

Tartalomjegyzék

- 1 Osztály
 - ♦ 1.1
Komplex
- 2 Iterálhatók
 - ♦ 2.1
Öröklődés
 - ◊ 2.1.1
Alakzatok

Osztály

Komplex

A feladat az előadáson elkezdett Komplex osztályt befejezni:

```
class Komplex(object):
    def __init__(self, real, imaginary):
        self.re = real
        self.im = imaginary

    def __add__(self, k2):
        uj_re = self.re + k2.re
        uj_im = self.im + k2.im
        return Komplex(uj_re, uj_im)

    def __str__(self):
        s = ""
        s += str(self.re)
        s += " + "
        s += str(self.im)
        s += "i"
        return s

k1 = Komplex(4, 3)
k2 = Komplex(-2, 1)
k3 = k1 + k2

print(k3)
```

- Az előző órán megvalósítottuk a kivonás és a szorzás műveleteket, most írjuk meg osztás műveletet is. (`__truediv__`) Az osztás előtt érdemes lehet a következő részt megoldani először.
- Valósítsuk meg a `norm` módszert, mely a komplex szám hosszát adja meg.
- Javítsuk ki a `__str__` módszert, hogy szépen írja ki a számokat, pl:

```
2 - 4i
5i
2
```

Teszteléshez használhatjuk pl ezt a kódot, de írjunk saját teszteket is!

```
k1 = Komplex(4, 3)
k2 = Komplex(-2, 1)
k3 = Komplex(4, 1)

print k1 + k2
print k1 - k3
```

```
print k2 * k1
print k3 / k1
print k1.norm()
```

Iterálhatók

Írjunk olyan iterálható osztályt, mint a **range**, de ne egy listát járjon be, hanem csak az aktuális elemet tárolja.

```
class Range:
    def __init__( ... ):
        ...
    def __iter__( ... ):
        ...
    def __next__( ... ):
        ?
```

- konstruktora egy számot vagy sztringet kapjon. Addig a számig lehessen iterálni rajta, nullától, egyesével.
- Ha a szám nem pozitív, akkor 0 hosszan lehessen iterálni rajta.
- Ha sztringet kap a konstruktor és az nem értelmezhető egészként, akkor emeljünk **ValueError** kivételt.
 - ♦ Ha értelmezhető egészként, akkor alakítsuk át egészszé és számoljunk azzal.
- Ha "<http://wiki.math.bme.huinf>"<http://wiki.math.bme.hu> sztringet kap a konstruktor, akkor végtelen sokáig lehessen rajta iterálni!

Örökl?dés

Alakzatok

Írjunk egy **Shape** osztályt.

- Legyen **x** és **y** változója, ezek tárolják az alakzat pozícióját a síkon.
- Legyen egy **move** metódusa, aminek egyetlen **v** paramétere van, egy kételemű lista, a vektor, amivel el kell mozgatni az alakzatot.

Definiáljuk a **Shape** osztály leszármazottaiként az

- **Ellipse** ellipszis, legyen meg a kis- és nagytengelye (**a,b**)
- **Rectangle** téglalap, legyen meg az oldalak hossza (**a,b**)

osztályokat. Mindkét esetben a pozíciójuk a súlypontjukat jelentse.

Írjunk mindkét osztályhoz egy **area** függvényt, ami kiszámítja az alakzat területét!

Definiáljuk az **Ellipse** osztály **equation** metódusát, ami kiírja az adott ellipszis egyenletét!