

LU05.L11 - Decorator mit Benutzerobjekten

main.py

```
class User:

    def __init__(self, username, permission_level):
        self.username = username
        self.permission_level = permission_level

    def check_permission(required_permission):
        """
        Ein Decorator, der das Berechtigungslevel des Benutzers überprüft.
        Druckt f'{user.username} hat nicht genügend Berechtigungen.', wenn
        das Berechtigungslevel nicht ausreicht.

        Args:
            required_permission (int): Das erforderliche
            Berechtigungslevel.

        Returns:
            function: Eine dekorierte Funktion, die nur ausgeführt wird,
            wenn das Berechtigungslevel ausreicht.
        """

        def decorator(original_function):
            def wrapper(user, *args, **kwargs):
                if user.permission_level >= required_permission:
                    return original_function(user, *args, **kwargs)
                else:
                    print(f'{user.username} hat nicht genügend
            Berechtigungen.')

            return wrapper

        return decorator

    @check_permission(2)
    def view_profile(user):
        print(f'{user.username} kann das Profil anzeigen.')

    @check_permission(4)
    def edit_profile(user):
        print(f'{user.username} kann das Profil bearbeiten.')
```

```
# Testen Sie Ihren Decorator
if __name__ == '__main__':
    alice = User('Alice', 3)
    bob = User('Bob', 1)

    view_profile(alice)
    edit_profile(alice)
    view_profile(bob)
    edit_profile(bob)
```



© Kevin Maurizi

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

Permanent link:
<https://wiki.bzz.ch/modul/m323/learningunits/lu05/loesungen/decorator2>

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

