

Tartalomjegyzék

- 1 Előadás
- 2 Feladatok
 - ◆ 2.1 Bevezető
 - ◇ 2.1.1
Komplex
 - ◆ 2.2 CloudCoder
 - ◇ 2.2.1
CloudCoder
használata
 - ◇ 2.2.2
Feladatok
ajánlott
sorrendje

Előadás

7. előadás

Feladatok

Bevezető

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 __repr__(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
```

- Valósítsuk meg a kivonás, szorzás és osztás műveleteket. (`__sub__`, `__mul__`, `__div__`) Az osztás előtt érdemes lehet a következő részt megoldani először.
- Valósítsuk meg a `__len__` metódust, mely a `len` függvény által adott eredményt szabályozza. Legyen

ez a komplex szám hossza.

- Javítsuk ki a `__repr__` metódust, 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 len(k1)
```

CloudCoder

CloudCoder használata

A legtöbb python feladathoz gyakorlaton egy CloudCodernek nevezett rendszert fogunk használni. Ennek előnye, hogy helyben ki is javítja a feladatot. Elérés:

- <https://ccweb.math.bme.hu/cloudcoder/>

Részletes leírás a tárgylapon.

Feladatok ajánlott sorrendje

1. foglalas_1
2. foglalas_2
3. foglalas_3
4. foglalas_4
5. foglalas_5
6. kinyer_metodus