Carl-Schurz-Schule 20.03.2012
Informatik, Kurs E2Info01 Victor Hahn

Klausur

Ihr Name: _____

Maximal erreichbare Anzahl Verrechnungspunkte (VP): 53 + 6

Einziges zugelassenes Hilfsmittel: Ein Blatt DIN A4, beidseitig beliebig beschrieben

Aufgabe 1

zusammen 9 VP

Folgende Begriffe entstammen der Welt der (objektorientierten) Programmierung.
Beschreiben Sie sie möglichst genau:

a) Objekt 3 VP

b) Klasse 3 VP

c) Variable 3 VP

Aufgabe 2

zusammen 16 VP

Unter 2.a) und 2.b) sind zwei Python-Programme abgedruckt.
Nur eines davon funktioniert, das andere erzeugt eine Fehlermeldung.

Markieren Sie das funktionierende und das fehlerhafte Programm entsprechend!

Bitte bearbeiten Sie **für das funktionierende Programm:**
Was tut das Programm? Was sieht der Benutzer auf dem Bildschirm?
Wie kann er mit dem Programm interagieren? Was passiert dann?

Bitte bearbeiten Sie **für das fehlerhafte Programm:**
Erläutern Sie den oder die Fehler: Wo genau tritt er oder treten sie auf?
Nennen Sie die genaue(n) Stelle(n) im Programmcode!
Beschreiben Sie den oder die Fehler so genau wie möglich und gehen Sie auf technische Aspekte ein.

2.a)

8 VP

```
def meinefunktion(parameter):
  anzahl_Text = paulchen.get_label()
  anzahl = int(anzahl_Text)
  anzahl = anzahl + 1
  neu_Anzahl_Text = str(anzahl)
  paulchen.set_label(neu_Anzahl_Text)

import gtk
paula = gtk.Window()
paula.set_title("Ich bin ein Fenster!")
paula.show()
paulchen = gtk.Button()
paulchen.set_label("0")
paulchen.show()
paulchen.connect("clicked", meinefunktion)
paula.add(paulchen)
gtk.main()
```

2.b) 8 VP

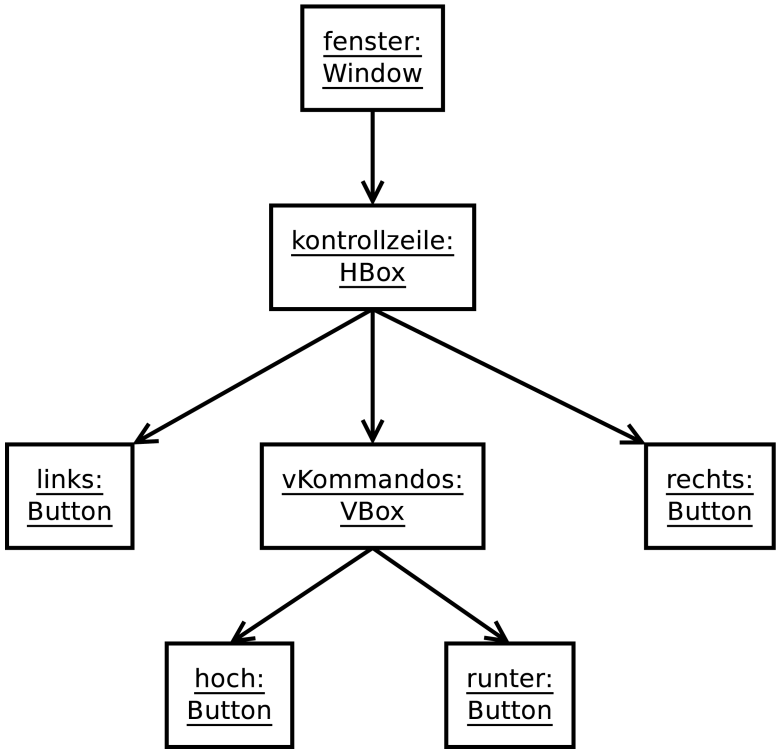
```
def action_handler(originatingWidget):
  originatingWidget.set_label("Operation succesful!")
  originatingWidget.connect("clicked", click_event)
def click_event():
  message = gtk.Label("Double operation detected")
  message.show()
  contentPane.add(message)
  parameter.set_label("A critical error occurred!")

import gtk
mainWindow = gtk.Window()
mainWindow.set_title("Mission Control")
mainWindow.show()
contentPane = gtk.VBox()
contentPane.show()
mainWindow.add(contentPane)
actionOne = gtk.Button("Initiate")
actionOne.show()
contentPane.add(actionOne)
actionOne.connect("clicked", action_handler)
actionTwo = gtk.Button("Incinerate")
actionTwo.show()
contentPane.add(actionTwo)
actionTwo.connect("clicked", action_handler)
gtk.main()
```

Aufgabe 3 8 VP

Schreiben Sie ein Python-Programm, das die in untenstehendem Objektdiagramm beschriebene Objektstruktur besitzt.

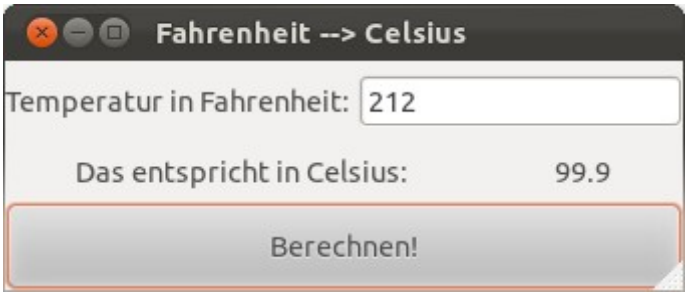
Das Programm muss keinerlei Funktionalität haben.



Aufgabe 4

20 VP

In vielen Ländern, wie beispielsweise in den USA, werden Temperaturen vorwiegend in Fahrenheit gemessen – anstatt wie bei uns in Celsius. Schreiben Sie mit Python und gtk ein Programm, das beim nächsten Amerikaurlaub hilft, den Wetterbericht zu verstehen!



In einem Fenster soll der Benutzer eine Temperatur in Fahrenheit eingeben. Nach Klick auf eine Schaltfläche bekommt er im selben Fenster die Temperatur in Celsius angezeigt.

Die Benutzeroberfläche kann, muss aber nicht so gestaltet werden, wie auf obigem Screenshot.

Sie berechnen die Temperatur in Grad Celsius aus einer Temperatur in Fahrenheit, indem Sie von dieser zunächst 32 abziehen und das Ergebnis mit 0,555 multiplizieren.

Als Formel:

$$\text{Temperatur in Grad Celsius} = (\text{Temperatur in Fahrenheit} - 32) * 0{,}555$$

Zusatzaufgabe

6 zusätzliche VP

Die Bearbeitung dieser Aufgabe ist zum Erreichen der regulären 53 VP nicht erforderlich

Der folgende kurze *Ausschnitt* eines Python-Programms entstammt dem BitTorrent-Client „Deluge“.

Übertragen Sie Ihr im Unterricht erworbenes Wissen, um Aussagen über diesen unbekannten Programmausschnitt treffen zu können:

a) Welche Objekte erzeugt der Programmausschnitt – unter welchen Variablennamen werden sie gespeichert und welchen Klassen gehören sie an?

b) Unterstreichen Sie alle Konstruktoraufrufe (und mehr nicht!)

c) Beschreiben Sie grob, wozu der hier wiedergegebene Programmcode dient, d.h. was für eine Art von Aufgabe er erfüllt.

```
hbox = gtk.HBox(spacing=5)
image = gtk.Image()
image.set_from_stock(icon, gtk.ICON_SIZE_DIALOG)
image.set_alignment(0.5, 0.0)
hbox.pack_start(image, False, False)
vbox = gtk.VBox(spacing=5)
label = gtk.Label("<b><big>" + header + "</big></b>")
label.set_use_markup(True)
label.set_alignment(0.0, 0.5)
label.set_line_wrap(True)
vbox.pack_start(label, False, False)
tlabel = gtk.Label(text)
tlabel.set_use_markup(True)
tlabel.set_line_wrap(True)
tlabel.set_alignment(0.0, 0.5)
vbox.pack_start(tlabel, False, False)
hbox.pack_start(vbox, False, False)
```