|  |  |  |
| --- | --- | --- |
| **Tasi Zoltán György** | **Progtech 1. beadandó/9. feladat** | 2021.október 19. |
| HM37UQ |
| [tasi.zoltan07@gmail.com](mailto:tasi.zoltan07@gmail.com) |
| 8. csoport |

# Feladat

Töltsön fel egy gyűjteményt különféle szabályos (kör, szabályos háromszög, négyzet, szabályos   
hatszög) síkidomokkal! Adja meg azt a síkidomot, amelynek a területe és a kerülete a legkisebb   
mértékben tér el egymástól! 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!

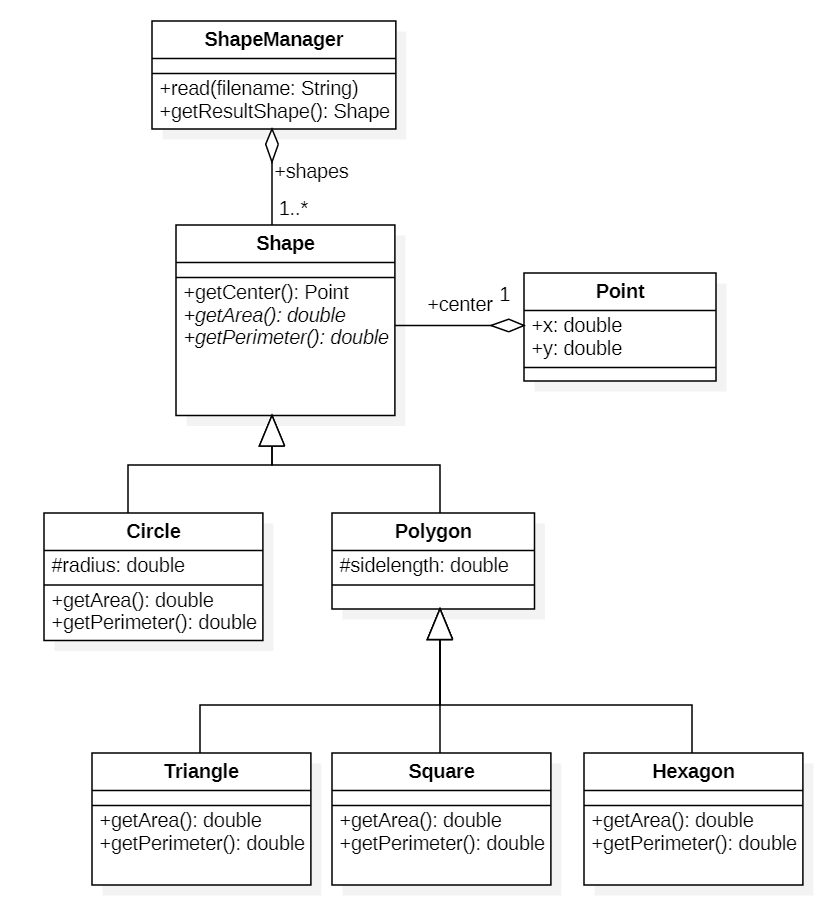
# Megoldási terv:

A megoldási tervhez különböző alakzatokat fogunk tárolni a ShapeManager osztályban, azon belül egy beépített ArrayList gyűjteményben. Ezekben általános Shape típusokat tárolunk, aminek meg van adva absztrakt getArea() és getPerimeter() metódusai. A négy fajta alakzatokat ebből a Shape osztályból származtatjuk, illetve a három poligonhoz még bevezetünk egy Polygon osztályt is, ami az oldalhosszúságot absztraktálja ki.

A ShapeManager osztályt a read() metódussal töltjük fel, ami egy filenevet kap. Ebből fogja beolvasni, létrehozni és eltárolni a különböző alakzatokat. Az alakzatok típusait így jelöljük: Kör – C, háromszög – T, négyzet – S, hatszög – H.

A feladat eredményét a ShapeManager getResultShape metódusa fogja visszaadni. Ez egy programozási tételt, a maximum kiválasztást fogja használni a gyűjteményen, hogy visszaadja a síkidomot aminek a területe és kerülete a legkisebb mértékben tér el. Ha a gyűjtemény üres, kivétel váltódik ki.

# Megoldás UML diagramja:



# Típusok

A feladat fő típusa a **Shape** absztrakt osztály, ami szülője minden használt alakzatnak. Ennek van egy Point típusú középpontja, és egy megvalósítatlan getArea és getPerimeter metódusa.

A **Point** osztálynak két adattagja van, a double typusú x és y koordináták.

Shape osztályból származik a **Circle** osztály, aminek van egy double típusú radius adattagja, és megvalósítja a szükséges metódusokat.

A Shape osztályból származik még a **Polygon** absztrakt osztály, aminek csak egy sideLength adattagot ad hozzá a Shape-hez.

Polygon-ból származik még a **Triangle**, **Square** és a **Hexagon**, ezek mind megvaósítják a getArea és getPerimeter metódusokat.

A **ShapeManager** osztály lesz ami kezeli a gyűjteményünket, beolvas, és megoldja a feladatot a Megoldási tervben leírtak szerint.

# Tesztelési terv

|  |  |  |
| --- | --- | --- |
| Általános teszt több alakzatra  (test1.txt) | 4  H 1 2 3  S 0 0 3  T -1 -4 3  C 0,5 -0,5 3 | Négyzet, aminek (0,0) a középpontja és 3 hosszú oldala van |
| Üres fájl  (test2.txt) | 0 | EmptyCollectionException |
| Nem létező fájl | - | FileNotFoundException |
| Rossz számformátumú fájl  (test4.txt) | 4  H egy 2 3  S 0 0 3  T -1 -4 3 | InvalidInputException |
| Nem létező V alakzat a fájlban  (test5.txt) | 4  H 1 2 3  V 0 0 3  T -1 -4 3 | InvalidInputException |
| Több alakzatra teszt  (test6.txt) | 4  H 1 2 5,5  S 0 0 6  T -1 -4 2  C 0,5 -0,5 5 | Háromszög, aminek (-1, -4) a középpontja és 2 az oldalhossza |
| Több alakzatra teszt, nagyobb és 1-nél kisebb oldalhosszakkal  (test7.txt) | 4  H 1 2 0,5  S 0 0 112  T -1 -4 100  C 0,5 -0,5 900 | Hexagon, aminek (1,2) a középpontja és 0,5 az oldalhossza |
| Egy alakzatra teszt  (test8.txt) | 1  H 1 2 0,5 | Hexagon, aminek (1,2) a középpontja és 0,5 az oldalhossza |