

Probeklausur

mit Musterlösung

Ihr Name: _____

Maximal erreichbare Anzahl Verrechnungspunkte (VP): 77

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

Aufgabe 1

12 VP

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

1. Programm 3 VP
Ein Programm ist eine Folge von Befehlen, die den Computer anweisen, eine bestimmte Aufgabe zu erfüllen. Programme können mit ihrem Benutzer kommunizieren, zum Beispiel per Text oder mit grafischen Benutzeroberflächen.
2. Funktion 3 VP
Ähnlich wie ein Programm ist eine Funktion eine Folge von Befehlen. Diese Befehle werden jedoch nicht direkt beim Programmstart ausgeführt, sondern erst beim Aufruf der Funktion. Hat ein Objekt eine Funktion, ist diese Funktion etwas, was das Objekt *kann*.
3. Objekt 3 VP
Ein Objekt ist eine konkrete Programmeinheit, die meistens etwas *hat* (z.B. ein Fenster einen Titel) und etwas *kann* (z.B. das Fenster sich anzeigen).
4. Klasse 3 VP
Eine Klasse ist eine abstrakte Idee, die der „Bauplan“ für Objekte sein kann. In diesem Fall beschreibt die Klasse, worum es sich bei diesen Objekten handelt.

Aufgabe 2

Was tun die nachfolgenden Python-Programme? Funktionieren sie oder stürzen sie ab?

Falls ein Programm abstürzt: Unterstreichen Sie die Zeile, in der das Programm mit einem Fehler abbricht. Beschreiben Sie den Fehler so genau wie möglich und gehen Sie auf technische Aspekte ein!

Falls ein Programm funktioniert: Was tut das Programm? Was sieht der Benutzer auf dem Bildschirm? Wie kann er mit dem Programm interagieren? Was passiert dann?

2.a)

5 VP

```
import gtk
mein_Fenster = gtk.Window()
mein_Fenster.set_title("Ich sage der Welt hallo!")
mein_Fenster.show()
begruessung = "hello, world"
mein_Fenster.add(begruessung)
gtk.main()
```

Dieses Programm stürzt ab.

In der Variablen *begruessung* befindet sich ein *str*-Objekt. Ein solches kann in gtk nicht einem Fenster hinzugefügt werden. Hierfür benötigen wir ein *Label*-Objekt.

2.b)

10 VP

```
def funktion1(parameter):
    parameter.set_label("ZONK!")
    b1.connect("clicked", funktion2)
    b2.connect("clicked", funktion2)
    b3.connect("clicked", funktion2)

def funktion2(parameter):
    b1.hide()
    b2.hide()
    b3.hide()
    info = gtk.Label("Du hast schon verloren!")
    info.show()
    breit.add(info)

import gtk
paula = gtk.Window()
paula.set_title("Der Preis ist heiss!")
paula.show()

breit = gtk.HBox()
breit.show()
paula.add(breit)

b1 = gtk.Button("Tor 1")
b1.show()
breit.add(b1)
b1.connect("clicked", funktion1)

b2 = gtk.Button("Tor 2")
b2.show()
breit.add(b2)
b2.connect("clicked", funktion1)

b3 = gtk.Button("Tor 3")
b3.show()
breit.add(b3)
b3.connect("clicked", funktion1)

gtk.main()
```

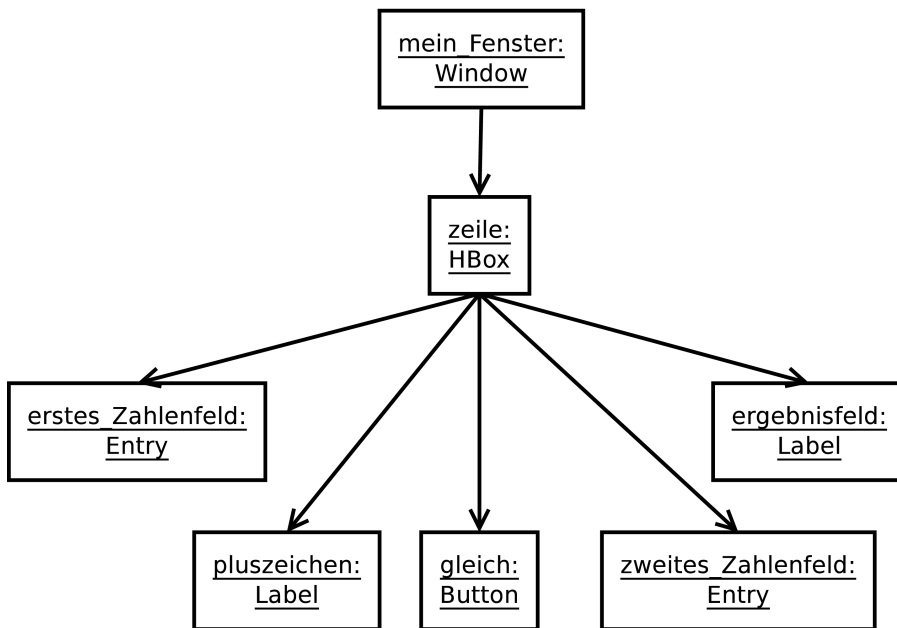
Das Programm zeigt ein Fenster mit dem Titel „Der Preis ist heiss!“ an, in dem sich nebeneinander drei Buttons mit den Aufschriften „Tor 1“, „Tor 2“ und „Tor 3“ befinden. Klickt der Benutzer auf einen dieser Buttons, ändert sich dessen Aufschrift zu „ZONK!“. Klickt er anschließend auf einen beliebigen Button, verschwinden alle drei Buttons und an ihrer Stelle erscheint der Text „Du hast schon verloren!“.

Aufgabe 3

10 VP

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

Das Programm muss keinerlei Funktionalität haben.



Programm 7 des Scripts (Addierprogramm, nur Oberfläche) ist eine geeignete Lösung.

Anmerkung: Für die Lösung dieser Aufgabe ist es nur erforderlich, dass das Programm die abgebildete Objektstruktur hat. Es ist keinerlei Funktionalität gefordert – nicht einmal, dass die grafischen Objekte mit der Funktion `show` angezeigt werden.

Aufgabe 4

10 VP

Der folgende kurze *Ausschnitt* eines Python-Programms entstammt dem Instant-Messaging-Client „Gajim“.

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

1. Was für eine Art Programmstruktur, was für ein programmiertechnisches „Ding“ wird hier programmiert? Wie heißt es? Welche Eigenschaften hat es?

Es wird eine Funktion mit dem Namen `on_message_combobox_changed` und den beiden Parametern `self` und `widget` definiert.

2. Was für Funktionsaufrufe enthält dieser Programmausschnitt? Zu welchen Objekten gehören die aufgerufenen Funktionen?

Der Programmausschnitt ruft die Funktionen `get_model` und `get_active` des Objekts `widget`, die Funktion `decode` des Objekts `model[active][0]`, die Funktion `set_text` des Objekts `message_buffer` sowie die Funktionen `draw_activity` und `draw_mood` des Objekts `self` auf.

3. Welche Variablen verwendet dieser Programmausschnitt? Beschreiben Sie die vorgenommenen Zuweisungen!

In diesem Programmausschnitt wird der Variablen `countdown_enabled` der Wert `false` zugewiesen. Der Variablen `model` wird das Ergebnis der Funktion `get_model`, der Variablen `active` das Ergebnis der Funktion `get_active` zugewiesen. Das Ergebnis der Funktion `decode` wird der Variablen `name` zugewiesen. Den Variablen `pep_dict['activity']`, `pep_dict['subactivity']`, `pep_dict['activity_text']`, `pep_dict['mood']`, `pep_dict['mood_text']` werden die Werte `preset_messages_dict[name][von 1 bis 5]` zugewiesen.

```
def on_message_combobox_changed(self, widget):
    self.countdown_enabled = False
    model = widget.get_model()
    active = widget.get_active()
    if active < 0:
        return None
    name = model[active][0].decode('utf-8')
    self.message_buffer.set_text(self.preset_messages_dict[name][0])
    self.pep_dict['activity'] = self.preset_messages_dict[name][1]
    self.pep_dict['subactivity'] = self.preset_messages_dict[name][2]
    self.pep_dict['activity_text'] = self.preset_messages_dict[name][3]
    self.pep_dict['mood'] = self.preset_messages_dict[name][4]
    self.pep_dict['mood_text'] = self.preset_messages_dict[name][5]
    self.draw_activity()
    self.draw_mood()
```

Aufgabe 5

30 VP

Schreiben Sie mit Python und gtk ein Programm „BMI-Rechner“: In einem Fenster soll der Benutzer seine Körpergröße und sein Gewicht eingeben können. Nach einem Klick auf eine Schaltfläche bekommt er seinen Body Mass Index ausgegeben.

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

Der Body Mass Index wird bestimmt durch die Formel:

$$BMI = \frac{\text{Körpergewicht in kg}}{(\text{Körpergröße in m})^2}$$

Programm 10 des Scripts erfüllt die Aufgabe.

