# Aufgabe 3 - Komposition implementieren

### Ziel

- Sie können eine Komposition korrekt implementieren.

## Hinweise

- Die lauffähige Lösung ist der Lehrperson zu präsentieren.

## Aufträge

1. Erstellen Sie die Klassen gemäss Klassendiagramm aber noch ohne Codierung des “Innenlebens“ der Methoden.
2. Ergänzen Sie die `get`- und `set`-Methoden der Klassen mit dem entsprechenden Code (falls möglich Properties verwenden).
3. Implementieren Sie die `print`-Methode, so dass die Ausgabe derjenigen in Abb. 2 entspricht.
4. Implementieren Sie nun die `main`-Methode gemäss dem gezeigten Sequenzdiagramm.
    - Löschen Sie das `Flower`-Objekt, im Diagramm mit “x” markiert.
    - Rufen Sie dann später das `Flower`-Objekt nochmals auf. Die Auswirkung studieren Sie bitte nach der Ausführung des Programms (nächste Teilaufgabe).
5. Führen Sie nun die `main`-Methode aus. Welche Feststellung können Sie machen?

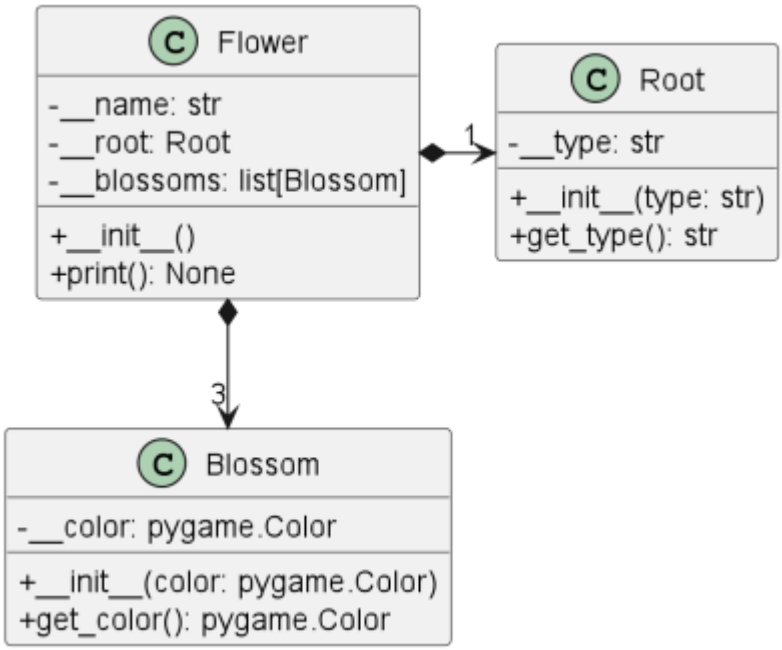


Abb. 1: Klassendiagramm

```
Blume: - Name 'Wunderblume'
       - Wurzel 'Tiefwurzel'
       - 1. Blüte, pygame.Color(255, 0, 0, 255)
       - 2. Blüte, pygame.Color(0, 0, 255, 255)
       - 3. Blüte, pygame.Color(255, 255, 0, 255)
```

Abb. 2: Ausgabe

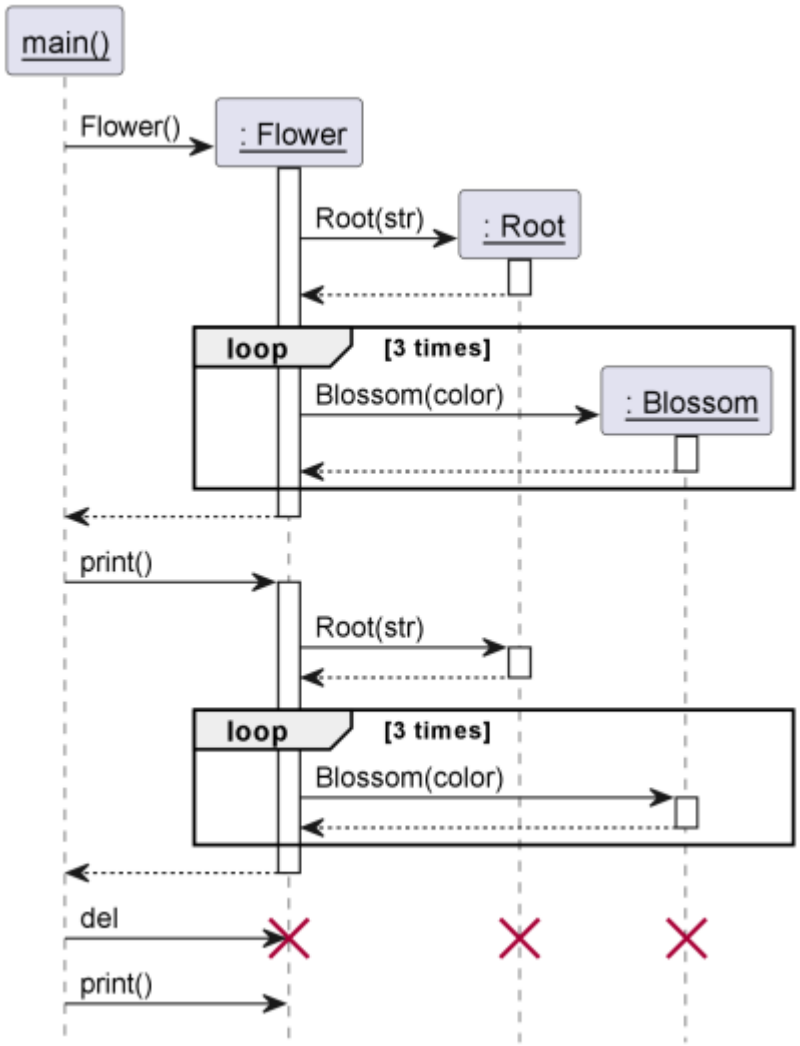



Abb. 3: Sequenzdiagramm

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

Permanent link:
**https://wiki.bzz.ch/modul/archiv/m320/learningunits/lu98/aufgaben/lu6-aufgabe_3**

Last update: **2026/08/30 14:09**