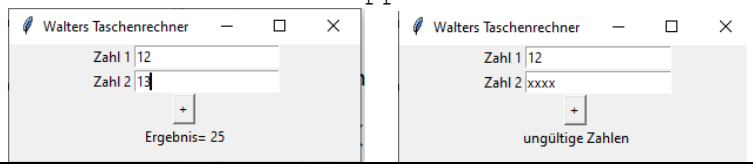
AAP11 :	Python – calculator GUI with tkinter
task	`create a calculator application` 
date:	
author:	
help:	Script Prof. Greinöcker [GR] Python „Erste Hilfe Kasten“ im moodle-Kurs „P80 tkinter decorator pattern“ Entry-Widget: https://www.python-kurs.eu/tkinter_entry_widgets.php Window title: https://www.python-forum.de/viewtopic.php?t=34021
Create the above calculator or a similar application.	
Start with a tkinter window containing a simple Label with your name. MeinFenster may be a self designed class that inherits from Tk. Explain why GUI programs are called “event driven”	Ich habe direkt einen Taschenrechner programmiert und erst danach den objektorientiert angepasst. Event-Driven, da ein event passieren muss damit sich was ändert. Bsp. In ein Feld eine Zahl eingeben.
How can you assign a function to a tkinter Button when you click the button? How do you express that the function is not immediately executed, during the definition/creation of the Button.	Eine methode schreiben und den Button auf die Methode beziehen. Warum sollte es ausgeführt werden. Die Methode wird nur ausgeführt wenn man den Button drückt.

How can you provide a message if invalid values are entered?	Mit exceptions. Meine lösung des in ein Feld reinschreiben.

```
try:
                total3 = int(e1.get()) / int(e2.get())
            except ValueError:
                print("Keine Interger!")
                e4 = Entry(frame)
                e4.grid(row = 5, column=1)
                e4.delete(0, END)
                e4.insert(0, "Bitte Zahl eingeben")
                time.sleep(2)
                frame.destroy()
                self.init_window()
```

Copy your program code and output to the addendum of your handin.

```
from tkinter import *
from math import *
import time

class MyWindow(Tk):
    def __init__(self, *args, **kargs):
        super().__init__()
        self.init_window()
        #self.l1 = Label(self, text="Test")
        #self.l1.pack()


    def init_window(self):
        #self.title = self.title("Rechner")
        #self.pack(fill=BOTH, expand=1)

        def reload():
            frame.destroy()
            self.init_window()
        def show_entry_fields():
            print("First Name: %s\nLast Name: %s" % (e1.get(), e2.get()))

        def add():
            #total = sum(int(e.get()) for e in (e1, e2))
            try:
                total=int(e1.get())+int(e2.get())
            except ValueError:
                print("Keine Interger!")
                e4 = Entry(frame)
                e4.grid(row = 5, column=1)
                e4.delete(0, END)
                e4.insert(0, "Bitte Zahl eingeben")
                time.sleep(2)
                frame.destroy()
                self.init_window()


            print("Test: %s" %(total))
            e3.delete(0, END)
            e3.insert(0, total)
```

```
        def multiply():
            try:
                total2 = int(e1.get()) * int(e2.get())
            except ValueError:
                print("Keine Interger!")
                e4 = Entry(frame)
                e4.grid(row = 5, column=1)
                e4.delete(0, END)
                e4.insert(0, "Bitte Zahl eingeben")
                time.sleep(2)
                frame.destroy()
                self.init_window()

            print("Multiply: %s" %(total2))
            e3.delete(0, END)
            e3.insert(0, total2)

        def divide():
            try:
                total3 = int(e1.get()) / int(e2.get())
            except ValueError:
                print("Keine Interger!")
                e4 = Entry(frame)
                e4.grid(row = 5, column=1)
                e4.delete(0, END)
                e4.insert(0, "Bitte Zahl eingeben")
                time.sleep(2)
                frame.destroy()
                self.init_window()

            print("divide: %s" %(total3))
            e3.delete(0, END)
            e3.insert(0, total3)

        frame = Frame(self)
        frame.pack(side = TOP)
        Label(frame, text="Zahl1").grid(row=0)
        Label(frame, text="Zahl2").grid(row=1)
        Label(frame, text="Ergebniss").grid(row=2)

        e1 = Entry(frame, justify = 'center')
        e2 = Entry(frame, justify = 'center')
        e3 = Entry(frame, justify = 'center')

        e1.grid(row=0, column=1)
        e2.grid(row=1, column=1)
        e3.grid(row=2, column=1)

        Button(frame, text='Quit', command=frame.quit).grid(row=7, column=0,
sticky=W, pady= 10)
        Button(frame, text='Show', command=show_entry_fields).grid(row=7, col
umn=1, sticky=W, pady=10)
```

	`` Button(frame, text="Reload", command=reload).grid(row=7, column=2, sticky = W) Button(frame, text="+", command=add).grid(row=5,column=0, sticky=W, pady=4) Button(frame, text="*", command=multiply).grid(row=5,column=1, sticky=W, pady=4) Button(frame, text=":", command=divide).grid(row=5,column=2, sticky=W, pady=4) if __name__ == '__main__': f1 = MyWindow() f1.title("Recher") f1.mainloop() ```
Hand in the PDF file in moodle AAP11-YourName.pdf	

Screenshots , documentation of your Classes/Objects , source code and Program output: