Aufgabe 1

```
1  class Fantasy:
2
3      n = 0
4
5      def __init__(self, x):
6          self.x = x
7          Fantasy.n += 1
8
9      def add(self, other):
10         return Fantasy(self.x + other.x)
11
12     def minus_one(self):
13         self.x = self.x - 1
14
15     def size():
16         return Fantasy.n
17
18 a = Fantasy(4)
19 b = Fantasy(3)
20 c = a.add(b)
21 b.minus_one()
22
23 print(a.x)
24 print(b.x)
25 print(c.x)
26 print(Fantasy.size())
```

(a) Welche Ausgabe(n) macht das folgende Python-Modul?

4
2
7
3

(b) Welche Eigenschafte(n) hat ein `Fantasy`-Objekt?

x

(c) Welche Eigenschafte(n) hat die `Fantasy`-Klasse?

n

(d) Welche Methode(n) hat ein `Fantasy`-Objekt?

add, minus_one

(e) Welche Methode(n) hat die `Fantasy`-Klasse?

size

Aufgabe 2

```
class A:

    c = 1

    def __init__(self, a):
        self.a = a

    def __str__(self):
        return 'a={0}'.format(self.a)

    def calc(self, x):
        return self.a * x

class B:

    c = 10

    def __init__(self, a):
        self.a = a

    def __str__(self):
        return 'a={0}'.format(self.a)

    def calc(self, x):
        return self.a + x

d = A(4)
e = B(5)

print(d)
print(e)
print(A.c + B.c)
print(d.calc(2))
print(e.calc(2))
```

```
a=4
a=5
11
8
7
```

Aufgabe 3

```
class Personal:

    L = []

    def __init__(self, name, abteilung, lohn):
        self.name = name
        self.abt = abteilung
        self.lohn = lohn # pro Monat
        Personal.L.append(self)

    def __str__(self):
        return '{0.name}/{0.abt}/{0.lohn}'.format(self)

    def lohnerhoehung(self, erhoehung):
        self.lohn += erhoehung

    def liste():
        return [self.name for self in Personal.L]

p1 = Personal('Lisa Meier', 'Marketing', 7000)
p2 = Personal('Sam Klein', 'Produktion', 6000)
p3 = Personal('Liz Hauser', 'Entwicklung', 8000)
p2.lohnerhoehung(500)

print(p2)
print(Personal.liste())
```

```
Sam Klein/Produktion/6500
['Lisa Meier', 'Sam Klein', 'Liz Hauser']
```

Aufgabe 4

```
class Image:

    def __init__(self):
        self.width = 4
        self.height = 3
        self.data = [['W', 'W', 'W', 'W'],
                     ['W', 'W', 'W', 'W'],
                     ['W', 'W', 'W', 'W']]

    def set_pixel(self, row, column, color):
        self.data[row][column] = color

    def show(self):
        for row in range(0, self.height):
            print(''.join(self.data[row]))

img = Image()
img.set_pixel(0, 0, 'B')
img.set_pixel(1, 1, 'B')
img.set_pixel(2, 2, 'B')
img.set_pixel(1, 3, 'B')
img.show()
```

```
BWWW
WBWB
WWBW
```

Aufgabe 5

```
class Quartierladen:

    def __init__(self):
        self.sortiment = {'brot': 0, 'orangen': 0}

    def neu(self, artikel, anzahl):
        self.sortiment[artikel] = anzahl

    def einkauf(self, artikel, anzahl):
        self.sortiment[artikel] += anzahl

    def verkauf(self, artikel, anzahl):
        ...

    def inventar(self):
        for art in self.sortiment:
            print('{0} {1}'.format(art, self.sortiment[art]))

q = Quartierladen()
q.einkauf('brot', 10)
q.neu('milch', 7)
q.einkauf('brot', 2)
q.inventar()
```

```
brot 12
orangen 0
milch 7
```

Aufgabe 6

```
# (a) Client Code
inf = Schulfach('Informatik')
inf.neue_note(3)
inf.neue_note(4)
inf.neue_note(5)
print(inf.get_note(1))
print(inf.durchschnitt())
```

```
class Schulfach:

    def __init__(self, name):
        '''Objekt mit name und leerer Notenliste'''
        self.name = name # (b)
        self.L = []      # (b)

    def neue_note(self, note):
        '''Fügt eine neue Note der Notenliste hinzu'''
        self.L.append(note) # (b)

    def get_note(self, i):
        '''Gibt die i-te Note zurück'''
        return self.L[i] # (b)

    def durchschnitt(self):
        '''Gibt den Notendurchschnitt zurück'''
        return sum(self.L)/len(self.L) # (b)
```

Aufgabe 7

```
from math import sqrt
class Dreieck:

    def __init__(self, a, b, c):
        '''Erzeugt ein Dreieck-Objekt'''
        self.a, self.b, self.c = a, b, c # Platzgründe!

    def __str__(self):
        '''Gibt eine Textdarstellung der Seiten zurück.'''
        return 'a={0.a}, b={0.b}, c={0.c}'.format(self)

    def umfang(self):
        '''Gibt den Umfang des Dreiecks zurück.'''
        return a + b + c

    def inhalt(self):
        '''Gibt den Flächeninhalt des Dreiecks zurück.'''
        s = (a + b + c)/2
        return sqrt(s*(s-a)*(s-b)*(s-c))
```

Aufgabe 8

Die objektorientierte Programmierung ist ein Modell, das sich beim Schreiben eines Programms an der Realität des zu lösenden Problems ausrichtet. Die Objekte sind die Akteure, welche Aufgaben erledigen, ihren Zustand mitteilen/ändern und mit anderen Objekten kommunizieren können.

Solche Programme (1) sind besser verständlich/lesbar (2) sind leichter zu pflegen/warten (3) sind wiedervendbar und (4) eignene sich für die Programmierung im Team.

Aufgabe 9

(a) *Klasse:* Bauplan für Objekte eines bestimmten Typs

(b) *Instanz:* Ein zur Laufzeit erzeugtes Objekt einer Klasse.

(c) *Objektvariable (Objekteigenschaft):* Variable, die an ein Objekt gebunden ist.

(d) *Klassenvariable (Klasseneigenschaft):* Variable, die an eine Klasse gebunden ist.

(e) *Objektmethode:* Funktion, die an ein Objekt gebunden ist.

(f) *Klassenmethode:* Funktion, die an eine Klasse gebunden ist.

(g) *Konstruktor:* Methode (Funktion), die beim Erzeugen eines Objekts aufgerufen wird.

(h) *Kapselung:* Das Prinzip, die Implementierung (also die Realisierung) einer Klasse nach aussen hin zu verbergen und den Zugriff über dafür vorgesehene Methoden (Schnittstelle) zu steuern.