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
⁝                                                                       ⁝
17                                                                      17
18  import matplotlib.pyplot as plt                                     18  import matplotlib.pyplot as plt
19                                                                      19
                                                                        20  import configparser
                                                                        21
20                                                                      22
21  # QApplication instance is always needed (just accept sys.argv)     23  # QApplication instance is always needed (just accept sys.argv)
22  app = QtWidgets.QApplication(sys.argv)                              24  app = QtWidgets.QApplication(sys.argv)
⁝                                                                       ⁝
35  mass1_label = QtWidgets.QLabel('mass trolley', dialog)              37  mass1_label = QtWidgets.QLabel('mass trolley', dialog)
36  mass1_edit = QtWidgets.QLineEdit('0.8', dialog)                     38  mass1_edit = QtWidgets.QLineEdit('0.8', dialog)
37                                                                      39
38  # ....                                                              40  length_label = QtWidgets.QLabel('pendulum length', dialog).
                                                                        41  length_edit = QtWidgets.QLineEdit('1.2', dialog)
                                                                        42
                                                                        43  mass2_label = QtWidgets.QLabel('mass pendulum load', dialog)
                                                                        44  mass2_edit = QtWidgets.QLineEdit('0.5', dialog)
                                                                        45
39                                                                      46
40  # task 11.1.2                                                       47  # task 11.1.2
                                                                        48  step_size_label = QtWidgets.QLabel('simulation step size', dialog)
                                                                        49  step_size_edit = QtWidgets.QLineEdit('0.01', dialog)
41                                                                      50
42  # ....                                                              51  duration_label = QtWidgets.QLabel('simulation duration', dialog).
                                                                        52  duration_edit = QtWidgets.QLineEdit('10.0', dialog)
43                                                                      53
44                                                                      54
45  # task 11.1.3                                                       55  # task 11.1.3
46  # buttons                                                           56  # buttons
47  # exit_button =.                                                    57  exit_button = QtWidgets.QPushButton('Exit', dialog).
                                                                        58  simulation_button = QtWidgets.QPushButton('Simulate', dialog)
                                                                        59
                                                                        60  exit_button.clicked.connect(dialog.close)
48                                                                      61
49                                                                      62
50  # task 11.1.4                                                       63  # task 11.1.4
51  # specify layout                                                    64  # specify layout
52  layout = QtWidgets.QGridLayout()                                    65  layout = QtWidgets.QGridLayout()
53  # layout.addWidget(....                                             66  layout.addWidget(length_label, 0, 0)  # widget, row, column.
54  # ....                                                              67  layout.addWidget(length_edit, 0, 1).
                                                                        68  layout.addWidget(mass1_label, 1, 0)
                                                                        69  layout.addWidget(mass1_edit, 1, 1)
                                                                        70  layout.addWidget(mass2_label, 2, 0)
                                                                        71  layout.addWidget(mass2_edit, 2, 1)
                                                                        72  layout.addWidget(step_size_label, 3, 0)
                                                                        73  layout.addWidget(step_size_edit, 3, 1)
                                                                        74  layout.addWidget(duration_label, 4, 0)
                                                                        75  layout.addWidget(duration_edit, 4, 1)
                                                                        76
                                                                        77  layout.addWidget(simulation_button, 5, 1, QtCore.Qt.AlignRight)
                                                                        78  layout.addWidget(exit_button, 6, 1, QtCore.Qt.AlignRight)
                                                                        79
55  dialog.setLayout(layout)                                            80  dialog.setLayout(layout)
```

```
56                                                                          81
57                                                                          82
58  # task 11.1.5                                                           83  # task 11.1.5
59  # limit input characters (only float numbers should be allowed)         84  # limit input characters (only float numbers should be allowed)
60  # ....                                                                  85  length_edit.setValidator(QtGui.QDoubleValidator(length_edit)).
                                                                            86  mass1_edit.setValidator(QtGui.QDoubleValidator(mass1_edit))
                                                                            87  mass2_edit.setValidator(QtGui.QDoubleValidator(mass2_edit))
                                                                            88  step_size_edit.setValidator(QtGui.QDoubleValidator(step_size_edit))
                                                                            89  duration_edit.setValidator(QtGui.QDoubleValidator(duration_edit))
                                                                            90
61                                                                          91
62  # task 11.1.6                                                           92  # task 11.1.6
63  # set alignment                                                         93  # set alignment
64  # ....                                                                  94  length_edit.setAlignment(QtCore.Qt.AlignRight).
                                                                            95  mass1_edit.setAlignment(QtCore.Qt.AlignRight)
                                                                            96  mass2_edit.setAlignment(QtCore.Qt.AlignRight)
                                                                            97  step_size_edit.setAlignment(QtCore.Qt.AlignRight)
                                                                            98  duration_edit.setAlignment(QtCore.Qt.AlignRight)
65                                                                          99
66                                                                          100
67  # optional: set focus to the exit button                                101 # optional: set focus to the exit button
⋮                                                                           ⋮
84                                                                          118
85      # Create ConfigParser and pass data                                 119     # Create ConfigParser and pass data
86      c = configparser.ConfigParser()                                     120     c = configparser.ConfigParser()
87      c.set("XXX", "XXX", mass1_edit.text())
88      c.XXX
89                                                                          121
                                                                            122     c.add_section('Parameter')
                                                                            123     c.set('Parameter', 'm1', str(mass1_edit.text()))
                                                                            124     c.set('Parameter', 'm2', str(mass2_edit.text()))
                                                                            125     c.set('Parameter', 'l',  str(duration_edit.text()))
                                                                            126
                                                                            127     c.add_section('Simulation')
                                                                            128     c.set('Simulation', 'dt', str(step_size_edit.text()))
                                                                            129     c.set('Simulation', 't_end', str(duration_edit.text()))
90                                                                          130
91      # write config file                                                 131     # write config file
92      with open(filename, 'w') as fid:                                    132     with open(filename, 'w') as fid:
⋮                                                                           ⋮
116         print("No configuration file loaded")                           156         print("No configuration file loaded")
117                                                                         157
118     # pass values to the according LineEdit instances                   158     # pass values to the according LineEdit instances
119     # mass1_edit.setText(...).                                          159     mass1_edit.setText(c.get('Parameter', 'm1')).
120 .                                                                       160     mass2_edit.setText(c.get('Parameter', 'm2')).
                                                                            161     duration_edit.setText(c.get('Parameter', 'l'))
121                                                                         162
                                                                            163     step_size_edit.setText(c.get('Simulation', 'dx'))
                                                                            164     duration_edit.setText(c.get('Simulation', 't_end'))
122                                                                         165
123                                                                         166
124 def simulate():                                                         167 def simulate():
⋮                                                                           ⋮
131                                                                         174
132     # fetch values from the gui                                         175     # fetch values from the gui
133     m1 = float(mass1_edit.text())                                       176     m1 = float(mass1_edit.text())
134     # ....                                                              177     m2 = float(mass2_edit.text()).
                                                                            178     l = float(duration_edit.text())
```

```

    # create time array
    # t = ....

    # execute simulation (todo: use solve_ivp here) see task description.
    # res ....

    # Plot the results
    # Here we have to do some trickery: we create a new dialog on which
⋮

    # result for the trolley
    ax1 = fig.add_subplot(2, 1, 1)
    # ....


    # result for the load
    # ax2 = ....

    # Here now the dialog is displayed and no longer the show function of
    # matplotlib is called
    plot_dialog.show().


# task 11.1.7
# connect button
# simulation_button.clicked.XXX.

# task 11.2.1
# ....

#------------------------------------------------------------------------------

```

```
    dx = float(step_size_edit.text())
    t_end = float(duration_edit.text())

    # alternatively:
    # m1 = mass1_edit.text().toDouble()[0]   # returns tuple like (value, OK)

    # create time array
    t = arange(0, t_end, dx).

    # execute simulation (todo: use solve_ivp here).
    res = odeint(rhs, [0, 0.3, 0, 0], t, args=(m1, m2, l)).

    # Plot the results
    # Here we have to do some trickery: we create a new dialog on which
⋮

    # result for the trolley
    ax1 = fig.add_subplot(2, 1, 1)
    ax1.plot(t, res[:, 0], label='x').
    ax1.plot(t, res[:, 2], label='dx')

    ax1.grid(True)
    ax1.legend()
    ax1.set_ylabel('trolley')

    # result for the load
    ax2 = fig.add_subplot(2, 1, 2).
    ax2.plot(t, res[:, 1], label=r"$\varphi$")
    ax2.plot(t, res[:, 3], label=r"$\dot \varphi$")

    ax2.grid(True)
    ax2.legend()
    ax2.set_xlabel('time [s]')
    ax2.set_ylabel('load')

    # Here now the dialog is displayed and no longer the show function of
    # matplotlib is called
    plotDialog.show().


# task 11.1.7
# connect button
simulation_button.clicked.connect(simulate).

# task 11.2.1
.
open_button = QtWidgets.QPushButton('Open', dialog)
save_button = QtWidgets.QPushButton('Save', dialog)

open_button.clicked.connect(openFile)
save_button.clicked.connect(saveFile)

layout.addWidget(open_button, 7, 1, QtCore.Qt.AlignRight)
layout.addWidget(save_button, 8, 1, QtCore.Qt.AlignRight)


#------------------------------------------------------------------------------

```