Diff:

Differences between given skeleton and solution
In order to make the sample solution easier to understand, the differences between it and the given skeleton source code were highlighted with the help of the program `diff`.

Legend:

- Gray: unchanged text (only excerpts).
- Green: new lines
- Yellow: changed lines
- Red: deleted lines

Note: Files not listed have not been changed.

This document was created with the help of diff2html erstellt.

```
Nur in ../course11-gui-part1/exercise/solution/: parametervalues.ini.
Gemeinsame Unterverzeichnisse: ../course11-gui-part1/exercise/code/__pycache__ und ../course11-gui-part1/exercise/solution/__pycache__.
diff -u ../course11-gui-part1/exercise/code/simgui.py ../course11-gui-part1/exercise/solution/simgui.py
```

../course11-gui-part1/exercise/code/simgui.py		../course11-gui-part1/exercise/solution/simgui.py	
⋮		⋮	
1			
2	"""	1	"""
3	This is the main program.	2	This is the main program.
4	"""	3	"""
⋮		⋮	
20		19	
21	import matplotlib.pyplot as plt	20	import matplotlib.pyplot as plt
22		21	
		22	import configparser
		23	
23		24	
24	# QApplication Instanz wird immer benötigt (sys.argv erst mal hinnehmen)	25	# QApplication Instanz wird immer benötigt (sys.argv erst mal hinnehmen)
25	app = QtWidgets.QApplication(sys.argv)	26	app = QtWidgets.QApplication(sys.argv)
⋮		⋮	
37	mass1_label = QtWidgets.QLabel('Masse Laufkatze', dialog)	38	mass1_label = QtWidgets.QLabel('Masse Laufkatze', dialog)
38	mass1_edit = QtWidgets.QLineEdit('0.8', dialog)	39	mass1_edit = QtWidgets.QLineEdit('0.8', dialog)
39		40	
40	# ...	41	mass2_label = QtWidgets.QLabel('Masse Last', dialog).
		42	mass2_edit = QtWidgets.QLineEdit('0.3', dialog)
		43	
		44	len_label = QtWidgets.QLabel('Pendellänge', dialog)
		45	len_edit = QtWidgets.QLineEdit('0.5', dialog)
		46	
		47	dx_label = QtWidgets.QLabel('Schrittweite', dialog)
		48	dx_edit = QtWidgets.QLineEdit('0.01', dialog)
41		49	
		50	t_end_label = QtWidgets.QLabel('Simulationsdauer', dialog)
		51	t_end_edit = QtWidgets.QLineEdit('10', dialog)
42		52	
43		53	
44	# Buttons	54	# Buttons
45	# exit_button = ...	55	sim_button = QtWidgets.QPushButton('Simulieren', dialog).
		56	open_button = QtWidgets.QPushButton('Öffnen', dialog)
		57	save_button = QtWidgets.QPushButton('Speichern', dialog)
		58	exit_button = QtWidgets.QPushButton('Exit', dialog)
46		59	
47		60	
48	# Ausrichtung anpassen	61	# Ausrichtung anpassen
49	#	62	mass1_edit.setAlignment(QtCore.Qt.AlignRight).
		63	mass2_edit.setAlignment(QtCore.Qt.AlignRight)
		64	len_edit.setAlignment(QtCore.Qt.AlignRight)
		65	dx_edit.setAlignment(QtCore.Qt.AlignRight)
		66	t_end_edit.setAlignment(QtCore.Qt.AlignRight)
		67	
50		68	
51	# zulässige Zeichen beschränken - hier nur Zahlen eingeben	69	# zulässige Zeichen beschränken - hier nur Zahlen eingeben
52	#	70	mass1_edit.setValidator(QtGui.QDoubleValidator(mass1_edit)).
		71	mass2_edit.setValidator(QtGui.QDoubleValidator(mass2_edit))
		72	len_edit.setValidator(QtGui.QDoubleValidator(len_edit))
		73	dx_edit.setValidator(QtGui.QDoubleValidator(dx_edit))
		74	t_end_edit.setValidator(QtGui.QDoubleValidator(t_end_edit))
53		75	
54		76	

```
# Layout
layout = QtWidgets.QGridLayout()
.
#layout.addWidget(...).
```

```
# Layout
layout = QtWidgets.QGridLayout()
layout.addWidget(mass1_label, 0, 0).
layout.addWidget(mass1_edit, 0, 1).
layout.addWidget(mass2_label, 1, 0)
layout.addWidget(mass2_edit, 1, 1)
layout.addWidget(len_label, 2, 0)
layout.addWidget(len_edit, 2, 1)
layout.addWidget(dx_label, 3, 0)
layout.addWidget(dx_edit, 3, 1)
layout.addWidget(t_end_label, 4, 0)
layout.addWidget(t_end_edit, 4, 1)

layout.addWidget(sim_button, 5, 1, 1, 1, QtCore.Qt.AlignRight)
layout.addWidget(open_button, 6, 1, 1, 1, QtCore.Qt.AlignRight)
layout.addWidget(save_button, 7, 1, 1, 1, QtCore.Qt.AlignRight)
layout.addWidget(exit_button, 8, 1, 1, 1, QtCore.Qt.AlignRight)

dialog.setLayout(layout)


# Focus auf Exit
#exit_button.setFocus().
.
.
```

```
# Focus auf Exit
exit_button.setFocus().
```

```
def saveFile():
⋮
    """

    # Dialog für Dateiname (gibt ein 2-Tupel zurück), siehe Folie 9
    #filename, type_filter = ...
    .

    if filename == "":
        return  # wenn "Abbrechen"/"Cancel" gedrückt wurde -> nichts tun

    # Configparser anlegen und Daten übergeben
    # (Modul configparser muss importiert sein, siehe oben)
    c = configparser.SafeConfigParser()
    c.set("XXX", "XXX", mass1_edit.text())
    c.XXX

    # Configfile schreiben
    with open(filename, 'w') as fid:
⋮
    """

    # Dialog für Dateiname (gibt ein 2-Tupel zurück), siehe Folie 9
```

```
def saveFile():
⋮
    """

    # Dialog für Dateiname (gibt ein 2-Tupel zurück), siehe Folie 9
    filename, type_filter = QtWidgets.QFileDialog.getSaveFileName().

    if filename == "":
        return  # wenn "Abbrechen"/"Cancel" gedrückt wurde -> nichts tun

    # Configparser anlegen und Daten übergeben
    c = configparser.SafeConfigParser()

    c.add_section('Parameter')
    c.set('Parameter', 'm1', str(mass1_edit.text()))
    c.set('Parameter', 'm2', str(mass2_edit.text()))
    c.set('Parameter', 'l',  str(len_edit.text()))

    c.add_section('Simulation')
    c.set('Simulation', 'dx', str(dx_edit.text()))
    c.set('Simulation', 't_end', str(t_end_edit.text()))

    # Configfile schreiben
    with open(filename, 'w') as fid:
⋮
    """

    # Dialog für Dateiname (gibt ein 2-Tupel zurück), siehe Folie 9
```

```
99     #filename, type_filter= ...
100
101    if filename == "":
102        return  # wenn "Abbrechen"/"Cancel" gedrückt wurde -> nichts tun
103
104    # Configparser anlegen
105    c = configparser.SafeConfigParser()
106    c.read(str(filename))
107
108    # Werte den LineEdits zuordnen.
109    #mass1_edit.setText(...).
110
111
112
113
114 def simulate():
⋮
116    Diese Funktion liest die Parameter aus allen LineEdits, konvertiert sie in
117    floats und führt damit die Simulation aus. Anschließend werden die
118    Ergebnisse mit matplotlib dargestellt. Die Startwerte der Simulation sind
119    hier noch statisch vorgegeben (Hart kodiert)..
120    """
121
122    # Werte holen
123    m1 = float(mass1_edit.text())
124    # ....
125
126    # Zeitachse anlegen (np.linspace).
127    # t = ....
128
129    # Simulation ausführen, siehe Aufgabenstellung.
130    # res ....
131
132    # Ergebnisse plotten
133    fig = plt.figure()
134
135    # Hier muss etwas getrickst werden: wir legen einen neuen Dialog an, auf den
136    # matplotlib zeichnet. Der plot_dialog hat unseren Hauptdialog als parent und.
137    # ist "nicht modal". Damit können wir beliebig viele Ergebnisfenster parallel.
138    # darstellen und Simulationsergebnisse vergleichen..
139    # Mehr Infos: https://en.wikipedia.org/wiki/Modal_window.
```

```
137    filename, type_filter = QtWidgets.QFileDialog.getOpenFileName().
138
139    if filename == "":
140        return  # wenn "Abbrechen"/"Cancel" gedrückt wurde -> nichts tun
141
142    # Configparser anlegen
143    c = configparser.SafeConfigParser()
144
145    # aus Datei lesen.
146    print('lade', filename).
147
148    if c.read(str(filename)):
149        print('OK')
150    else:
151        print('Keine Konfigurationsdatei geladen')
152
153    # Werte den LineEdits zuordnen
154    mass1_edit.setText(c.get('Parameter', 'm1'))
155    mass2_edit.setText(c.get('Parameter', 'm2'))
156    len_edit.setText(c.get('Parameter', 'l'))
157
158    dx_edit.setText(c.get('Simulation', 'dx'))
159    t_end_edit.setText(c.get('Simulation', 't_end'))
160
161
162 def simulate():
⋮
164    Diese Funktion liest die Parameter aus allen LineEdits, konvertiert sie in
165    floats und führt damit die Simulation aus. Anschließend werden die
166    Ergebnisse mit matplotlib dargestellt. Die Startwerte der Simulation sind
167    hier noch statisch vorgegeben..
168    """
169
170    # Werte holen
171    m1 = float(mass1_edit.text())
172    m2 = float(mass2_edit.text()).
173    l = float(len_edit.text())
174    dx = float(dx_edit.text())
175    t_end = float(t_end_edit.text())
176
177    # oder auch:.
178    #m = mass1_edit.text().toDouble()[0]   # gibt Tupel ala (wert, OK) zurück.
179
180    # Zeitachse anlegen.
181    t = arange(0, t_end, dx).
182
183    # Simulation ausführen
184    res = odeint(rhs, [0, 0.3, 0, 0], t, args=(m1, m2, l))
185
186    # Ergebnisse plotten
187    fig = plt.figure()
188
189    # Hier muss etwas getrickst werden: wir legen einen neuen Dialog an, auf den
190    # matplotlib zeichnet. Der plotDialog hat unseren Hauptdialog als parent und.
191    # ist modeless. Damit können wir beliebig viele Ergebnisfenster parallel.
192    # darstellen..
193    plotDialog = QtWidgets.QDialog(dialog).
```

```
140     plot_dialog = QtWidgets.QDialog(dialog).
141     fig.canvas.parent().setParent(plot_dialog).
142
143     # Ergebnisse für Laufkatze
144     ax1 = fig.add_subplot(2, 1, 1)
145     # ....
146
147
148     # Ergebnisse für Last
149     # ax2 = .....
150
151     # Hier wird jetzt der Dialog angezeigt und nicht mehr die show-Funktion von
152     # matplotlib aufgerufen!
153     plot_dialog.show().
154
155
156
157 # Buttons verknüpfen
158 #exit_button.clicked.connect(dialog.close).
159 .
160 .
161
162
163 #--------------------------------------------------------------------------------
```

```
194     fig.canvas.parent().setParent(plotDialog).
195
196     # Ergebnisse für Laufkatze
197     ax1 = fig.add_subplot(2, 1, 1)
198     ax1.plot(t, res[:, 0], label='x').
199     ax1.plot(t, res[:, 2], label='dx')
200
201     ax1.grid(True)
202     ax1.legend()
203     ax1.set_ylabel('Laufkatze')
204
205     # Ergebnisse für Last
206     ax2 = fig.add_subplot(2, 1, 2).
207     ax2.plot(t, res[:, 1], label='phi')
208     ax2.plot(t, res[:, 3], label='dphi')
209
210     ax2.grid(True)
211     ax2.legend()
212     ax2.set_xlabel('Zeit [s]')
213     ax2.set_ylabel('Last')
214
215     # Hier wird jetzt der Dialog angezeigt und nicht mehr die show-Funktion von
216     # matplotlib aufgerufen!
217     plotDialog.show().
218
219
220
221 # Buttons verknüpfen
222 sim_button.clicked.connect(simulate).
223 open_button.clicked.connect(openFile).
224 save_button.clicked.connect(saveFile).
225 exit_button.clicked.connect(dialog.close)
226
227
228 #--------------------------------------------------------------------------------
```