<-- vissza

Síkidom osztálystruktúra

Beadási határid?: 2012. máj. 7.

Készíts osztályokat a következ? struktúra szerint:

```
Pont
  |
  * - Negyzet
        |
        * - Teglalap
        |
        * - Kor
Vektor
```

A **Pont** osztály definiálja hogy egy koordináta rendszerben hol helyezkedik el (x,y) Legyen neki két függvénye, amik konstans nullával térnek vissza.:

- terulet
- kerulet

Konstruktorában az x és az y legyen paraméter, és ha nem adják meg, akkor alapból a 0,0 legyen hozzájuk rendelve.

A pontot az + (__add__) operátorral el lehessen tolni, ha egy vektort kap.

A **Vektor** osztály két pontot tartalmaz, amit a konstruktorban inicializálunk, ha nem adnak meg koordinátákat akkor alapból a 0,0 és az 1,0 értéket kapja. A konstruktor Pontokat hozzon létre és ezt tárolja a Vektor osztályba (ezt nevezzük tartalmazásnak)

Definiáljuk továbbá a **+** (__add__), **-** (__sub__), ***** (__mul__) operátorokat is. Az operátorok minden esetben egy új vektorral térjenek vissza.

A **Negyzet** osztály a Pont leszármazottja, a konstruktora használja az ?se konstruktorát is a super() függvény segítségével. És értelem szer?en definiálja felül a kerület, terület függvényeket. Írd felül az __add__ operátort ha kell, hogy hasonlóan az eltolást valósítsa meg.

A **Teglalap** és a **Kor** ?se a Negyzet, ?k is használják fel ?sük konstruktorát a super() segítségével és írják felül értelemszer?en a kerület, terület függvényeket. Írd felül az __add__ operátort ha kell, hogy hasonlóan az eltolást valósítsa meg.

Ezután tegyétek ezeket egy modulba és írjatok egy fv. ami csak abban az esetben fut le ha nem importaljak a modult hanem futtatják. A fv. pedig rátok bízom, de demonstrálja a megírt osztályokat.

pl.:

- példányosítás
- vektorok összeadása, ...
- területek kerületek számítása, ...

Levonunk pontot ha nincs:

- modul
- import soran is lefut a demonstracio
- demonstráció nem tér ki mindenre
- nincsenek meg a kért osztályok
- nincsenek meg a kért fv-ek
- nincsenek meg a kért osztályok operátorai

super() függvény

a super fv egy típust vár paraméterként és visszaad egy "http://wiki.math.bme.huproxy"http://wiki.math.bme.hu-t amin keresztül elérjük az ?s függvényeit. A proxy-n az operátorok nem használhatók!

super(<type>) - type ?sének közvetít?jét adja vissza super(<type>, <obj>) - type ?sének közvetít?jét adja vissza ha az obj type típusú super(<type1>, <type2>) - az type1 type2 ?sének közvetít?jét adja vissza (többes öröklés esetén)

```
class A(object):
  def __init__(self, x):
    self.hey = x * x

class B(A):
  def __init__(self, x, y):
    self.zed = y / x
    super(B, self).__init__(x)

print B
a = A(5)
print a.hey
b = B(5, 30)
print b.zed, b.hey
```

super függvény leírása

<-- vissza