

**Aufgabe 1**

```
1 class Example:
2
3     a = 3
4
5     def xyz(s, t):
6         return s * t
7
8     def __init__(self, u, v):
9         self.u = u
10        self.v = v
11
12    def abc(self):
13        return (self.u + self.v)
14
15 obj = Example(2, 5)
16 print(obj.abc())
17 print(Example.a)
18 print(Example.xyz(4, 9))
```

- (a) Welche Ausgabe macht das Programm in Zeile 16?
- (b) Welche Ausgabe macht das Programm in Zeile 17?
- (c) Welche Ausgabe macht das Programm in Zeile 18?
- (d) In welchen Zeilen werden Instanzvariablen definiert?
- (e) In welchen Zeilen werden Instanzmethoden definiert?
- (f) In welchen Zeilen werden Klassenvariablen definiert?
- (g) In welchen Zeilen werden Klassenmethoden definiert?
- (h) In welcher Zeilen wird der Konstruktor definiert?

**Aufgabe 2**

Welche Ausgabe macht das folgende Python-Modul?

```
1 class Example:
2
3     c = 3
4
5     def __init__(self, b):
6         self.a = Example.c + b
7
8 x = Example(5)
9 print(x.a)
```

### Aufgabe 3

Was gibt das Python-Modul in den Zeilen 17–19 aus?

```
1 class Quadrat():
2
3     n = 0
4
5     def __init__(self, a):
6         self.a = a
7         Quadrat.n += 1
8
9     def flaeche(self):
10        return self.a * self.a
11
12    def umfang(self):
13        return 4 * self.a
14
15 a = Quadrat(5)
16 b = Quadrat(4)
17 print(a.umfang())
18 print(b.flaeche())
19 print(Quadrat.n)
```

### Aufgabe 4

Welche Ausgabe macht das folgende Programm?

```
1 class Parent:
2     def __init__(self, a):
3         self.a = a
4     def methodA(self, x):
5         return (self.a + x + 1)
6     def methodB(self, x):
7         return (self.a + x + 2)
8
9 class Child(Parent):
10    def __init__(self, a):
11        super().__init__(a)
12    def methodA(self, x):
13        return (self.a + x + 3)
14
15 x = Parent(1)
16 y = Child(2)
17 print(x.methodA(3))
18 print(x.methodB(8))
19 print(y.methodA(2))
20 print(y.methodB(5))
```

### Aufgabe 5

Implementiere gemäss dem folgenden Klassendiagramm eine Klasse für die Berechnung von Volumen und Oberfläche von Quader-Objekten.

Quader
a: number b: number c: number
Quader(a: number, b: number, c: number) volumen(): number oberflaeche(): number

### Aufgabe 6

Was ist eine Klasse?

### Aufgabe 7

Was ist eine Instanz?

### Aufgabe 8

Was ist (a) eine Instanzmethode und (b) eine Klassenmethode?

### Aufgabe 9

Was ist ein Konstruktor?

### Aufgabe 10

Was ist der Unterschied zwischen einer Instanzvariablen und einer Klassenvariablen?

### Aufgabe 11

Welche Semantik hat die Python-Funktion `super()`?

## Aufgabe 12

Die Klasse `Vector` implementiert einen Datentyp für die Komponentendarstellung von Vektoren im dreidimensionalen Raum.

- (a) Welche Ausgaben macht der Python-Code in den den Zeilen 31–33?
- (b) Vervollständige die Spezialmethode `__sub__(self, other)`, welche die Differenz der beiden Vektor-Objekte zurückgibt.
- (c) Vervollständige die Spezialmethode `__abs__(self)`, welche den Betrag, d.h. die Länge eines Vektor-Objekts zurückgibt.
- (d) Vervollständige die Methode `dot(self, other)`, die das Skalarprodukt der beiden Vektoren berechnet und zurückgibt. *Zur Erinnerung:*  $\vec{a} \cdot \vec{b} = x_a \cdot x_b + y_a \cdot y_b + z_a \cdot z_b$ .
- (e) Teste die implementierten Methoden aus (b)–(d) mit geeigneten Beispielen.

```
1 class Vector:
2
3     def __init__(self, x, y, z):
4         self.x = x
5         self.y = y
6         self.z = z
7
8     def __str__(self):
9         return '{0},{1},{2}^T'.format(self.x, self.y, self.z)
10
11    def __add__(self, other):
12        x = self.x + other.x
13        y = self.y + other.y
14        z = self.z + other.z
15        return Vector(x, y, z)
16
17    def __rmul__(self, a):
18        return Vector(a*self.x, a*self.y, a*self.z)
19
20    def __sub__(self, other):
21        ...
22
23    def __len__(self):
24        ...
25
26    def dot(self, other):
27        ...
28
29 a = Vector(2,3,1)
30 b = Vector(3,6,-5)
31 print(a)
32 print(a+b)
33 print(10*b)
```