Aufgabe 1

```
class Point:

    n = 0

    def __init__(self, x, y):
        self.x = x
        self.y = y
        Point.n += 1

    def translate(self, x, y):
        self.x += x
        self.y += y

    def reflection(self):
        self.x = -self.x
        self.y = -self.y

    def __str__(self):
        return '({0},{1})'.format(self.x, self.y)

A = Point(2, 5)
B = Point(-4, 3)
A.translate(10, 20)
B.reflection()
print(A)
print(B)
print(Point.n)
```

```
(12,25)
(4,-3)
2
```

Aufgabe 2

```
class Example:

    c = 3

    def __init__(self, x):
        self.a = x
        Example.c += x

e1 = Example(5)
e2 = Example(7)
print(e1.a)
print(e2.a)
print(Example.c)
```

```
5
7
15
```

Aufgabe 3

```
class Dog():

    species = 'Canis familiaris'

    def __init__(self, name, age):
        self.name = name
        self.age = age

    def speak(self, sound):
        print('{0} says {1}'.format(self.name, sound))


d1 = Dog('Miles', 5)
d2 = Dog('Jack', 4)
print(d1.age)
print(d2.name)
d1.speak('Woof')
d2.speak('Bow Bow')
```

```
5
Jack
Miles says Woof
Jack says Bow Bow
```

Aufgabe 4

```
class Kreis:

    pi = 3.14

    def __init__(self, radius):
        self.r = radius

    def inhalt(self):
        return self.r**2 * Kreis.pi

    def umfang(self):
        return 2 * self.r * Kreis.pi

k1 = Kreis(10)
k2 = Kreis(1)
print(k1.inhalt())
print(k2.umfang())
```

```
314.0
6.28
```

Aufgabe 5

```
class Robot:

    def __init__(self, name, x=0, y=0):
        self.name = name
        self.x = x
        self.y = y
        self.energy = 100

    def go_north(self, steps):
        self.y += steps
        self.energy -= steps

    def go_south(self, steps):
        self.y -= steps
        self.energy -= steps

    def go_east(self, steps):
        self.x += steps
        self.energy -= steps

    def go_west(self, steps):
        self.x -= steps
        self.energy -= steps

    def __str__(self):
        txt  = '{0}\n'.format(self.name)
        txt += 'Position: ({0},{1})\n'.format(self.x, self.y)
        txt += 'Energy: {0}\n'.format(self.energy)
        return txt

r1 = Robot('R2-D2', 3, 5)
r2 = Robot('Asimov')
r1.go_south(4)
r1.go_west(10)
r1.go_north(12)
r2.go_east(7)
r2.go_north(3)
r2.go_west(5)

print(r1)
print(r2)
```

```
R2-D2
Position: (-7,13)
Energy: 74

Asimov
Position: (2,3)
Energy: 85
```

Aufgabe 6

```
class Rechteck:
    '''Klasse für Rechtecksberechnungen'''

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

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

    def inhalt(self):
        return self.a * self.b

    def umfang(self):
        return 2*(self.a + self.b)

if __name__ == '__main__':

    r1 = Rechteck(3, 2)
    r2 = Rechteck(1, 5)

    print(r1)
    print(r1.inhalt())
    print(r2.umfang())
```