

Kapitel 14.1: Sortierung – Einführung

In Lektion 3 haben wir gelernt, dass man eine Liste mit der Methode `sort()` sortieren kann – Integer nach Größe und Strings alphabetisch. Wenn man mit Klassen und Objekten arbeitet, die viele verschiedene Attribute haben, dann reicht das aber normalerweise nicht aus. Nach welchem Attribut soll überhaupt sortiert werden? Und ist das wirklich in jedem Anwendungsfall gleich?

Um dieses Problem zu lösen, gibt es mehrere Möglichkeiten, die Sortierung zu bestimmen.

main.py

```
from deck import Deck

# Hier beginnt der Hauptteil
deck = Deck()
deck.mischen()
deck.sortieren1() # Rufe hier auch die anderen Sortier-Funktionen von Deck
auf!
```

karte.py

```
class Karte:
    def __init__(self, farbe, symbol, wert):
        self.farbe = farbe
        self.symbol = symbol
        self.wert = wert

    def get_farbe(self):
        return self.farbe

    def get_symbol(self):
        return self.symbol

    def get_wert(self):
        return self.wert

    def print(self):
        print(self.farbe + ", " + self.symbol + ", " + str(self.wert))

    def __lt__(self, other):
        if self.farbe < other.get_farbe():
            return True
        if self.farbe > other.get_farbe():
            return False

        if self.wert < other.get_wert():
            return True
```

```
if self.wert > other.get_wert():  
    return False  
  
return self.symbol < other.get_symbol()
```

deck.py

```
import random  
from karte import Karte  
  
class Deck:  
    def __init__(self):  
        self.farben = ("Herz", "Karo", "Pik", "Kreuz")  
        self.symbole = (  
            ("7", 7), ("8", 8), ("9", 9), ("10", 10), ("Bauer", 10),  
            ("Dame", 10), ("König", 10), ("As", 11))  
        self.karten = []  
  
        self.karten_erzeugen()  
  
    def karten_erzeugen(self):  
        print("Deck erzeugen")  
        for f in self.farben:  
            for (s, w) in self.symbole:  
                k = Karte(f, s, w)  
                self.karten.append(k)  
        self.drucken()  
  
    def sortieren1(self):  
        print("Sortierung über karten.__lt__ Methode")  
        self.karten.sort()  
        self.drucken()  
  
    def sortieren2(self):  
        print("Sortierung nach Wert aufsteigend")  
        self.karten.sort(key=lambda x: x.get_wert())  
        self.drucken()  
  
    def sortieren3(self):  
        print("Sortierung nach Farbe absteigend")  
        self.karten.sort(key=lambda x: x.get_farbe(), reverse=True)  
        self.drucken()  
  
    def sortieren4(self):  
        print("Übliche Karten-Sortierung")  
        self.karten.sort(key=lambda x: self.get_sort_value(x))  
        self.drucken()  
  
    def get_sort_value(self, karte):  
        v = (len(self.farben) - self.farben.index(karte.get_farbe())) * 10
```

```
        i = len(self.symbole)
        for (s, w) in self.symbole:
            if s == karte.get_symbol():
                v += i
            i -= 1
        return v

def mischen(self):
    print("Deck mischen")
    random.shuffle(self.karten)
    self.drucken()

def drucken(self):
    print("-----")
    for k in self.karten:
        k.print()
    print("-----")
    print("")
```