

L1.4 Taschenrechner

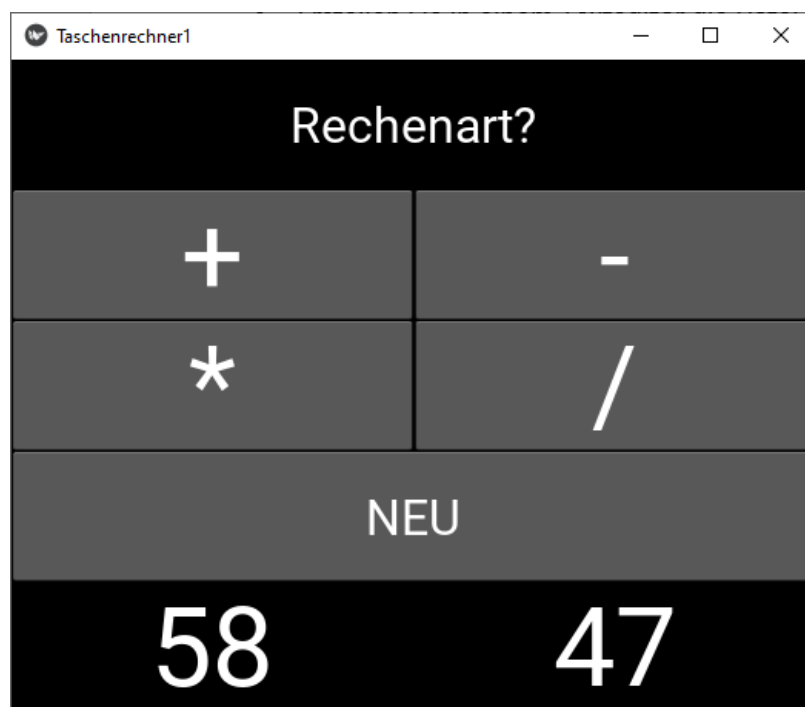
In diesem Arbeitsauftrag erstellen Sie einen Taschenrechner.

1.4.1 Einrichtung des Projektes

Führen Sie in Thonny die folgenden Schritte durch:

- Erstellen Sie in einem Texteditor die Datei main.kv.
- Erstellen Sie in Thonny die Datei main.py.
- Erstellen Sie eine Taschenrechner-App. Die genaue Beschreibung der App finden Sie unten.

1.4.2 Beispielhafter Aufbau der App (Screenshot)



1.4.3 Widgets

	ID	Text
Label	lb_ausgabe	"Klicke auf NEU"
Button	<i>nicht benötigt</i>	"+"
Button	<i>nicht benötigt</i>	"-"
Button	<i>nicht benötigt</i>	"*"
Button	<i>nicht benötigt</i>	"/"
Button	<i>nicht benötigt</i>	"NEU"
Label	lb_zahl1	" "
Label	lb_zahl2	" "

J2	BPE 8: Entwicklung von mobilen Applikationen Arbeitsauftrag	Informatik
----	---	------------

1.4.4 Ereignisse

Widget	Ereignis	Reaktion
<i>Plus-Button</i>	on_press	button_plus()
<i>Minus-Button</i>	on_press	button_minus()
<i>Mal-Button</i>	on_press	button_mal()
<i>Geteilt-Button</i>	on_press	button_geteilt()
<i>Neu-Button</i>	on_press	button_neu()

1.4.5 Reaktionen

Methode	Inhalt
button_neu()	Die zwei Zufallszahlen werden erstellt und in <i>lb_zahl1</i> und <i>lb_zahl2</i> geschrieben. In <i>lb_ausgabe</i> wird der Text „Rechenart?“ angezeigt.
button_plus()	Die beiden Zahlen werden addiert und das Ergebnis in <i>lb_ausgabe</i> geschrieben.
button_minus()	Die beiden Zahlen werden subtrahiert und das Ergebnis in <i>lb_ausgabe</i> geschrieben.
button_mal()	Die beiden Zahlen werden multipliziert und das Ergebnis in <i>lb_ausgabe</i> geschrieben.
button_geteilt()	Die beiden Zahlen werden geteilt und das Ergebnis in <i>lb_ausgabe</i> geschrieben.

1.4.6 Information: Erstellung von Zufallszahlen

Zufallszahlen können in Python folgendermaßen erzeugt werden:

```
1 import random
2 zahl = random.randint(1,7)
```

Es ist zu beachten, dass bei der Ausgabe von Zahlen in einem Label diese vorher mit der Methode *str()*, d.h. *str(zahl)* in eine Zeichenkette umgewandelt werden müssen.

1.4.7 Information: Zahlen als Attribute einrichten

Damit die erzeugten Zufallszahlen im gesamten Programm sichtbar sind, ist es sinnvoll, diese als Attribute innerhalb der Klasse *MyGridLayout* einzurichten.

```
1 def __init__(self, **kwargs):
2     super().__init__(**kwargs)
3     self.zahl1 = 0
4     self.zahl2 = 0
```

Zeile 1: Diese Methode ist der Konstruktor der Klasse, der aufgerufen wird, wenn eine Instanz von *MyGridLayout* erstellt wird.

Zeile 2: Diese Zeile ruft den Konstruktor der übergeordneten Klasse (*GridLayout*) auf. Dies stellt sicher, dass alle grundlegenden Initialisierungen, die in *GridLayout* definiert sind, auch für *MyGridLayout* durchgeführt werden.