

Problemsituation: Informationen und Fehlermeldungen werden gleich formatiert angezeigt.

```
1 System started
2 Disk not found
3 Connection established
4 Permission denied
```

Lösung: Wir schreiben zwei Klassen InfoLog und ErrorLog, die von der gemeinsamen Superklasse LogEntry erben. Die Formatierung (durch die Methode display()) wird durch die beiden Klassen unterschiedlich implementiert.



Polymorphie

Verschiedene Subklassen, die eine Methode von der Superklasse erben, können unterschiedliche Implementierungen der Methode haben. Damit ist es möglich, in verschiedenen Subklassen unterschiedliche Funktionalitäten bei Aufruf der Methode zu erreichen. Dieses Konzept nennt man in der objektorientierten Programmierung **Polymorphie**.

```
1 class LogEntry:
2     def __init__(self, message):
3         self.message = message
4     def display(self):
5         print(self.message)
6
7 class InfoLog(LogEntry):
8     def display(self):
9         print(f"[INFO] {self.message}")
10
11 class ErrorLog(LogEntry):
12     def display(self):
13         print(f"[ERROR] {self.message} (!!!)")
14
15 log1 = InfoLog("System started")
16 log2 = ErrorLog("Disk not found")
17 log3 = InfoLog("Connection established")
18 log4 = ErrorLog("Permission denied")
19
20 log1.display()
21 log2.display()
22 log3.display()
23 log4.display()
```