

Aufgabe 1

Welche Ausgaben macht das folgende Programm?

```
1 class Point:
2
3     n = 0
4
5     def __init__(self, x, y):
6         self.x = x
7         self.y = y
8         Point.n += 1
9
10    def translate(self, x, y):
11        self.x += x
12        self.y += y
13
14    def reflection(self):
15        self.x = -self.x
16        self.y = -self.y
17
18    def __str__(self):
19        return '{0},{1}'.format(self.x, self.y)
20
21 A = Point(2, 5)
22 B = Point(-4, 3)
23 A.translate(10, 20)
24 B.reflection()
25 print(A)
26 print(B)
27 print(Point.n)
```

Aufgabe 2

Welche Ausgabe macht das folgende Python-Modul?

```
1 class Example:
2
3     c = 3
4
5     def __init__(self, x):
6         self.a = x
7         Example.c += x
8
9 e1 = Example(5)
10 e2 = Example(7)
11 print(e1.a)
12 print(e2.a)
13 print(Example.c)
```

Aufgabe 3

Welche Ausgabe macht das folgende Python-Modul?

```
1 class Dog():
2
3     species = 'Canis familiaris'
4
5     def __init__(self, name, age):
6         self.name = name
7         self.age = age
8
9     def speak(self, sound):
10        print('{0} says {1}'.format(self.name, sound))
11
12
13 d1 = Dog('Miles', 5)
14 d2 = Dog('Jack', 4)
15 print(d1.age)
16 print(d2.name)
17 d1.speak('Woof')
18 d2.speak('Bow Bow')
```

Aufgabe 4

Welche Ausgabe macht das folgende Python-Modul?

```
1 class Kreis:
2
3     pi = 3.14
4
5     def __init__(self, radius):
6         self.r = radius
7
8     def inhalt(self):
9         return self.r**2 * Kreis.pi
10
11    def umfang(self):
12        return 2 * self.r * Kreis.pi
13
14 k1 = Kreis(10)
15 k2 = Kreis(1)
16 print(k1.inhalt())
17 print(k2.umfang())
```

Aufgabe 5

Welche Ausgabe macht das folgende Python-Modul?

```
1  class Robot:
2
3      def __init__(self, name, x=0, y=0):
4          self.name = name
5          self.x = x
6          self.y = y
7          self.energy = 100
8
9      def go_north(self, steps):
10         self.y += steps
11         self.energy -= steps
12
13     def go_south(self, steps):
14         self.y -= steps
15         self.energy -= steps
16
17     def go_east(self, steps):
18         self.x += steps
19         self.energy -= steps
20
21     def go_west(self, steps):
22         self.x -= steps
23         self.energy -= steps
24
25     def __str__(self):
26         txt = '{0}\n'.format(self.name)
27         txt += 'Position: ({0},{1})\n'.format(self.x, self.y)
28         txt += 'Energy: {0}\n'.format(self.energy)
29         return txt
30
31 r1 = Robot('R2-D2', 3, 5)
32 r2 = Robot('Asimov')
33 r1.go_south(4)
34 r1.go_west(10)
35 r1.go_north(12)
36 r2.go_east(7)
37 r2.go_north(3)
38 r2.go_west(5)
39
40 print(r1)
41 print(r2)
```

Aufgabe 6

Schreibe gemäss der folgenden Vorlage eine Klasse mit dem Namen `Rechteck` für Rechtecksberechnungen, so dass der Testcode ganz unten in der Vorlage die angegebenen Ausgaben macht.

```
1 class Rechteck:
2     '''Klasse für Rechtecksberechnungen'''
3
4     def __init__(self, a, b):
5         ...
6
7     def __str__(self):
8         ...
9
10    def inhalt(self):
11        ...
12
13    def umfang(self):
14        ...
15
16 if __name__ == '__main__':
17
18     r1 = Rechteck(3, 2)
19     r2 = Rechteck(1, 5)
20
21     print(r1)          # sollte 'a=3, b=2' ausgeben
22     print(r1.inhalt()) # sollte 6 ausgeben
23     print(r2.umfang()) # sollte 12 ausgeben
```