

LU05.A10 - Einfacher Decorator

Erstellen Sie einen einfachen Decorator, der die Ausführung einer Funktion zählt.

Aufgabenstellung

Ihr Ziel ist es, einen Decorator namens `count_calls` zu schreiben, der die Anzahl der Aufrufe für jede Funktion, die er dekoriert, zählt. Der Decorator soll bei jedem Aufruf der dekorierten Funktion eine Nachricht ausgeben, die besagt, wie oft die Funktion bisher aufgerufen wurde.

Code-Vorlage

main.py

```
def count_calls(original_function):
    """
    Ein Decorator, der die Anzahl der Aufrufe einer Funktion zählt.

    Args:
        original_function (function): Die Funktion, deren Aufrufe
        gezählt werden sollen.

    Returns:
        function: Eine dekorierte Version der ursprünglichen Funktion,
        die die Anzahl der Aufrufe zählt.
    """
    ...

@count_calls
def my_function():
    print("Ich tue etwas")

# Testen Sie Ihren Decorator
if __name__ == "__main__":
    my_function()
    my_function()
```

Vorgehen

1. Akzeptiere das GitHub Classroom Assignment
2. Kclone dein persönliches Repository in die Entwicklungsumgebung
3. Implementiere den Decorator `count_calls` gemäß der Anleitung

Abgabe

Die Abgabe der Lösung erfolgt als Push in das persönliche GitHub-Repository.

⇒ *GitHub Repo für externe Besucher*

GitHub Repository <https://github.com/templates-python/m323-lu05-a10-decorator>

Lernende am BZZ müssen den Link zum GitHub Classroom Assignment verwenden

 © Kevin Maurizi

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

Permanent link:
<https://wiki.bzz.ch/modul/m323/learningunits/lu05/aufgaben/decorator>

Last update: **2024/03/28 14:07**

