Informatics 2, 2$^{\text{nd}}$ midterm (2020-04-28)

Submit the solutions to `hazi@math.bme.hu`*, deadline: 19:15 CEST*

1. Write a `Shape` class that represents shapes on the plane. And also two other classes inherited from `Shape`: `Line` and `Circle`. *(8 points)*

The constructor of the `Line` class should have three parameters: the x and y coordinates of one of its points and its slope. The slope can be ∞ if the line is vertical.

The constructor of the `Circle` class should have three parameters: the x and y coordinated of its center and its radius. Raise a `ValueError` exception if the radius is not positive.

Both shapes should have a `move` method, that translates the object with a given planar vector. Return `None`, but change the place of the object. The parameters should be the x and y coordinates of the translation vector.

Both shapes should have a `__str__` method which returns the string representation in the form of an equation describing the shape. It can be any bi-variate equation for which the solutions form the desired shape. Try something like this for the line.

Example:

```
>>> l = Line(0, 0, 1)
>>> print(l)
y-0=1*(x-0)
>>> x = Circle(0, 0, 1)
>>> x.move(1, 0)
>>> print(x)
(x-1)^2+(y-0)^2=1
```

You have to mind the infinite slope for the line.

Hint:

```
class Shape:
    pass

class Line(Shape):
    def __init__(self, x, y, m):
        pass

class Circle(Shape):
    def __init__(self, x, y, r):
        pass
```

2. Theoretical questions *(4 points)*

1. Show an example of inheritance and also name the classes according to their role in the inheritance (kinship).
2. How to make the objects of your class printable? Which method should you write and how?
3. What is the difference between the member functions and data members? Illustrate with an example or an explanation.
4. Show an example usage of the `str.format` method!

3. Write an iterable class called `Collatz2` that can iterate over the Collatz sequence starting from a given positive integer until it reaches 1, but retrieve the **square** of each element along the way. The iteration should take at least one step, because the last element in the sequence is always 1. Its constructor should have one parameter (except self): `x`. Check whether `x` is an integer and positive, if not then raise `ValueError` exception. *(4 points)*

Note that the Collatz sequence is as follows:

$$x_{n+1} = \begin{cases} \frac{x_n}{2} & \text{if } x_n \text{ is even} \\ 3x_n + 1 & \text{if } x_n \text{ is odd} \end{cases} \tag{1}$$

The class should work like in this example:

```
for i in Collatz(3):
    print(i, end=" ")
```

Output:

```
9 100 25 256 64 16 4 1
```

Because the original sequance is would be this:

```
3 10 5 16 8 4 2 1
```

Note that the square function is a bijection on the positive numbers with 1 as its unique fixed point, therefore the squared sequence can be defined similarly to (1). But you can do it much simpler using the solution of the original Collatz exercise!

4. There are 4 mistakes in the following code. Find those and fix it. *(4 points)*
There is no more than 1 error in each line.

```
class A:
    x = 3
    def __init__(self):
        self.x = []
    def __str__(self):
        return self.x * A.x

class B:
    def add(self, x)
        self.x.append(x)

b = B(3)
b.add(3)
print(b)
```