Lösung 6 - Duck-Typing nutzen

Code-Analyse

1. Duck-Typing erfolgt über die Methoden `who_are_you` sowie `walk`, `run` und `fly`.
2. Dank der Funktion
   - `hasattr` kann dynamisch überprüft werden, ob ein Objekt über eine bestimmte Methode verfügt.
   - `getattr` kann diese Methode dynamisch aufgerufen werden.

Repetition-Vererbung

animals.py

```
class Animal:
    def who_are_you(self):
        return f"I'm a {self.__class__.__name__}"


class Dog(Animal):
    def walk(self):
        print(f"{self.who_are_you()} and I'm walking.")


class Duck(Animal):
    def run(self):
        print(f"{self.who_are_you()} and I'm running.")


class Cup(Animal):
    pass


class Man(Animal):
    pass


class Bird(Animal):
    def fly(self):
        print(f"{self.who_are_you()} and I'm flying.")


def please(do, obj):
    method = do
    if hasattr(obj, method):
        getattr(obj, method)()
    else:
        print(f"I can not {method} because {obj.who_are_you()}.")
```

```
if __name__ == '__main__':
    # Wir müssen nicht wissen, was das Objekt ist. Aber wir wollen das Objekt nur etwas tun lassen, wenn es das kann.
    please("walk", Dog())
    please("run", Duck())
    please("walk", Cup())
    please("fly", Man())
    please("fly", Bird())
```

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

Permanent link:
https://wiki.bzz.ch/modul/archiv/m320/learningunits/lu99/loesungen/lu08-aufgabe_5

Last update: **2026/08/30 14:06**