

LU10.A01 - Taschenrechner als Modul

Ausgangslage

In der [Aufgabe LU09.A01](#) haben Sie einen Taschenrechner programmiert.

calc.py

```
import math

def add(num1, num2):
    """
    Addition of two numbers
    :param num1: number 1 for
    calculation
    :param num2: number 2 for
    calculation
    :return: result of calculation
    """
    return num1 + num2

def sub(num1, num2):
    """
    Substracts two numbers
    :param num1: number 1 for
    calculation
    :param num2: number 2 for
    calculation
    :return: result of calculation
    """
    return num1 - num2

def mul(num1, num2):
    """
    multiply of two numbers
    :param num1: number 1 for
    calculation
    :param num2: number 2 for
    calculation
    :return: result of calculation
    """
    return num1 * num2
```

```
def div(num1, num2):
    """
    division of two numbers
    :param num1: number 1 for
    calculation
    :param num2: number 2 for
    calculation
    :return: result of calculation
    """
    return num1 / num2

def pow(num1):
    """
    Squares two numbers
    :param num1: number to square
    :return: result of calculation
    """
    return num1 * num1

def sqrt(num1):
    """
    return the root of a number
    :param num1: number to get the root
    of
    :return: result of calculation
    """
    return math.sqrt(num1)

def main():
    result_add = add(5, 5.5)
    result_div = div(10, 3)
    result_mul = mul(3, 3)
    result_sub = sub(10, 4.4)
    result_pow = pow(23)
    result_sqrt = sqrt(81)

    print(result_add)
    print(result_div)
    print(result_mul)
    print(result_sub)
    print(result_pow)
    print(result_sqrt)

if __name__ == '__main__':
```

```
main()
```

Aufgabe

In dieser Aufgabe werden sie diesen Taschenrechner nun als Modul für ein neues Projekt einbinden. Nehmen Sie dazu die Github-Classroom-Aufgabe an.

Teilaufgabe 1

Erstellen Sie das File `calc.py` im Homeverzeichnis (`m319-lu10-a01-calculator-username`) Ihres Projekts.



Sie werden von PyCharm gefragt, ob Sie das File zu Git hinzufügen möchten. Klicken Sie auf **Add**



und ergänzen Sie den [Code](#) von oben.

Passen Sie `Teilaufgabe1.py` so an, damit der Code funktioniert:

```
#Add import statement for calc

def main():
    print(calc.add(5,5))
    print(calc.mul(5, 5))

if __name__ == '__main__':
    main()
```

Testen Sie die Implementierung mit `Teilauftrag1_test.py`

Teilaufgabe 2

Importieren Sie das Modul `calc.py` so, dass folgender Code-Schnippel funktioniert:

```
#Add import statement for calc

def main():
    print(c.add(5,5))
    print(c.mul(5, 5))

if __name__ == '__main__':
    main()
```

Testen Sie die Implementierung mit `Teilauftrag2_test.py`

Teilaufgabe 3

Importieren Sie das Modul `calc.py` so, dass folgender Code-Schnippel funktioniert:

```
#Add import statement for calc

def main():
    print(add(5,5))
    print(mul(5, 5))

if __name__ == '__main__':
    main()
```

Testen Sie die Implementierung mit `Teilauftrag3_test.py`

Teilaufgabe 4

Importieren Sie `add` und `mul` das Modul `calc.py` so, dass folgender Code-Schnippel funktioniert:

```
#Add import statement for add as addition and
mul as multiplication from calc

def add(num1, num2):
    """
    Builds the sum of two numbers and prints
    the result
    :param num1: first number
    :param num2: second number
    :return: None
```

```
    '''
    print(num1 + num2)

def mul(num1, num2):
    '''
    Multiplies of two numbers and prints the
    result
    :param num1: first number
    :param num2: second number
    :return: None
    '''
    print(num1 * num2)

def main():
    add(5,5)
    mul(5, 5)
    print(addition(5,5)) # function add from
    module calc
    print(multiplication(5,5)) # function mul
    from module calc

if __name__ == '__main__':
    main()
```

Testen Sie die Implementierung mit `Teilauftrag4_test.py`

Teilaufgabe 5

Importieren Sie das Modul `calculator` aus dem **Paket** `math_operations` damit der Codeschnipsel funktioniert.

```
#Add import statement for calculator in
math_operations package

def main():
    print(calculator.add(5,5)) # function add
    from module calc
    print(calculator.mul(5,5)) # function mul
    from module calc

if __name__ == '__main__':
    main()
```

Testen Sie die Implementierung mit `Teilauftrag5_test.py`

Teilaufgabe 6

Importieren Sie `add` und `mul` aus dem Modul `calculator` aus dem **Paket** `math_operations` damit der Codeschnipsel funktioniert.

```
#Add import statement for add and mul from  
calculator in math_operations package  
  
def main():  
    print(add(5,5)) # function add from  
lib.calculator calc  
    print(mul(5,5)) # function mul from  
lib.calculator calc  
  
if __name__ == '__main__':  
    main()
```

Testen Sie die Implementierung mit `Teilauftrag6_test.py`

⇒ *GitHub Repo für externe Besucher*

GitHub Repository <https://github.com/templates-python/m319-lu10-a01-calculator>

Lernende am BZZ müssen den Link zum GitHub Classroom Assignment verwenden

[M319-LU10](#)



© Kevin Maurizi

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/de/modul/m319/learningunits/lu10/aufgaben/taschenrechner?rev=1750657505>

Last update: **2025/06/23 07:45**

