

Diff: Übung 06a

Unterschiede zwischen Vorgabe und Lösung

Um die Musterlösung leichter nachvollziehbar zu gestalten wurden die Unterschiede zur Quelltext-Vorgabe mit Hilfe des Programms `diff` grafisch aufbereitet.

Legende:

- Grau: unveränderter Text (nur auszugsweise)
- Grün: neue Zeilen
- Gelb: veränderte Zeilen
- Rot: gelöschte Zeilen (kommt hier nicht vor)

Hinweis: nicht aufgeführte Dateien wurden nicht verändert.

Dieses Dokument wurde mit Hilfe von [diff2html](#) erstellt.

```
diff -u ../kurs06a_GUI1/uebung/code/interaktiv.py ../kurs06a_GUI1/uebung/loesung/interaktiv.py
```

../kurs06a_GUI1/uebung/code/interaktiv.py	../kurs06a_GUI1/uebung/loesung/interaktiv.py
:	:
11 # QApplication Instanz wird immer benoetigt (sys.argv erst mal hinnehmen)	11 # QApplication Instanz wird immer benoetigt (sys.argv erst mal hinnehmen)
12 app = QtGui.QApplication(sys.argv)	12 app = QtGui.QApplication(sys.argv)
13	13
	14
	15
	16
	17 # Dialog erzeugen
14 dialog = QtGui.QDialog()	18 dialog = QtGui.QDialog()
15	19
16	20
	21 #----- einfachstes Beispiel -----
	22
	23 '''
	24 # Widgets erzeugen
	25 label = QtGui.QLabel('TextLabel', dialog)
	26 button = QtGui.QPushButton('Button', dialog)
	27 edit = QtGui.QLineEdit(dialog)
	28
	29
	30 # Layout erzeugen und Widgets in Layout packen (vertical box)
	31 layout = QtGui.QVBoxLayout()
	32 layout.addWidget(label)
	33 layout.addWidget(button)
	34 layout.addWidget(edit)
	35
	36 dialog.setLayout(layout)
	37 '''
	38
	39
	40 #----- jetzt mit GridLayout und Filebuttons -----
	41
	42
	43 def openFile():
	44 print QtGui.QFileDialog.getOpenFileName()
	45
	46 def saveFile():
	47 print QtGui.QFileDialog.getSaveFileName()
	48
	49
	50 # Widgets erzeugen
	51 massLabel = QtGui.QLabel('Masse', dialog)
	52 lenLabel = QtGui.QLabel('Pendellaenge', dialog)
	53
	54 massEdit = QtGui.QLineEdit('1', dialog)
	55 lenEdit = QtGui.QLineEdit('1', dialog)
	56

17	56
	57 openButton = QtGui.QPushButton('Oeffnen', dialog)
	58 saveButton = QtGui.QPushButton('Speichern', dialog)
	59 exitButton = QtGui.QPushButton('Exit', dialog)
18	60
19	61
	62 # Ausrichtung anpassen
	63 massEdit.setAlignment(QtCore.Qt.AlignRight)
	64
	65 # zulaessige Zeichen beschraenken - hier nur Zahlen eingeben
	66 massEdit.setValidator(QtGui.QDoubleValidator(massEdit))
	67
	68
	69 # Layout
	70 layout = QtGui.QGridLayout()
	71 layout.addWidget(massLabel, 0,0)
	72 layout.addWidget(massEdit, 0,1)
	73 layout.addWidget(lenLabel, 1,0)
	74 layout.addWidget(lenEdit, 1,1)
	75
	76 layout.addWidget(openButton, 2,1, 1,1, QtCore.Qt.AlignRight)
	77 layout.addWidget(saveButton, 3,1, 1,1, QtCore.Qt.AlignRight)
	78 layout.addWidget(exitButton, 4,1, 1,1, QtCore.Qt.AlignRight)
	79
	80 dialog.setLayout(layout)
	81
	82
	83 # Buttons verknuepfen
	84 openButton.clicked.connect(openFile)
	85 saveButton.clicked.connect(saveFile)
	86 exitButton.clicked.connect(dialog.close)
	87
	88
	89 # Focus auf Exit
	90 exitButton.setFocus()
	91
	92
	93
	94 #-----
	95
	96 # Dialog anzeigen
20 dialog.exec_()	97 dialog.exec_()
	98
	99
	100 # Inhalt des Eingabefeldes ausgeben
	101 print massEdit.text()

\ No newline at end of file

Only in ../kurs06a_GUI1/uebung/loesung/: model.pyc

diff -u ../kurs06a_GUI1/uebung/code/simgui.py ../kurs06a_GUI1/uebung/loesung/simgui.py

../kurs06a_GUI1/uebung/code/simgui.py	../kurs06a_GUI1/uebung/loesung/simgui.py
:	:
11 from scipy.integrate import odeint	11 from scipy.integrate import odeint
12 from scipy import arange	12 from scipy import arange
13	13
	14 import matplotlib
	15 matplotlib.use('Qt4Agg') # Backend explizit festlegen
	16
14 import matplotlib.pyplot as plt	17 import matplotlib.pyplot as plt
15	18
	19 import ConfigParser
16	20
17	21
18 # QApplication Instanz wird immer benoetigt (sys.argv erst mal hinnehmen)	22 # QApplication Instanz wird immer benoetigt (sys.argv erst mal hinnehmen)
:	:
31 mass1Label = QtGui.QLabel('Masse Laufkatze', dialog)	35 mass1Label = QtGui.QLabel('Masse Laufkatze', dialog)
32 mass1Edit = QtGui.QLineEdit('0.8', dialog)	36 mass1Edit = QtGui.QLineEdit('0.8', dialog)
33	37
34 # ...	38 mass2Label = QtGui.QLabel('Masse Last', dialog).
	39 mass2Edit = QtGui.QLineEdit('0.3', dialog)
	40
	41 lenLabel = QtGui.QLabel('Pendellaenge', dialog)
	42 lenEdit = QtGui.QLineEdit('0.5', dialog)
	43
	44 dxLabel = QtGui.QLabel('Schrittweite', dialog)
	45 dxEdit = QtGui.QLineEdit('0.01', dialog)
35	46
	47 tEndLabel = QtGui.QLabel('Simulationsdauer', dialog)
	48 tEndEdit = QtGui.QLineEdit('10', dialog)
36	49
37	50
38 # Buttons	51 # Buttons
39 # exitButton = ...	52 simButton = QtGui.QPushButton('Simulieren', dialog).
	53 openButton = QtGui.QPushButton('Oeffnen', dialog)
	54 saveButton = QtGui.QPushButton('Speichern', dialog)
	55 exitButton = QtGui.QPushButton('Exit', dialog)
40	56
41	57
42 # Ausrichtung anpassen	58 # Ausrichtung anpassen
43 #	59 mass1Edit.setAlignment(QtGui.Core.Qt.AlignRight).
	60 mass2Edit.setAlignment(QtGui.Core.Qt.AlignRight)
	61 lenEdit.setAlignment(QtGui.Core.Qt.AlignRight)
	62 dxEdit.setAlignment(QtGui.Core.Qt.AlignRight)
	63 tEndEdit.setAlignment(QtGui.Core.Qt.AlignRight)
	64
44	65
45 # zulaessige Zeichen beschraenken - hier nur Zahlen eingeben	66 # zulaessige Zeichen beschraenken - hier nur Zahlen eingeben
46 # ...	67 mass1Edit.setValidator(QtGui.QDoubleValidator(mass1Edit)).
	68 mass2Edit.setValidator(QtGui.QDoubleValidator(mass2Edit))

```

47
48
49 # Layout
50 layout = QtGui.QGridLayout()
51 .
52 #layout.addWidget(...).

53
54 dialog.setLayout(layout)
55
56 :
67 ""
68
69 # Dialog zum Oeffnen eine Datei
70 #filename = ...
71
72 # Configparser anlegen
73 c = ConfigParser.SafeConfigParser()
74 c.read(str(filename)).

75
76 # Werte den LineEdits zuordnen
77 #mass1Edit.setText(...).

69 lenEdit.setValidator(QtGui.QDoubleValidator(lenEdit))
70 dxEdit.setValidator(QtGui.QDoubleValidator(dxEdit))
71 tEndEdit.setValidator(QtGui.QDoubleValidator(tEndEdit))
72
73
74 # Layout
75 layout = QtGui.QGridLayout()
76 layout.addWidget(mass1Label, 0,0).
77 layout.addWidget(mass1Edit, 0,1).
78 layout.addWidget(mass2Label, 1,0)
79 layout.addWidget(mass2Edit, 1,1)
80 layout.addWidget(lenLabel, 2,0)
81 layout.addWidget(lenEdit, 2,1)
82 layout.addWidget(dxLabel, 3,0)
83 layout.addWidget(dxEdit, 3,1)
84 layout.addWidget(tEndLabel, 4,0)
85 layout.addWidget(tEndEdit, 4,1)
86
87 layout.addWidget(simButton, 5,1, 1,1, QtCore.Qt.AlignRight)
88 layout.addWidget(openButton, 6,1, 1,1, QtCore.Qt.AlignRight)
89 layout.addWidget(saveButton, 7,1, 1,1, QtCore.Qt.AlignRight)
90 layout.addWidget(exitButton, 8,1, 1,1, QtCore.Qt.AlignRight)
91
92 dialog.setLayout(layout)
93
94 :
105 ""
106
107 # Dialog zum Oeffnen eine Datei
108 filename = QtGui.QFileDialog.getOpenFileName().
109
110 # Configparser anlegen
111 c = ConfigParser.SafeConfigParser()
112 .
113 # aus Datei lesen
114 print 'lade', filename
115
116 if c.read(str(filename)):
117     print 'OK'
118 else:
119     print 'Keine Konfigurationsdatei geladen'
120
121 # Werte den LineEdits zuordnen
122 mass1Edit.setText(c.get('Parameter', 'm1')).
123 mass2Edit.setText(c.get('Parameter', 'm2'))
124 lenEdit.setText(c.get('Parameter', 'l'))
125
126 dxEdit.setText(c.get('Simulation', 'dx'))
127 tEndEdit.setText(c.get('Simulation', 'tEnd'))

```

78		128	
79		129	
80		130	
:		:	
86	'''	136	'''
87		137	
88	# Dialog fuer Dateiname	138	# Dialog fuer Dateiname
89	#filename = ...	139	filename = QtGui.QFileDialog.getSaveFileName().
90		140	
91	# Configparser anlegen und Daten uebergeben	141	# Configparser anlegen und Daten uebergeben
92	#c = ...	142	c = ConfigParser.SafeConfigParser().
93		143	
		144	c.add_section('Parameter')
		145	c.set('Parameter', 'm1', str(mass1Edit.text()))
		146	c.set('Parameter', 'm2', str(mass2Edit.text()))
		147	c.set('Parameter', 'l', str(lenEdit.text()))
		148	
		149	c.add_section('Simulation')
		150	c.set('Simulation', 'dx', str(dxEdit.text()))
		151	c.set('Simulation', 'tEnd', str(tEndEdit.text()))
94		152	
95	# Configfile schreiben	153	# Configfile schreiben
96	fid = open(filename, 'w')	154	fid = open(filename, 'w')
:		:	
112		170	
113	# Werte holen	171	# Werte holen
114	m1 = float(str(mass1Edit.text()))	172	m1 = float(str(mass1Edit.text()))
115	# ...	173	m2 = float(str(mass2Edit.text())).
		174	l = float(str(lenEdit.text()))
		175	dx = float(str(dxEdit.text()))
		176	tEnd = float(str(tEndEdit.text()))
		177	
		178	# oder auch:
		179	#m = mass1Edit.text().toDouble()[0] # gibt Tupel ala (wert, OK) zurueck
116		180	
117	# Zeitachse anlegen	181	# Zeitachse anlegen
118	# t = ...	182	t = arange(0,tEnd,dx).
119		183	
120	# Simulation ausfuehren	184	# Simulation ausfuehren
121	# res ...	185	res = odeint(rhs, [0,0.3,0,0], t, args=(m1,m2,l)).
122		186	
123	# Ergebnisse plotten	187	# Ergebnisse plotten
124	fig = plt.figure()	188	fig = plt.figure()
:		:	
126	# Hier muss etwas getrickst werden: wir legen einen neuen Dialog an, auf den	190	# Hier muss etwas getrickst werden: wir legen einen neuen Dialog an, auf den
127	# matplotlib zeichnet. Der plotDialog hat unseren Hauptdialog als parent und	191	# matplotlib zeichnet. Der plotDialog hat unseren Hauptdialog als parent und
128	# ist modeless. Damit koennen wir beliebig viele Ergebnisfenster parallel	192	# ist modeless. Damit koennen wir beliebig viele Ergebnisfenster parallel
129	# darstellen und Simulationsergebnisse vergleichen..	193	# darstellen..
130	plotDialog = QtGui.QDialog(dialog)	194	plotDialog = QtGui.QDialog(dialog)

```

130 plotDialog = QDialogBox(dialog)
131 fig.canvas.parent().setParent(plotDialog)
132
133 # Ergebnisse für Laufkatze
134 ax1 = fig.add_subplot(2,1,1)
135 # ...
136
137
138 # Ergebnisse für Last
139 # ax2 = ...
140
141 # Hier wird jetzt der Dialog angezeigt und nicht mehr die show-Funktion von
142 # matplotlib aufgerufen!
143 :
144
145
146
147 # Buttons verknuepfen
148
149
150 exitButton.clicked.connect(dialog.close)

```

```

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

```