Übung 05a: Einführung in die GUI Programmierung mit PyQt5

Ziel dieser Übung ist das Erstellen eines Dialoges zur Steuerung der Simulation des verschieblichen Pendels (Bild: siehe letzte Übung). Dabei sollen die Parameter des Modells (Masse, Seillänge) in der graphischen Oberfläche anpassbar sein, ebenso wie die Simulationsdauer und die Simulationsschrittweite. Auf Knopfdruck soll die Simulation ausgeführt werden.
Eine häufig vorkommende Aufgabe ist das Speichern von Programmeinstellungen oder Parametern in Konfigurationsdateien (ini-Files). Dazu existiert in der Python Standardbibliothek das Modul `configparser`, mit dessen Hilfe die Daten aus der Eingabemaske geladen und gespeichert werden sollen.

Übungsaufgabe 1:
Graphische Oberfläche für die Simulationssteuerung (`simgui.py`)

1. Erstellen Sie Labels und Eingabefelder für die Parameter `l` (Pendellänge), `m1` und `m2` (Massen für Wagen und Pendel). Belegen Sie die Felder mit sinnvollen Werten als Voreinstellung.
2. Wiederholen Sie 1. für die Simulationsschrittweite und die Simulationsdauer.
3. Legen Sie zwei Buttons für das Simulieren und das Beenden des Programms an.
4. Fügen Sie alle Widgets in ein (7x2)-Gridlayout ein (siehe: `example-code/gui-example4.py` bzw. Doku) und ordnen Sie die Buttons rechtsbündig an. Probieren Sie die GUI-Anwendung jetzt aus.
5. Geben Sie jedem `LineEdit` einen Validator für Zahlenwerte.
6. Stellen sie die Textausrichtung in den LineEdits auf rechtsbündig.
7. „connecten“ Sie den Simulationsbutton mit der `simulate`-Funktion; alle folgende Aufgaben sind innerhalb der `simulate`-Funktion zu bearbeiten
8. Holen Sie alle Werte aus den `LineEdits` und wandeln Sie diese in Fließkommazahlen (`float(...)`) um.
9. Legen Sie mit `arange` (aus scipy) eine Zeitachse an (Parameter aus GUI verwenden!).
10. Simulieren Sie das System mit odeint:
    `res = odeint( rhs, y0, t, args=(m1, m2, l) )`
    Die Startwerte `y0` (entspricht $[x(0), \varphi(0), \dot{x}(0)\dot{\varphi}(0)]$) können frei gewählt werden.
11. Erzeugen Sie zwei Subplots auf `<fig>` und stellen Sie die Ergebnisse jeweils für Laufkatze und Last in einem Subplot grafisch dar. Achten Sie auf Achsenbeschriftung, Legende, Hilfslinien etc.

Übungsaufgabe 2: Speichern und Laden der Konfiguration

1. Legen Sie zwei weitere Buttons zum Laden und Speichern an.
2. Verbinden Sie diese mit den Funktionen `openFile` und `saveFile`.
3. Komplettieren Sie beide Funktionen (mit saveFile beginnen!) mit Aufrufen der entsprechenden Dialoge für Dateinamen und schreiben/lesen Sie die Daten mit dem `configparser` Modul:
   Schreiben:

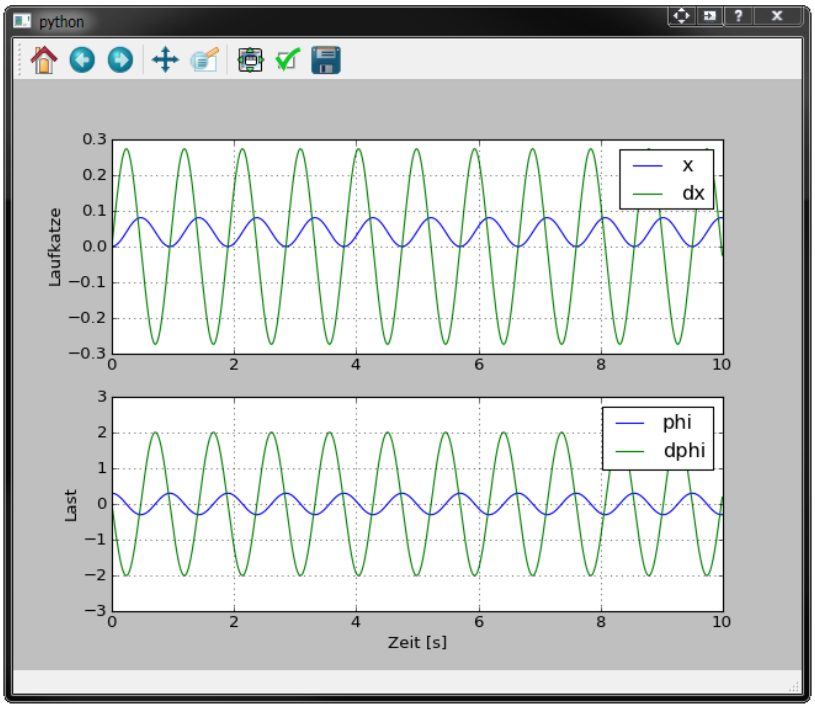
```
c = configparser.SafeConfigParser()
c.add_section("Parameter")o
c.set("Parameter", "m2", str(mass1Edit.text()))
```

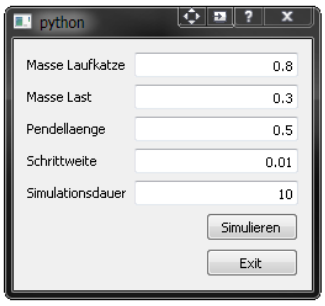
   Lesen:

```
mass1Edit.setText(c.get("Parameter", "m1"))
```

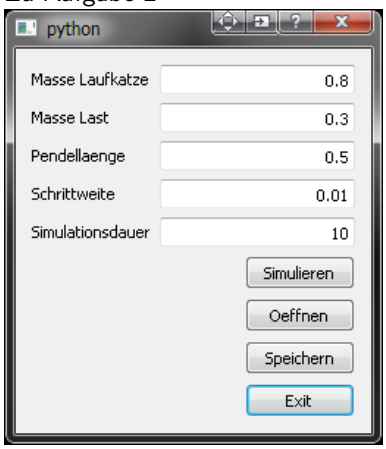
o Ergebnis und Beispielkonfigurationsdatei siehe Abb. 2

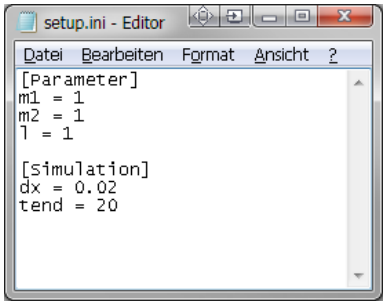
Zu Aufgabe 1





Zu Aufgabe 2





Zusatzaufgabe

- Erweitern Sie das Programm um die Eingabe der Startwerte der Simulation.