Informatika 2, 2. ZH (2020-04-14)

A megoldásokat a **hazi@math.bme.hu** *címre küldjétek el 19:00-ig!*

- Az eredményeket nem fogjátok látni a határidő lejártáig
- Az utolsó beküldés számít, nem a legjobb
- A megszerzett pontok a helyes tesztesetek számával lesznek arányosak

1. Írjunk egy `Piece` nevű osztályt, ami sakkfigurákat reprezentál a sakktáblán. *(6 pont)*

A konstruktorának legyen 2 paramétere (a `self`-en kívül): a pozíció és a figura színe (`"black"` vagy `"white"`) string-ként megadva. Tároljuk el ezeket az adatokat a self-be. A pozíció az alábbi regex-re fog illeszkedni: `\w\d`. Kis és nagybetűs algebrai jelöléssel is működnie kell.

Implementáljunk egy `move` metódust, ami áthelyez egy bábut. A bábuk lépéseinek szabályait nem kell figyelembe venni, de az új pozíciónak a táblán kell lennie. Ha helytelen mezőt adunk meg, vagy a táblán kívülre akarjuk helyezni a bábut, mind a konstruktorban mind a `move` metódusban, akkor emeljünk saját `PieceMoveError` kivételt (meg kell írnunk a saját kivétel osztályunkat). A `move` metódus `None`-t adjon vissza, de változtassa meg az eltárolt pozíciót vagy emeljen kivételt.

A `__str__` metódust is implementáljuk, ami a helyet és a színt egy string-ként adja vissza.
Példa:

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

2. Felelet választás *(4 pont)*

1. Az öröklődésnél hogyan hívjuk a zárójelben lévő osztályt?
   `class Pawn(Piece): pass`

   A gyerek

   B ős

   C leszármazott

   D testvér

2. Mi a `*` operátor felüldefiniálásának helyes módja?

   A `def mul(self, other):`

   B `def __mul__(self):`

   C `def mul(self):`

   D `def __mul__(self, other):`

3. Mit ír ki az alábbi parancs: `print("{:{}^{a}}".format("|", ".", a=3))`

   A `.a.`

   B `...`

   C `.|.`

   D `|3|`

4. Melyik *helytelen* osztály-definíció?

   A `class class: pass`

   B `class Class: pass`

   C `class Class(): pass`

   D `class Class(object): pass`

3. Írjunk egy `Collatz` nevű iterálható osztályt, amivel egy pozitív egészből indított Collatz sorozaton lehet iterálni, amíg el nem érjük az 1-et. Az iterálás sosem lehet üres, mert az utolsó elem mindig 1. A konstruktorának egy paramétere legyen (a self-en kívül): `x`. Ellenőrizzük, hogy `x` egész és pozitív, ha nem az akkor emeljünk `ValueError` kivételt. *(6 pont)*
elmékezzünk, hogy a Collatz sorozat képzési szabálya a következő:

$$x_{n+1} = \begin{cases} \frac{x_n}{2} & \text{ha } x_n \text{ páros} \\ 3x_n + 1 & \text{ha } x_n \text{ páratlan} \end{cases}$$

Az osztálynak így kell működnie:

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

Kimenet:

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

4. Az alábbi kódban 4 hiba található, javítsuk ki ezeket! *(4 pont)*
Mindegyik sorban legfeljebb egy hiba van.

```
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)
```

A javítások után az alábbinak működnie kell:

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