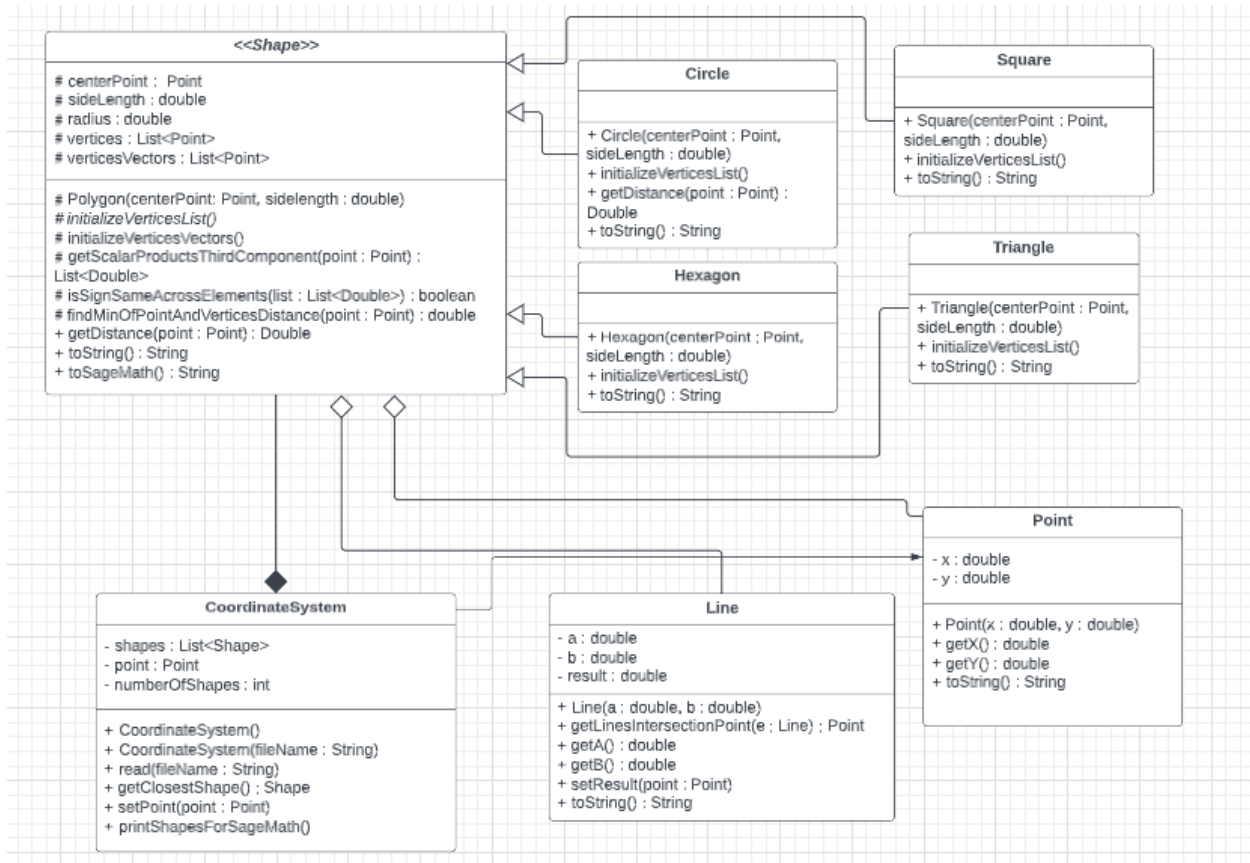


# Dokumentáció

## Feladat szövege

8. Rögzítsen a síkon egy pontot, és töltsön fel egy gyűjteményt különféle szabályos (kör, szabályos háromszög, négyzet, szabályos hatszög) síkidomokkal! Keresse meg, melyik síkidom van legközelebb a ponthoz! Kör esetén a közelséget a körvonaltól vett távolság adja meg, ha a pont a körön kívül van. Ha a pont a körön belül helyezkedik el, akkor a távolságukat nullának tekintjük. Minden síkidom reprezentálható a középpontjával és az oldalhosszal, illetve a sugárral, ha feltesszük, hogy a sokszögek esetében az egyik oldal párhuzamos a koordináta rendszer vízszintes tengelyével, és a többi csúcs ezen oldalra fektetett egyenes felett helyezkedik el. A síkidomokat szövegfájlból töltsse be! A fájl első sorában szerepeljen a síkidomok száma, majd az egyes síkidomok. Az első jel azonosítja a síkidom fajtáját, amit követnek a középpont koordinátái és a szükséges hosszúság. A feladatokban a beolvasáson kívül a síkidomokat egységesen kezelje, ennek érdekében a síkidomokat leíró osztályokat egy közös ősosztályból származtassa!

# UML - Class diagram



## Osztályleírás

**CoordinateSystem:** a koordináta rendszert összefogó és reprezentáló osztály, ide kerülnek feltöltésre a síkidomok, illetve minden a program élete során létrehozott elemet ezen értelmezünk

**Shape:** a síkidomot általánosan reprezentáló osztály, a specifikusan meghatározott síkidomok pl. kör ősszüelője

**Line:** egy vonal osztálybéli reprezentációja, számításhoz használt

**Point:** egy pont osztálybéli reprezentációja, a síkidomok csúcsának meghatározásán túl sok mindenhez használjuk, mint pl. a feladat lényege is egy ponthoz mért síkidom távolságok minimumának kiszámítása

**Circle:** a kör reprezentációja, ami egy síkidom típus, Shape leszármazottja

**Triangle:** a háromszög reprezentációja, ami egy síkidom típus, Shape leszármazottja

**Square:** a négyzet reprezentációja, ami egy síkidom típus, Shape leszármazottja

**Hexagon:** a hatszög reprezentációja, ami egy síkidom típus, Shape leszármazottja

## Tesztelési terv

**Fontos:** a tesztek rögzített (3,3) pontnál történtek

Teszteset leírása	Bemenet	Kimenet
1. Egy síkidom esetén a program ezt az egy síkidomot adja eredményként	1 C 1 1 1	Closest Shape: Circle( central point: ( x: 1.0, y: 1.0 ), side-length: 1.0, radius: 0.5 )
2. Üres fájl esetén a program Exception-t dob vissza	<üres>	File is empty!
3. Nem teljes fájl (pl. csak szám vagy nem teljes adatsor) esetén a program Exception-t dob	1 H 1 1	Input is invalid, could not read a full line!
4. Invalid síkidom jelölésnél is Exception-t dob a program	1 R 1 1 1	Illegal Shape type received!
5. Számok helyett betűkkel is Exceptiont kapunk	1 C b e t	Input is not a number!
6. Két egyező síkidom esetén a program a közelebbit adja vissza eredményül	2 C 0 0 1 C 0 0 2	Closest Shape: Circle( central point: ( x: 0.0, y: 0.0 ), side-length: 2.0, radius: 1.0 )
7. Két különböző síkidom esetén a program a közelebbit adja vissza eredményül	2 C 0 0 1 T 0 0 2	Closest Shape: Triangle( central point: ( x: 0.0, y: 0.0 ), side-length: 2.0 )
8. Három különböző síkidom esetén a program a közelebbit adja vissza eredményül	3 H 3 3 1 C 0 0 1 T 0 0 2	Closest Shape: Hexagon( central point: ( x: 3.0, y: 3.0 ), side-length: 1.0 )
9. Három síkidomnál, ha az egyik tartalmazza a másikat, akkor a másik alapvetően hiába van közelebb	3 C 3 3 10 C 5 5 1 T 5 5 1	Closest Shape: Circle( central point: ( x: 3.0, y: 3.0 ), side-length: 10.0, radius: 5.0 )