

Feladatok

A feladatok megoldásához szükség lesz az el?adáson használt kódra!

Bevezet?

Másoljuk be a bináris fa definícióját. Ezen az osztályon fogunk dolgozni.

- Írjunk egy `count(self)` metódust, ami megszámolja a fa elemeinek a számát!
- Írjunk egy `sum(self)` metódust, ami a fa összes csúcsában lévő értékeket összegezi!
- Írjunk egy `height(self)` metódust, ami megmondja, hogy milyen magas a fa!
- Írjunk egy `is_list(self)` metódust, ami megmondja, hogy a bináris fa listává fajult-e. Egy bináris fa akkor fajult listává, ha minden csúcsának legfeljebb egy gyereke van.

Számológép

Az el?adáson mutatott számológép-kezdeményt fejlesztjük tovább. Kezdjük új fájlba. El?ször hozzunk létre egy üres Node osztályt!

- Írjuk meg a konstruktort a következ?képp. A bemenet egy sztring, ezt kell értelmeznünk. Egyelőre a bemeneti sztring nem tartalmazhat zárójelet, sem negatív számokat.
 - ◆ Ha a sztring egy szám, akkor egyszer?en eltároljuk a `self.data` változóban mint számot.
 - ◆ Ha nem szám (azaz a négy alapművelet egyike szerepel a sztringben), akkor szétvágjuk a sztringet két részre a legalacsonyabb precedenciájú művelet mentén.

Mivel zárójeleket nem engedünk meg, a legalacsonyabb szintű művelet az első + vagy - jel, ha nincs ilyen, akkor pedig az első * vagy /. A `self.data` értéke a műveleti jel lesz mint karakter, a bal, illetve a jobb gyereke pedig a sztring két széthasított része alapján épüljön fel!

- Számoljuk ki egy ilyen műveletsor értékét!
- Szeretnénk hatványozni is, építsük bele ezt a ^ jellel!
- Írjunk hozzá `__str__` metódust, ami kiírja fát.
 - ◆ Ez volt el?adáson, most metódusként írjuk meg.