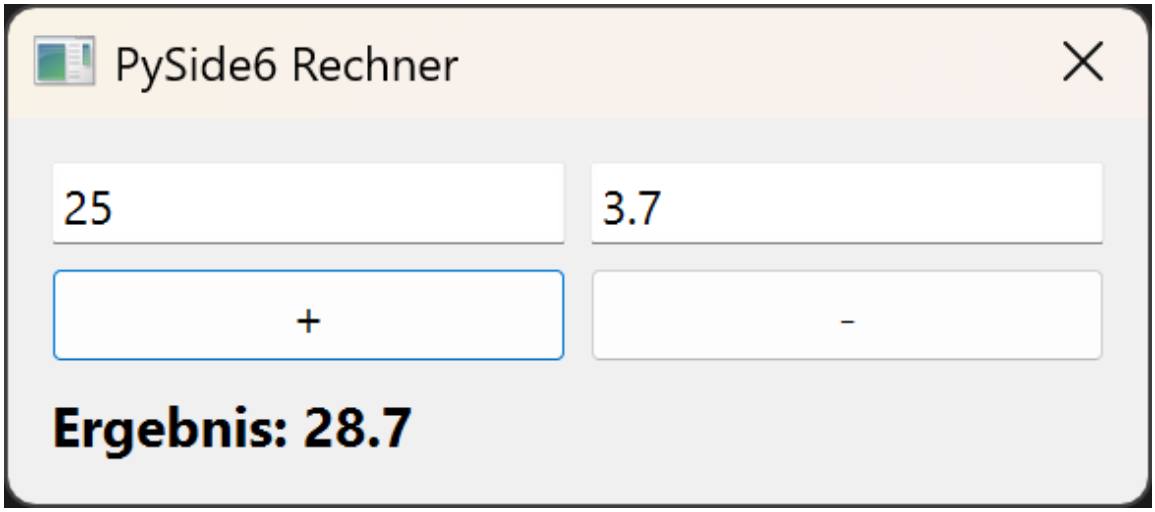
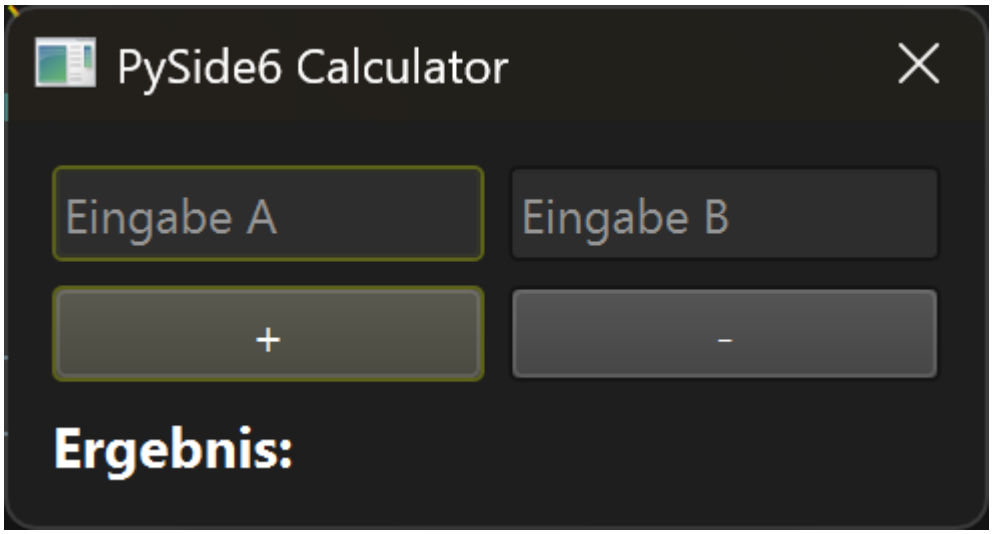
PySide6 GUI-Beispiel in Python

Dieses Jupyter Notebook zeigt, wie eine grafische Benutzeroberfläche (GUI) mit PySide6 erstellt wird. Die GUI enthält:

- Zwei Eingabefelder (`QLineEdit`) für Benutzereingaben.
- Zwei Schaltflächen (`QPushButton`) für Addition (`+`) und Subtraktion (`-`).
- Eine Beschriftung (`QLabel`), um das Ergebnis der Berechnungen anzuzeigen.



oder mit Style



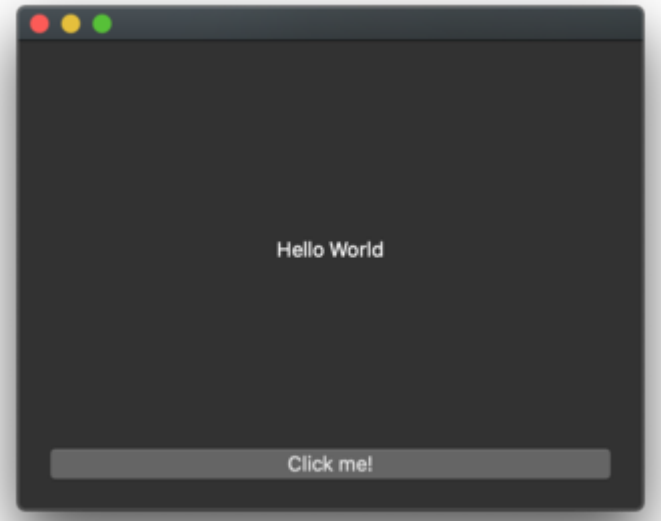
Funktionen

- **Validierung**: Die Eingaben werden überprüft, um sicherzustellen, dass sie Zahlen sind. Falls nicht, wird eine Fehlermeldung angezeigt.
- **Dynamische Ergebnisanzeige**: Das Ergebnis wird aktualisiert, sobald eine der Schaltflächen gedrückt wird.

Im Folgenden finden Sie den Python-Code, um die Anwendung zu erstellen und auszuführen.

Beispiel in der Dokumenation

Schauen sie sich das Beispiel der Dokumentation an: https://doc.qt.io/qtforpython-6/gettingstarted.html#getting-started



Bearbeitungshinweis

- Bitte straten Sie je nach Gruppe mit einerm der Grundrechenoperatoren, erweitern Sie je nach Fortschritt eigenständig um die Fehlenden, ergänzen Sie gerne auch weitere.
- Nutzen Sie, wenn Sie viel erfahrung haben die Nachfolgenden Dokumentation und vezichten Sie für das erste Beispiel auf sonstige Werkzeuge!
- Nutzen Sie wenn Sie wenig erfahrung haben alle Tools wie auch KI um einen Dialog erstellen zu können.

Mögliche Wichtige Komponenten im Code

Einen Guten Einstieg bietet https://doc.qt.io/qtforpython-6/. Folgende Kompenenten könnten wichtig sein, je nach Umsetzung:

- **`QLineEdit`** : Wird für die Eingabefelder verwendet.
- **`QPushButton`** : Buttons für Addition und Subtraktion.
- **`QLabel`** : Zeigt das Ergebnis an.
- **`QMessageBox`** : Zeigt Warnungen für ungültige Eingaben.

Es könnte auch folgende Elemente hinzukommen:

- **Layouts (`QVBoxLayout` , `QHBoxLayout`)**: Organisiert die GUI-Komponenten.
- **Klassen**: `QApplication` , `QDialog` , `QVBoxLayout` , `QHBoxLayout` , `QLabel` , `QLineEdit` , `QPushButton` , `QMessageBox` .
- **Methoden**: `exec_()` , `show()` , `setGeometry()` , `setFixedSize()` , `setLayout()` , `addWidget()` , `setText()` , `setDisabled()` , `showMessageBox()` .

Was man braucht damit man das Programm ausführen kann

1. Installieren Sie `PySide6` , falls es noch nicht installiert ist, mit folgendem Befehl:

   ```
   pip install PySide6
   ```
2. Speichern Sie den Code in einer Python-Datei, z. B. `calculator.py` , und führen Sie ihn aus mit:

   ```
   python calculator.py
   ```
3. Ein GUI-Fenster wird erscheinen, das zwei Eingabefelder, zwei Schaltflächen (`+` und `-`) und eine Ergebnisanzeige enthält.
4. Geben Sie Zahlen in die Felder ein und klicken Sie auf die Schaltflächen, um die Addition oder Subtraktion auszuführen. Ungültige Eingaben lösen eine Fehlermeldung aus.

Grobe Idee zum Code

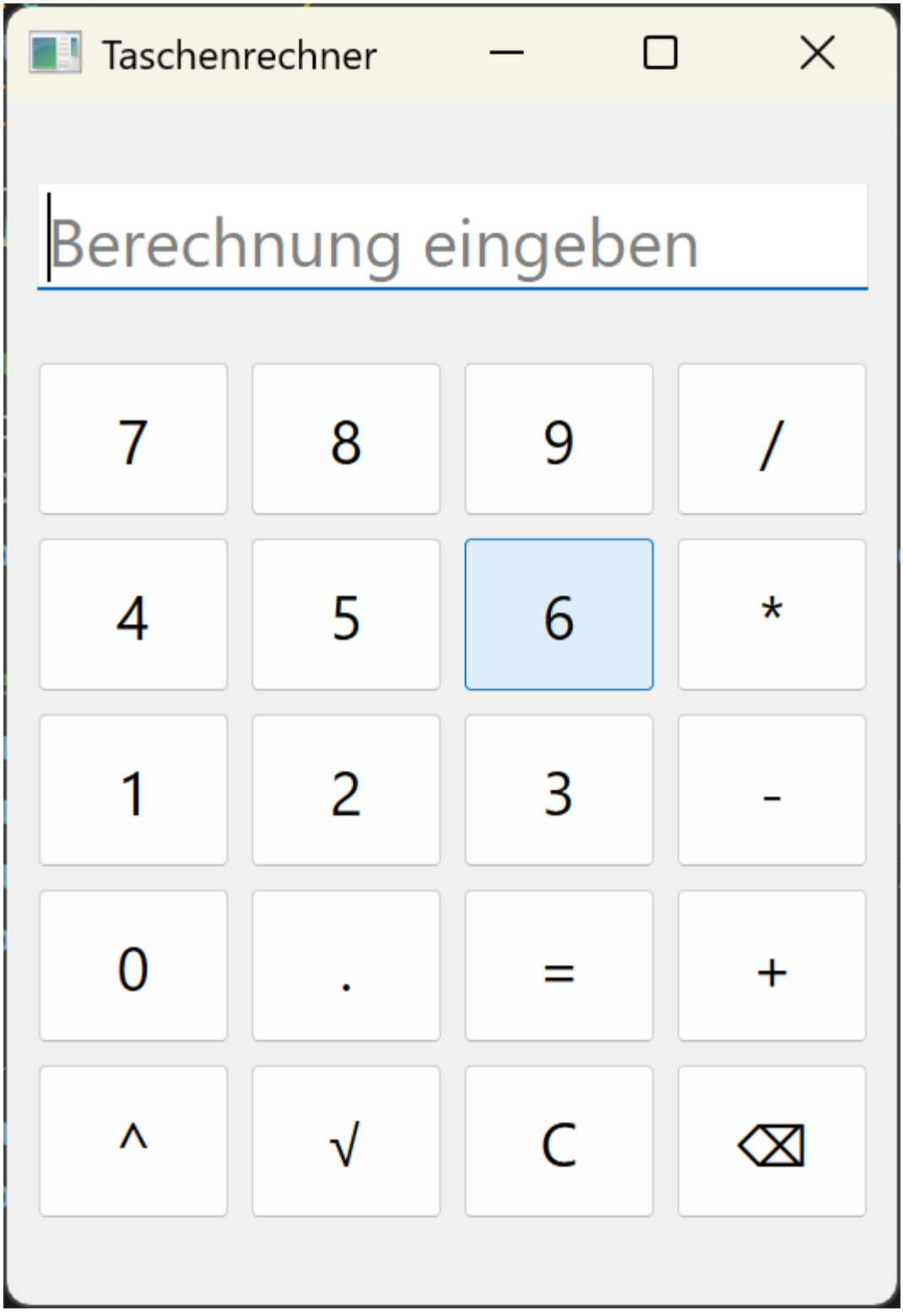
Eine Klasse z.b. CalculatorDialog Eine **init** Methode die beim Ausführen des Programms direkt aufgerufen wird und von der aus alle "started", in der Regel kann hier direkt bei kleinen Beispielen, besser in einer eigenen Methode auch das Fenster aufgebaut werden. Einige Methode zum Eingabeprüfen, zum Rechnen

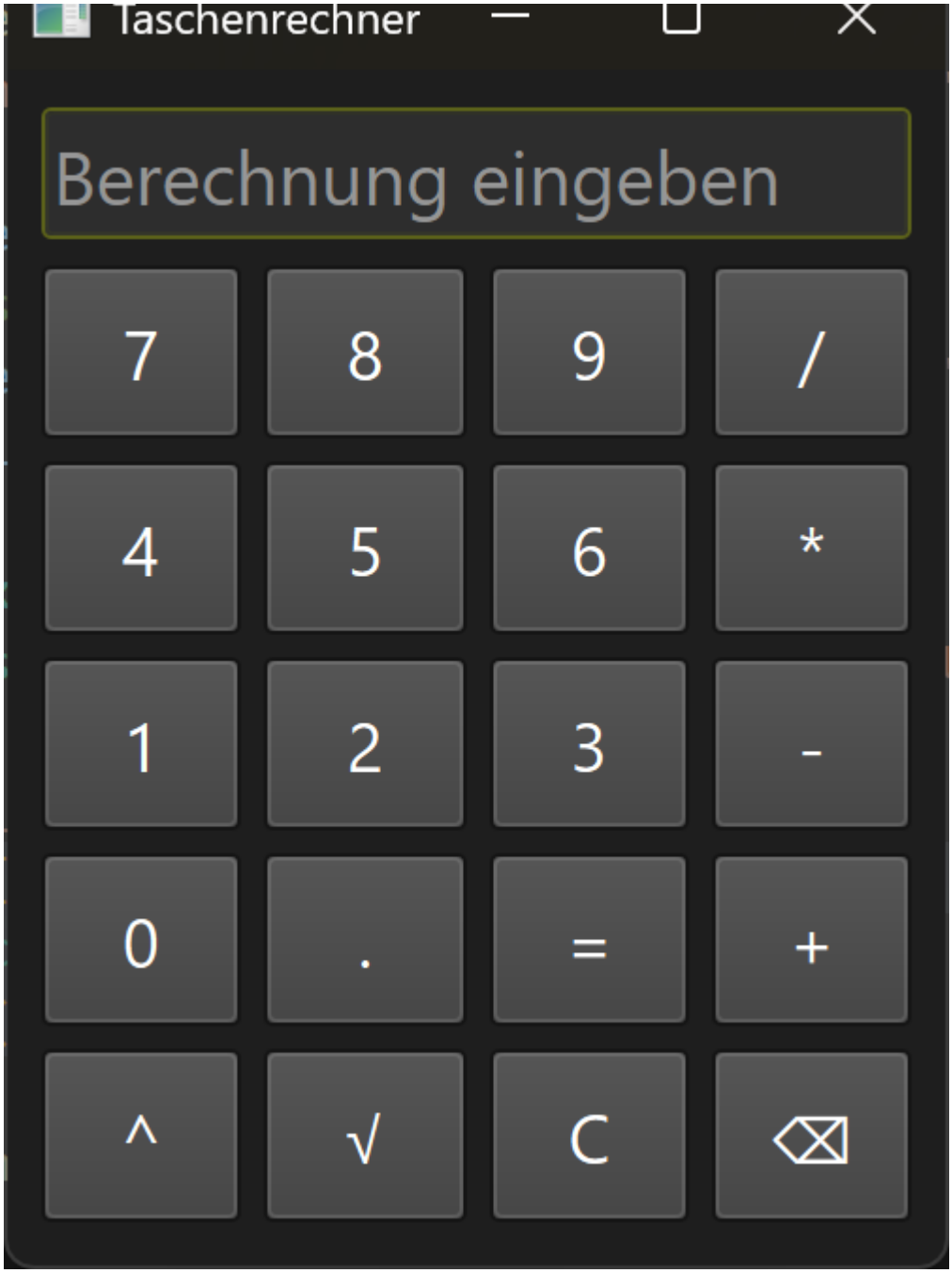
In [ ]:

```
from PySide6.QtWidgets import QApplication, QDialog, QVBoxLayout, QHBoxLayout, QLab
import sys

class CalculatorDialog(QDialog):
    def __init__(self):
        # Initialisiere QDialog

        #setze ein Titel


        # Layouts


        # Eingabefelder


        # Ergebnis-Label


        # Buttons


        # Input Layout


        # Button Layout


        # Hinzufügen zum Hauptlayout



        # Button-Events verbinden


    def validate_inputs(self):
        """Validiert die Eingaben und zeigt Fehler an, wenn etwas nicht stimmt."""


    def add_numbers(self):
        """Führt Addition durch und zeigt das Ergebnis an."""

    def subtract_numbers(self):
        """Führt Subtraktion durch und zeigt das Ergebnis an."""


if __name__ == "__main__":
    app = QApplication(sys.argv)
    dialog = CalculatorDialog()
    dialog.show()
    sys.exit(app.exec())
```

Unser Ziel ist am einde einen Taschenrechner wie folgt Programmieren und den Code verstehen zu können. Wenn Sie also schneller sind dann gerne schon einmal weiter daran arbeiten, wichtig, aber bitte zuerst den Obigen Code verstehe, hierzu dann auch nun gerne mit KI arbeiten und sich den Code erklären bzg. DOkumentieren lassen!

Idee fürs "goßee" Ziel der ersten GUI:




Taschenrechner
Berechnung eingeben
7
8
9
/
4
5
6
*
1
2
3
-
0
.
=
+
^
√
C
⌫