OOP

```
class Bankkonto:
    def __init__(self,startbetrag):
        self.kontostand  =  startbetrag

    def einzahlung(self, betrag):
         self.kontostand  =  self.kontostand + betrag

    def auszahlung(self, betrag):
        self.kontostand  =  self.kontostand - betrag

    def anzeigen(self):
        print(self.kontostand)


konto1  =  Bankkonto(100)
konto1.anzeigen( )
konto1.einzahlung(200)
konto1.anzeigen( )
konto1.auszahlung(125)
konto1.anzeigen( )
```

OOP

```
class Auto:
    benzinstand = 0.0    # Tankfuellung in Liter
    kilometerstand = 0.0

    def __init__(self, typ, verbrauch):
        self.verbrauch = float(verbrauch)
        self.typ = typ

    def tanken(self, liter):
        self.benzinstand + =  liter
        print("  Benzinstand des ",self.typ ," nach dem Tanken: ", self.benzinstand, "
                                                                                Liter")

    def fahren(self, kilometer):
        print("Ich soll ", kilometer, " km mit dem ",self.typ   ," fahren")
        verbrauch_fahrt = (kilometer/100)*self.verbrauch
        if self.benzinstand < verbrauch_fahrt:
            print("  So weit kann ich nicht mit dem ",self.typ  ," fahren")
            print("  Benzinstand:", self.benzinstand,
                  " Verbrauch für Fahrt wäre:", verbrauch_fahrt)
            return
        else:
            self.benzinstand - =  verbrauch_fahrt
            self.kilometerstand + =  kilometer
            print("  Ich bin mit dem ", self.typ, kilometer, " Kilometer gefahren.")
            print("  Neuer Tachostand: ", self.kilometerstand)
            print("  Neuer Benzinstand: ", self.benzinstand)



A1 = Auto("2er Cabrio", 8.0)
A2 = Auto("3er Touring",6.0)
A1.tanken(60)
A2.tanken(50)
A1.fahren(500)
A2.tanken(20)
A2.fahren(300)
A1.fahren(350)
```

Zufallsstreckenzug

```
from tkinter import *
from random import *

fenster  =  Tk ( )
flaeche  =  Canvas(fenster, width = 600, height = 400, bg = 'gray')
flaeche.pack ( )

def circle(x,y,r):
    flaeche.create_oval(x-r,y-r,x+r,y+r,fill = "white", outline = "blue")



class punkt:
  def __init__(self,x,y):
      self.x  =  x
      self.y  =  y

  def plotten(self):
      circle(self.x,self.y,5)



punkteliste  =  [ ]
m  =  10
for n in range(m):
  punkteliste.append( punkt(randint(1,600),  randint(1,400)) )

for n in range(m):
    punkteliste[n].plotten ( )


for n in range(m-1):
  flaeche.create_line(punkteliste[n].x, punkteliste[n].y, punkteliste[n+1].x,
                                                  punkteliste[n+1].y, fill = "white", width = 1)
```

geschlossener Streckenzug

```
for n in range(m):
      flaeche.create_line(punkteliste[n].x, punkteliste[n].y, punkteliste[(n+1)%m].x,
        punkteliste[(n+1)%m].y, fill = "white", width = 1)
```

```
from  math  import *

def lange( ):
  summe  =  0
  for n in range(m):
     summe  =  summe+sqrt(
          (punkteliste[n].x - punkteliste[(n+1)%m].x)**2 +
          (punkteliste[n].y - punkteliste[(n+1)%m].y)**2  )
  return summe
```

nach einem Attribut sortieren

```
punkteliste  =  sorted(punkteliste, key  =  lambda c: c.x)
                                          #    in  c: c.x   ist der Name c beliebig
                                          #     absteigend mit   sorted (...  ,reverse = True)
```

Grafik

```
from tkinter import *
import time
import math
import random
master  =  Tk( )
w  =  Canvas(master, width = 600, height = 600, bg = 'white')
w.pack( )
anzahl = 5


class kreis:
    def __init__(self,x,y,r):
        self.x = x
        self.y = y
        self.r = r
        self.o = w.create_oval(self.x-self.r,self.y-self.r,self.x+self.r,self.y+self.r,
                                                                    outline = "blue")

    def move(self):
        w.delete(self.o)
        self.x = self.x+random.randint(1,8)
        self.y = self.y+random.randint(1,8)
        self.o = w.create_oval(self.x-self.r,self.y-self.r,self.x+self.r,self.y+self.r,
                                                                    outline = "blue")

L =  []
for k in range (0,anzahl):
    x = random.randint(10,50)
    y = random.randint(10,50)
    C = kreis(x,y,10)
    L.append(C)


while True:
    for n in range (0,anzahl):
        L[n].move ( )
        time.sleep(0.001)
        w.update ( )
```

Liste mit Objekten sortieren

```
class Student:
    def __init__(self, Vorname, Buchstabe, Alter):
        self.Vorname=Vorname
        self.Buchstabe=Buchstabe
        self.Alter=Alter

L = [Student('John', 'A', 15), Student('Jane', 'B', 12), Student('Dave', 'B', 10)]

L = sorted(L, key=lambda Student: Student.Alter)

for n in range (0,len(L)):
    print(L[n].Vorname, L[n].Alter)
```