Test 7ab, 13.03.2013
Bewertungsschema

Punkte gibt es für zur Erfüllung der Aufgabe notwendige und an der richtigen Stelle befindlichen Code-Teile gemäß den Tabellen auf den folgenden Seiten. Der Code muss syntaktisch (von der Form her) und semantisch (vom Sinn her) korrekt sein. Teilpunkte sind möglich, werden jedoch nicht an allen Stellen gleichermaßen gegeben.

Für Code-Teile, die der Erfüllung der Aufgabe entgegenstehen, werden Punkte abgezogen, und zwar in der Größenordnung, wie durch in ähnlichem Umfang fehlerhaftes Weglassen nötiger Code-Teile Punkte verloren gegangen wären.

Wo immer verschiedene gültige Lösungen möglich sind, wird auf den alternativen Lösungsweg die Punktzahl gegeben, die für die Musterlösung gegeben würde. Auf ggf. abweichenden Umfang oder Aufwand der Lösung wird keine Rücksicht genonmen.

Im Fall von - auch nach Abschluss des Tests erkannten - Täuschungsversuchen oder Störungen des ordnungsgemäßen Ablaufs des Tests wird die Note 6 gegeben.

Gesamtpunktzahl: **33P**

Note	1	2	3	4	5	6
ab Punkten	29	24	19	14	7	0

Ergebnis / Notenspiegel:

Note	1	2	3	4	5	6
Anzahl	4	1	5	4	0	1

Durchschnitt: 2,86
Durchfallquote: 0,06%

Gruppe A („maximize“)

Befehl	**Punkte**	**Aufg.**
def neuefunktion(parameter):	3	4
parameter.get_parent().set_opacity(0.5)	2	4
def meinefunktion(parameter):	3	3
fenestra = gtk.Window()	2 - 1 Befehl, 1 Name	3
fenestra.show()	1	3
fenestra.set_title("Ich bin ein richtiges Fenster!")	1	3
button = gtk.Button()	1	3
button.show()	1	3
button.set_label("durchgucken")	1	3
fenestra.add(button)	1	3
button.connect("clicked", neuefunktion)	3	4
import gtk	1	1
fenster = gtk.Window()	1	1
fenster.show()	1	1
fenster.set_title("Reicht dir ein Fenster?")	1	1
fenster.maximize()	2	1
nein = gtk.Button()	2 - 1 Befehl, 1 Name	2
nein.show()	1	2
nein.set_label("Das ist ja langweilig!")	/	/
fenster.add(nein)	1	2
nein.connect("clicked", meinefunktion)	3	3
gtk.main()	1	1

Gruppe B („iconify“)

Befehl	**Punkte**	**Aufg.**
def neuefunktion(parameter):	3	4
parameter.get_parent().set_opacity(0.4)	2	4
mado.iconify()	2	4
def meinefunktion(parameter):	3	3
fenestra = gtk.Window()	2 - 1 Befehl, 1 Name	3
fenestra.show()	1	3
fenestra.set_title("Ich bin ein richtiges Fenster!")	1	3
button2 = gtk.Button()	1	3
button2.show()	1	3
button2.set_label("durchgucken")	1	3
fenestra.add(button2)	1	3
button2.connect("clicked", neuefunktion)	3	4
import gtk	1	1
mado = gtk.Window()	2 - 1 Befehl, 1 Name	1
mado.show()	1	1
button = gtk.Button()	1	2
button.show()	1	2
button.set_label("Das reicht mir nicht!")	1	2
mado.add(button)	1	2
button.connect("clicked", meinefunktion)	3	3
gtk.main()	1	1

Notizen zur Auslegung

nicht eingerückt: 50% auf die zwei höchst bepunkteten Befehle - dann ff
Großschreibung Klasse / Funktion: 1x -0,5 - dann ff
Leerzeichen in Name: 1x f - dann ff
connect zu nicht vorhandener Funktion: 1/3
Connect mit streunendem Anführungszeichen am Ende: 2/3
Klammern vergessen: -0,5 - immer
sinnloser Befehl führt zu Absturz: -1
Button „durchgucken“ im Hauptprogramm: 0
set_label mit Objektnamen auf Window: -1
wörtlicher Befehl als Parameter in Window.add: 0
def ohne Doppelpunkt: 2/3
def zu spät: 2/3
label „Das_reicht_mir_nicht!“: 0,5
madi statt mado: -0,5