Aufgabe 1

(a) Vervollständige die Methoden mit den Auslassungspunkten (...).

(b) Bestimme mit dem bestehenden Codes und den Docstrings (’’’...’’’) die Ausgaben der `print(...)`-Anweisungen und notiere sie rechts der Fragezeichen.

(c) Kennzeichne die *Klassenvariablen*, *Objektvariablen*, *Klassenmethoden* und *Objektmethoden*, sofern es solche gibt.

(d) In welchen Zeilen steht der Konstruktur und wo werden Instanzen erzeugt?

```
1  class Rechteck:
2
3      anzahl = 0
4
5      def __init__(self, breite, hoehe):
6          '''Erzeugt ein Rechteck mit <breite> und <hoehe>'''
7          self.b = breite
8          self.h = hoehe
9          Rechteck.anzahl += 1
10
11     def umfang(self):
12         '''Gibt den Umfang des Recktecks zurück'''
13         ...
14
15
16
17     def inhalt(self):
18         '''Gibt den Flächeninhalt des Recktecks zurück'''
19         ...
20
21
22
23     def print_anzahl():
24         '''Gibt die Anzahl erzeugter Rechtecke zurück'''
25         ...
26
27
28
29 r1 = Rechteck(5, 8)
30 r2 = Rechteck(2, 9)
31 r3 = Rechteck(1, 10)
32 print(r1.umfang())  # => ?
33 print(r2.inhalt())  # => ?
34 print(Rechteck.print_anzahl()) # => ?
```

Aufgabe 2

(a) Vervollständige die Methoden mit den Auslassungspunkten (...).

(b) Bestimme mit dem bestehenden Codes und den Docstrings (`'''...'''`) die Ausgaben der `print(...)`-Anweisungen und notiere sie rechts der Fragezeichen.

(c) Kennzeichne die *Klassenvariablen*, *Objektvariablen*, *Klassenmethoden* und *Objektmethoden*, sofern es solche gibt.

(d) In welchen Zeilen steht der Konstruktur und wo werden Instanzen erzeugt?

```
1  from math import sqrt
2
3  class Vektor:
4      '''Klasse für Vektoren im Raum'''
5
6      def __init__(self, x, y, z):
7          '''Erzeugt einen Vektor mit den Komponenten x, y, z'''
8          self.x = x
9          self.y = y
10         self.z = z
11
12     def __str__(self):
13         '''Gibt eine Textdarstellung des Vektors zurück'''
14         return f'({self.x}, {self.y}, {self.z})^T'
15
16     def __abs__(self):
17         '''Gibt den Betrag (Länge) des Vektors zurück'''
18         ...
19
20
21
22     def __add__(self, other):
23         '''Gibt die Summe der Vektoren als Vektor zurück'''
24         ...
25
26
27
28     def dot(self, other):
29         '''Gibt das Skalarprodukt der Vektoren zurück'''
30         ...
31
32
33
34 u = Vektor(1, 2, -2)
35 v = Vektor(3, 0, 4)
36 print(u+v)       # ?
37 print(abs(u))    # ?
38 print(u.dot(v))  # ?
```