**Feladat**

*Rögzítsen a síkon egy pontot, és töltsön fel egy gyűjteményt különféle szabályos (kör, szabályos háromszög, négyzet, szabályos hatszög) síkidomokkal! Számolja meg, hogy a pontot hány síkidom tartalmazza! 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 töltse 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!*

pl.:

4  
circle 5 -6.2 3  
triangle -3 6 4  
square 8 6 1.5  
hexagon 5 5 2

Azaz, 4 síkidom van. Egy kör, egy háromszög, egy négyzet és egy hatszög.

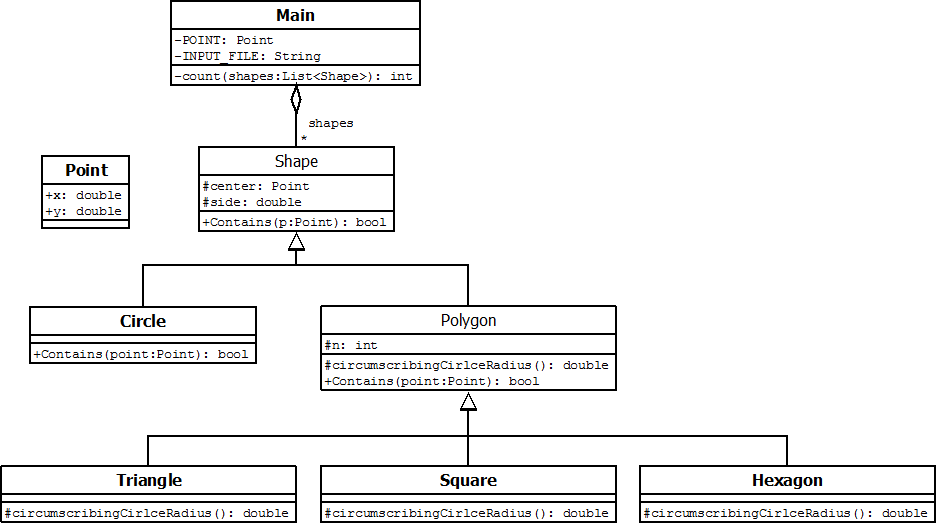
Sajátosságok (hogyan kell kiszámolni, hogy tartalmazza-e a pontot):

* circle – kör
  + középpont: K(u, v)
  + vizsgált pont: P(x ,y)

A sokszögek esetében ki kell számolni a köré írható kör sugarát, majd egy algoritmus kiszámolja a csúcsok koordinátáit, ezek alapján pedig megmondja, hogy benne van-e a pont az alakzatban.

* triangle – háromszög
  + oldalhossz: a
  + köré írható kör sugara:
* square – négyzet
  + oldalhossz: a
  + köré írható kör sugara:
* hexagon – hatszög
  + oldalhossz: a
  + köré írható kör sugara:

**Osztályszerkezet**

Az alábbi ábrán az entitás-osztályok kapcsolatait láthatjuk.

**Megvalósítás**

* A feladatot egy Java8 nyelven írt, objektumorientált szemléletet követő programmal oldottuk meg
* A program belépési pontja a Main osztályban van. Ez felel a síkidomok beolvasásáért és az eredmény kiírásáért. A felmerülő hibákat is itt kezeljük le azáltal, hogy kiírjuk, hogy hiba történt
* A Main osztály példányosít egy ShapeParser példányt, ami az alakzatok fájlból való beolvasását végzi. Az alábbi kivételeket dobhatja:
  + FileNotFoundException – helytelen fájlnév
  + NumberFormatException – ha a síkidomok száma vagy paramétereik nem értelmezhetők számként
  + NoSuchElementException – ha a fájl üres, ha egy nem felismert alakzatot tartalmaz, ha kevesebb alakzat van megadva, mint az első sorban lévő szám, valamint ha kevesebb vagy több paramétere van egy alakzatnak
  + *IllegalArgumentException –* ha az oldalhossz nem pozitív
  + IOException – a Scanner bezárásakor
  + *Megjegyzés: a FileNotFoundExceptiont külön nem tüntettem fel a szignatúrában, hiszen az egyben IOException is*
* A síkidomok típusa a Shape absztrakt osztály. Mindegyiknél le lehet kérdezni, hogy tartalmazza-e a rögzített pontot (Contains()). Ennek a kiszámításához szükség van a csúcspontok koordinátáira, amihez a köré írható kör sugarát kell kiszámolni, ezért két típusra vannak osztva (Circle és Polygon).
* A Circle a kör egyenletének segítségével megmondja, hogy tartalmazza-e a pontot. (Mivel körről van szó, itt nem kell köré írható kört számolni.)
* A Polygon a köré írható kör sugarának és a csúcsok számának segítségével mondja meg, hogy tartalmazza-e az adott sokszög (Triangle, Square vagy Hexagon) a pontot.
* Az egyes alakzatokat a Shapeből leszármazott típusokkal valósítottuk meg. A ShapeParser fogja a megfelelő típusokat példányosítani és egy Shape-listába.

**Tesztelés**

A programhoz az alábbi érvénytelen teszteseteket készítettük:

* Nem létező fájl [e00.txt] – kivételt dob
* Üres fájl [e01.txt] – kivételt dob
* Szükségesnél több vagy kevesebb alakzatot tartalmazó fájl [e02.txt, e03.txt] – kivételt dob
* Nem pozitív oldalhosszú alakzatot tartalmazó fájl [e04.txt] – kivételt dob
* Nem számként értelmezhető koordinátákat vagy oldalhosszot tartalmazó fájl [e05.txt, e06.txt] – kivételt dob
* Ismeretlen alakzattípust tartalmazó fájl [e07.txt] – kivételt dob
* Szükségesnél kevesebb vagy több paraméterű alakzatot tartalmazó fájl [e08.txt, e09.txt] – kivételt dob
* Első sorban alakzatot tartalmazó fájl, nincs alakzatszám megadva [e10.txt] – kivételt dob
* Első sorban számként nem értelmezhető adatot tartalmazó fájl megadva [e11.txt] – kivételt dob
* Egy soros, számot tartalmazó fájl e12.txt] – kivételt dob, hisz nincs alakzat

Érvényes tesztesetek:

* 4 különböző alakzat [c01.txt] – helyesen számol
* 2 ugyanolyan típusú alakzat [c02.txt] – helyesen számol
* 3 ugyanolyan típusú alakzat [c03.txt] – helyesen számol
* 4 ugyanolyan típusú alakzat [c04.txt] – helyesen számol