersicht entsprechend in einer Gesamtanzeige aufgelistet werden.

Der Kunde / die Kundin soll sich dann beliebige Unterlisten einer Aktienausswahl nach den Spaltenbezeichnungen der Tabelle angeben lassen können. Dabei sollen eine Unterliste der den Auswahlprinzipien entsprechenden Firmen gemeinsam mit den restlichen Angaben aus den Spaltenbezeichnungen geeignet angezeigt werden. Auswählbar sollen folgende Eigenschaften (Spaltenbezeichnungen) sein:

- Die Branche: der/die Kunde/Kundin soll sich alle Unternehmen einer Branche anzeigen lassen. Die Branche soll sich idealerweise mit einem geeigneten GUI-Steuerelement zur Auswahl (z.B. eine Combobox) auswählen lassen.
- Für die Eigenschaften: Index-Gewichtung, Anzahl an Aktien, Marktkapitalisierung, KGV und Dividendenrendite sollen geeignete Einstellungen zur Auswahl realisiert sein (z.B. ein Schieberegler oder ein ähnliches Steuerelement), der diese Eigenschaften in geeigneten Zahlenschritten auswählbar gestaltet.
- Die restlichen Eigenschaften (Firmenname, Firmensitz und ISIN=internationale Wertpapierkennnummer) sollen nicht auswählbar sein. Sie sollen aber in den ausgewählten Unterlisten als zusätzliche Informationen erscheinen.
- Die Spalte „geheimer Trendwert“ soll nicht angezeigt werden können, da sie dem Kunden nicht zur Verfügung steht!

- In der 1.Spalte für die Firmennamen sind URL-Links angegeben. Der GUI-Benutzer soll sich bei einem Klick auf einen Firmennamen in der Unterliste weitere Informationen zum Unternehmen anzeigen lassen können (entweder in einem Webbrowser oder in der GUI selber).

Der Kunde soll sich aus der Gesamtanzeige oder den ausgewählten Unterlisten sein Aktienportfolio zusammenstellen können. Er soll sich dann entscheiden können, wie er sein Anlagekapital auf die einzelnen Portfolio-Positionen verteilt. Dabei muss er nicht sein gesamtes Anlagekapital ausnutzen müssen. Das Anlagekapital von 5 Beispielkunden ist ebenfalls in der Tabelle unten aufgeführt.

Teilaufgabe 2: Realisierung eines zufälligen Aktienverlaufs

Das ausgewählte Portfolio eines Kunden soll sich nach einer zufälligen Änderung der Aktienkurse über 12 Monate verändern. Dabei soll für jedes Monatsende eine Veränderung berechnet werden. Es soll eine Zufallszahl im Bereich -0.2 bis + 0.2 erzeugt werden (damit also zunächst eine monatliche zufällige Änderung von maximal 20% erzeugt werden können). Um den Kurs nicht rein zufällig entwickeln zu lassen, sollen zudem für jede Aktie die „geheimen Trendwerte“ einbezogen werden, die in der letzten Spalte der Excel-Tabelle stehen. Die monatliche Kurswertänderung einer Aktie (KW) soll nach folgender Formel berechnet werden:

Neuer KW für den Folgemonat = KW vom Vormonat + KW vom Vormonat * Zufallszahl + KW vom Vormonat * geheimer Trendwert

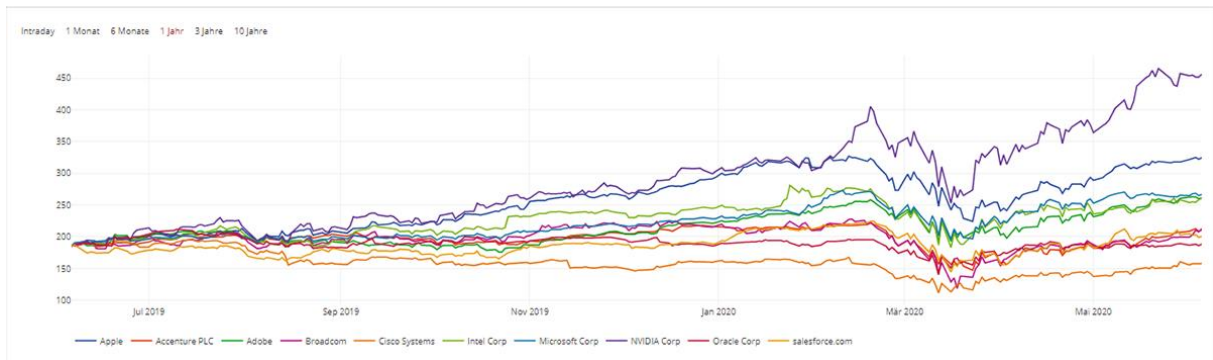
Die monatliche Kurswertänderung soll monatlich zum Aktienkurs hinzuaddiert (bzw. abgezogen) werden, um daraus den Aktienkurs nach einem Jahr zu ermitteln. Der Ausgangswert für den Aktienkurs entspricht dem in der Tabelle aufgelisteten Kurs (vom 07.10.22). Die monatliche Veränderung soll also 12 mal ermittelt werden so dass sich 12 monatliche Kurswerte ergeben.

Das Portfolio soll nach 12 Monaten ebenfalls angezeigt werden können, um einen Gesamtgewinn oder Verlust für jeden Kunden anzeigen zu können.

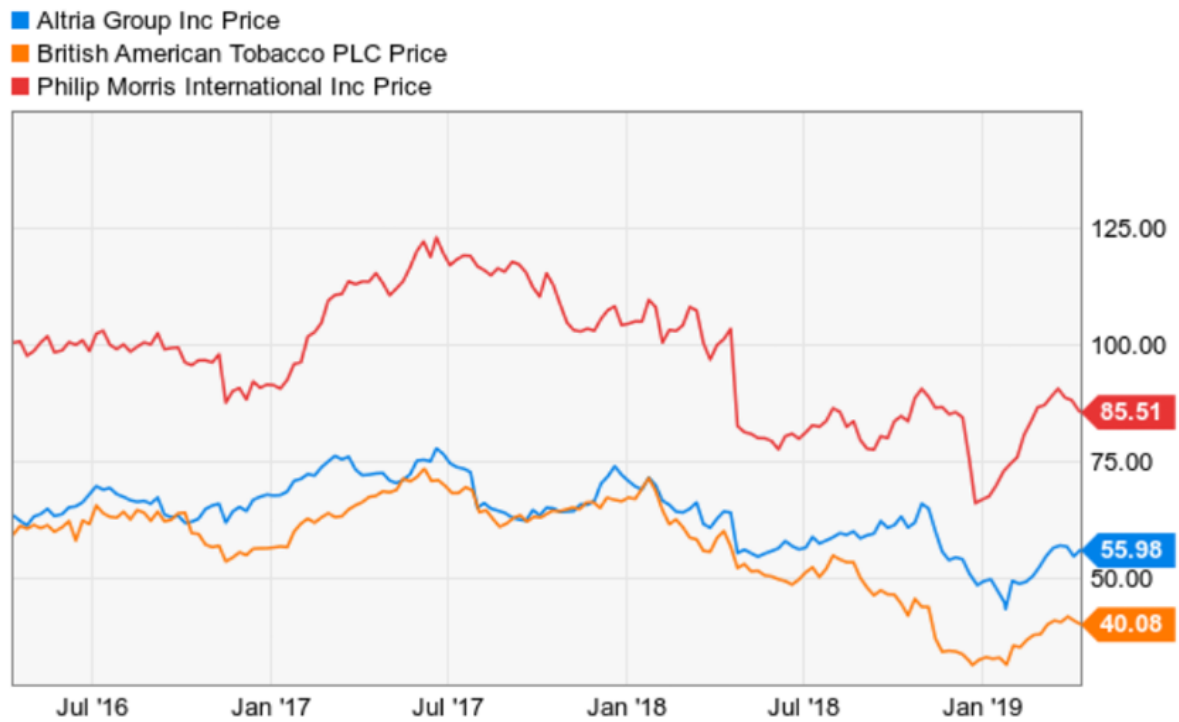
Teilaufgabe 3: Anzeige und Gegenüberstellung von Aktiencharts

Jeder Kunde soll sich die 12 monatigen Aktienverläufe sollen für jede Aktie seines Aktiendepots anzeigen lassen können. Diese Aktienverläufe bzw. Charts können z.B. mit der Bibliothek Matplotlib in einem Linienplot angezeigt werden. Dabei soll der Anleger die Möglichkeit erhalten, sich die Charts von mehreren Aktien seines Aktiendepots (z.B. in verschiedenen ausgewählten Farben) anzeigen zu lassen. Damit soll der Kunde die Aktienverläufe seiner Einzelpositionen visuell vergleichen können.

Aktiencharts-Vergleich Beispiel 1



Aktiencharts-Vergleich Beispiel 2



Apr 12 2019, 4:43AM EDT. Powered by **YCHARTS**

Generelle Hinweise für alle Praktikumsaufgaben

Um die Kriterien für eine gute Usability zu garantieren sollen optional auch undo-Operationen implementiert werden. Das Einbinden von Undo-Operationen bei den einzelnen GUIs ist ein Add-on und soll nur optional (bei Interesse und genügend Restzeit) behandelt werden. Geeignete Quellen finden sich für Python unter:

Einfache und/redo Möglichkeiten für Text Buttons in TkInter:

https://www.youtube.com/watch?v=yd2a_olJ4WM

Undo/redo mit dem Command Design Pattern in Python:

https://medium.com/@sean_bradley/command-design-pattern-in-python-2f15b09f3774

<https://medium.com/design-patterns-in-python/undo-redo-pattern-in-python-70ade29644b3>

https://www.youtube.com/watch?v=FM71_a3txTo&t=1036s

Generelle Hilfen zum Arbeiten mit Verzeichnissen und Dateien sowie Arbeiten mit Bildern in Python:

Bilder Laden und Speichern:

<https://towardsdatascience.com/loading-and-saving-images-in-python-ba5a1f5058fb>

<https://www.delftstack.com/de/howto/numpy/save-numpy-array-as-image/>

Öffnen von Dateien und Iterieren über Verzeichnisse:

<https://www.geeksforgeeks.org/how-to-iterate-over-files-in-directory-using-python/>

<https://www.codegrepper.com/code-examples/python/iterate+over+folders+python>

Das Speichern von Passwörtern kann für alle Praktikumsaufgaben in einfachen Textdateien geschehen, wo die entsprechenden Zugangsdaten abgelegt sind.

Mp3-Dateien in Python abspielen:

<https://www.delftstack.com/de/howto/python/python-play-mp3/>

Erzeugen von Tabellen mit Tkinter bzw. Zeichnen mit Python-Canvas:

<https://pythonguides.com/python-tkinter-table-tutorial/>

<https://www.geeksforgeeks.org/create-table-using-tkinter/>

<https://anzelg.github.io/rin2/book2/2405/docs/tkinter/canvas.html>