Informatics 2, 2nd midterm (2020-04-14)

Submit the solutions to `hazi@math.bme.hu`*, deadline: 19:00 CEST*

- You won't know the results until the deadline is over
- Latest submission counts, not the best
- Your points will be proportional to the number of correct test cases

1. Write a `Piece` class that represents a chess piece on a chessboard. *(6 points)*

Its constructor should receive 2 arguments (except `self`): the position of the piece and its color (`"black"` or `"white"`) as strings. Store these data into self. The position will match the regex `\w\d`. It should work with both upper- and lowercase algebraic notation.

Implement a `move` method that repositions the piece to a new place. The moves of the specific figures don't matter, but the piece cannot move outside of the chessboard. If either the constructor or the move method is called with a position that's outside of the chessboard or the position is invalid, raise your own `PieceMoveError` exception (you have to write a custom exception class). The `move` method should return `None` but modify the position or raise an exception.

Also implement a `__str__` method which returns the position and the color as a string in one word. Example:

```
>>> p = Piece('A3', 'black')
>>> print(p)
A3black
>>> p.move('a0')
PieceMoveError: invalid position
>>> p.move('a1')
>>> print(p)
A1black
```

2. Multiple choice question *(4 points)*

1. In case of inheritance, how do we call the class in parenthesis?
   `class Pawn(Piece): pass`

   A child
   B parent
   C descendant
   D sibling

2. Which is the correct signature of overriding the `*` operator?

   A `def mul(self, other):`
   B `def __mul__(self):`
   C `def mul(self):`
   D `def __mul__(self, other):`

3. What will be printed by the following: `print("{:{}^{a}}".format("|", ".", a=3))`

   A `.a.`
   B `...`
   C `.|.`
   D `|3|`

4. Which one of these is an *invalid* class definition?

   A `class class: pass`
   B `class Class: pass`
   C `class Class(): pass`
   D `class Class(object): pass`

3. Write an iterable class called `Collatz` that can iterate over the Collatz sequence starting from a given positive integer until it reaches 1. 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. *(6 points)*
Note that the Collatz sequence is as follows:

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

The class should work like in this example:

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

Output:

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

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.

```
def A:
    def __init__(self, x):
        self.x = x
    def __str__(self):
        return str(self.x)
class B():
    def __init__(self, *x)
        super.__init__(x[0])
    def f(self, other):
        return B(self.x + other.x)
```

The following code should work after the corrections:

```
a = A(3)
b = B(3, 4)
print(a, b, b.f(a))
```