

Programmieren II: Prüfung 2

20.11.2019 von 14:30 bis 15:30

Dauer: Maximal 60 Minuten

Erlaubte Hilfsmittel: Skript / Übungen / Zusammenfassungen / Bücher / alles aus Papier

Nicht erlaubt sind: Elektronische Hilfsmittel wie z.B. Laptops/Taschenrechner
Lösungen müssen **handschriftlich direkt** auf dem abgegebenen Papier erfolgen. Abgabe von
Zusatzblättern ist nicht erlaubt! Bei Platzmangel kann die Rückseite der Prüfungsblätter verwendet
werden.Name: Silja GlausPunkte: 34Note: (6)

Aufgabe 1: PyQt Grundlagen (6 Punkte)

a) Nennen Sie je ein Beispiel eines Signales für QCheckBox und QRadioButton und erstellen Sie ein kleines Beispiel (nur Code zur Initialisierung des Widgets, kein komplettes Fenster)

b) Wie kann auf den Text eines QLabel zugegriffen werden? Wie kann dieser verändert werden?

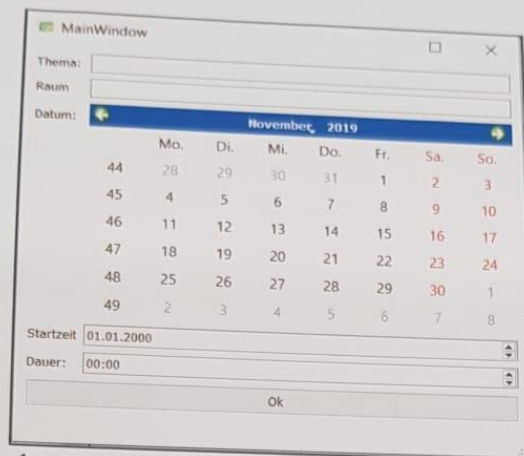
```
a) self.checkbox = QCheckBox("Bla")  
self.checkbox.stateChanged.connect (import :)  
self.checkbox.setChecked(Qt.CheckState.Checked)  
↳ self.checkbox.stateChanged.connect(self.checkbox.changed) (wo?)  
  
self.radioButton = QRadioButton("Bla2")  
self.radioButton.toggled.connect(self.radioButton.changed)
```

6

```
b) self.label = QLabel("Hallo")  
t = self.label.text()
```

Aufgabe 2 (4 Punkte)

Für das neue Programm „PyOffice“ soll ein Terminplanungs-Fenster implementiert werden. Das GUI soll noch ohne Logik versehen werden (also nur das Layout). Nach Vorgaben der Projektleitung soll das GUI so aussehen:



Hinweis: Diese Aufgabe muss ohne Qt-Designer gelöst werden.

```
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
```

```
class MainWindow(QMainWindow):
```

```
    def __init__(self):
```

```
        super().__init__()
```

```
        self.createLayout()
```

```
        self.createConnects()
```

```
    def createLayout(self):
```

```
        self.setWindowTitle("MainWindow")
```

```
        layout = QFormLayout()
```

```
        self.thema = QLineEdit()
```

```
        self.raum = QLineEdit()
```

```
        self.datum = QCalendarWidget()
```

```
        self.startzeit = QTimeEdit() QDateEdit()
```

```
        self.dauer = QTimeEdit() ← da bin ich nicht ganz sicher  
                               evtl. besser mit einem slider
```

```
        self.button = QPushButton("Ok")
```

```
layout.addRow("Thema:", self.thema)
layout.addRow("Raum:", self.raum)
layout.addRow("Datum:", self.datum)
layout.addRow("Startzeit:", self.startzeit)
layout.addRow("Dauer:", self.dauer)
layout.addWidget(self.button)
```

```
center = QWidget()
center.setLayout(layout)
```

```
self.setCentralWidget(center)
```

```
self.show()
self.raise_()
```

```
def createConnects(self):
    self.button.clicked.connect(self.buttonClicked)
```

```
def buttonClicked(self):
    print("blabla")
```

```
def main():
    app = QApplication(sys.argv)
    mainWindow = MyWindow()
    app.exec_()
```

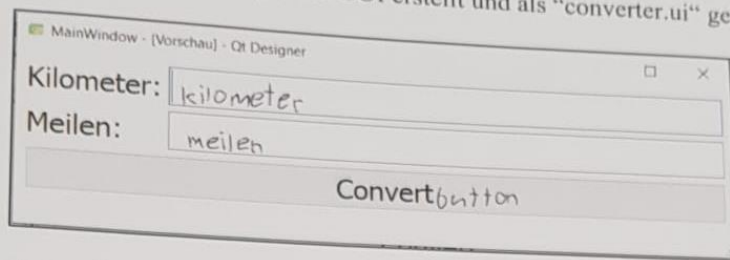
```
if __name__ == '__main__':
    main()
```

```
← def main():
    app = QApplication(sys.argv)
    mainWindow = MyWindow()
    mainWindow.raise_()
    app.exec_()
```

```
if __name__ == '__main__':
    main()
```


Aufgabe 3 (6 Punkte)

Mit dem Qt Designer wurde folgendes GUI erstellt und als "converter.ui" gespeichert:



Implementieren Sie nun die Logik, sprich ein lauffähiges Programm, welches beim Klick auf den „Convert Button“ Kilometer nach Meilen umwandelt. (Das Feld für Meilen ist «read-only»). Die Objektnamen sind dabei die folgenden:

kilometer
meilen
convertbutton

Hinweis: 1 Kilometer sind 0.62137 Meilen

Stellen Sie sicher, dass das Programm bei ungültiger Eingabe nicht abstürzt.

```
import sys
from PyQt4.QtCore import *
from PyQt4.QtGui import *
from PyQt4.uic import *
```

```
def convert():
    try:
        kilometer = kilometer.text().float()
        meilen = 0.62137 * float(kilometer)
        meilen.setText(str(meilen))
    except:
        print("keine gültige Eingabe")
```

```
def convert():
    try:
        kilometer = float(window.kilometer.text())
        window.meilen.setText(str(round(0.62137 * kilometer, 2)))
    except:
        window.meilen.setText("ungültige Eingabe")
```

```
app = QApplication([])
window = loadUi("converter.ui")
window.convertbutton.clicked.connect(convert)
window.show()
app.exec_()
```

Aufgabe 4 (6 Punkte)

Sind dieser Aussagen wahr oder falsch ? (kreuzen Sie entsprechend an)
(nur ein Kreuz pro Antwort, es gibt nur Punkte, keinen Abzug)

Wahr	Falsch	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	In numpy können einzelne Arrays (sprich Listen) nur einen einzigen Datentypen enthalten.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Die Bibliothek Pandas wird heute vor allem in der Filmindustrie ^{10/10} eingesetzt. Damit wurde unter anderem die Animations-Skripts für die Zeichentrickserie „Kung Fu Panda“ erstellt. Daher kommt auch der Name „Pandas“.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Das Modul „matplotlib“ muss zusätzlich installiert werden, es gehört nicht zum Standardumfang von Python.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Mit dem Qt Designer können GUI erstellt werden. Diese werden in XML basierten „.ui“-Files gespeichert.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Dieses Programm stürzt nach der Installation von numpy ab: <pre>import numpy as np s = np.array([1,2,3])</pre>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	In Numpy wird mit np.arange(1,5) folgendes array erstellt: <pre>array([1, 2, 3, 4, 5])</pre>

6

Aufgabe 5 (4 Punkte)

Gegeben ist eine Liste data bestehend aus Tupeln mit Kantonsnamen und der Wahlbeteiligung in Prozent der Nationalratswahlen. Daraus wird mit Pandas ein Dataframe erstellt, und die Spalten bekommen die Namen „kanton“ und „wahlbeteiligung“ (unten noch ein Teil-Screenshot des DataFrames)

```
import pandas as pd
```

```
data=[("Zürich",47.2),("Bern",49.1),("Luzern",50.9),("Uri",57.1),  
("Schwyz",53.7),("Obwalden",59.5),("Nidwalden",58.3),("Glarus",41.5),  
("Zug",53.7),("Freiburg",47.2),("Solothurn",50.2),  
("Basel-Stadt",50.4),("Basel-Landschaft",46.8),("Schaffhausen",62.6),  
("Appenzell Ausserrhoden",47.1),("Appenzell Innerrhoden",36.7),  
("St.Gallen",46.5),("Graubünden",46.0),("Aargau",48.3),  
("Thurgau",46.6),("Tessin",54.4),("Waadt",42.9),("Wallis",59.8),  
("Neuenburg",41.8),("Genève",42.9),("Jura",54.3)]
```

```
df = pd.DataFrame(data)
```

```
df.columns = ["kanton","wahlbeteiligung"]
```

	kanton	wahlbeteiligung
0	Zürich	47.2
1	Bern	49.1
2	Luzern	50.9
3	Uri	57.1
4	Schwyz	53.7

Wie kann ein neues DataFrame erstellt werden für alle Kantone mit Wahlbeteiligung grösser als 50 % ?

`df[df['wahlbeteiligung'] > 50]`

4

Aufgabe 6 (4 Punkte)

Schreiben Sie ein Python-Programm, welches überprüft, ob das Modul „numpy“ installiert ist.

```
try:
    import numpy
except ImportError:
    print("numpy ist nicht installiert")
```

oder

```
import sys
'numpy' in sys.modules
↳ True ← wenn installiert
```

Aufgabe 7 (4 Punkte)

- a) Was geschieht, wenn die unterstehende Zelle im Jupyter Notebook ausgeführt wird?

```
from matplotlib.pyplot import *
axis("equal")
plot([1,2,3,4,5],[3,3,3,2,2], "bo-")
show()
```

- b) Und was geschieht, wenn die folgende Zelle ausgeführt wird?

```
import numpy as np
a = np.array([1,2,3], dtype=np.float)
b = np.array([2,3,4], dtype=np.float)
a*b
```

- a) Es wird eine Plot-Fenster erstellt, in dem die Punkte ~~ander~~ als blaue Kreise erscheinen & verbunden sind mit einer durchgezogenen Linie.
 $[1,2,3,4,5] \rightarrow x\text{-werte}$
 $[3,3,3,2,2] \rightarrow y\text{-werte}$
 Mit `axis("equal")` sind die Koordinatenachsen gleich skaliert

- b) Die beiden Arrays werden elementweise multipliziert.
 Das Resultat ist ein Array $[2,6,12]$
 Der Datentyp des Elements ist float.