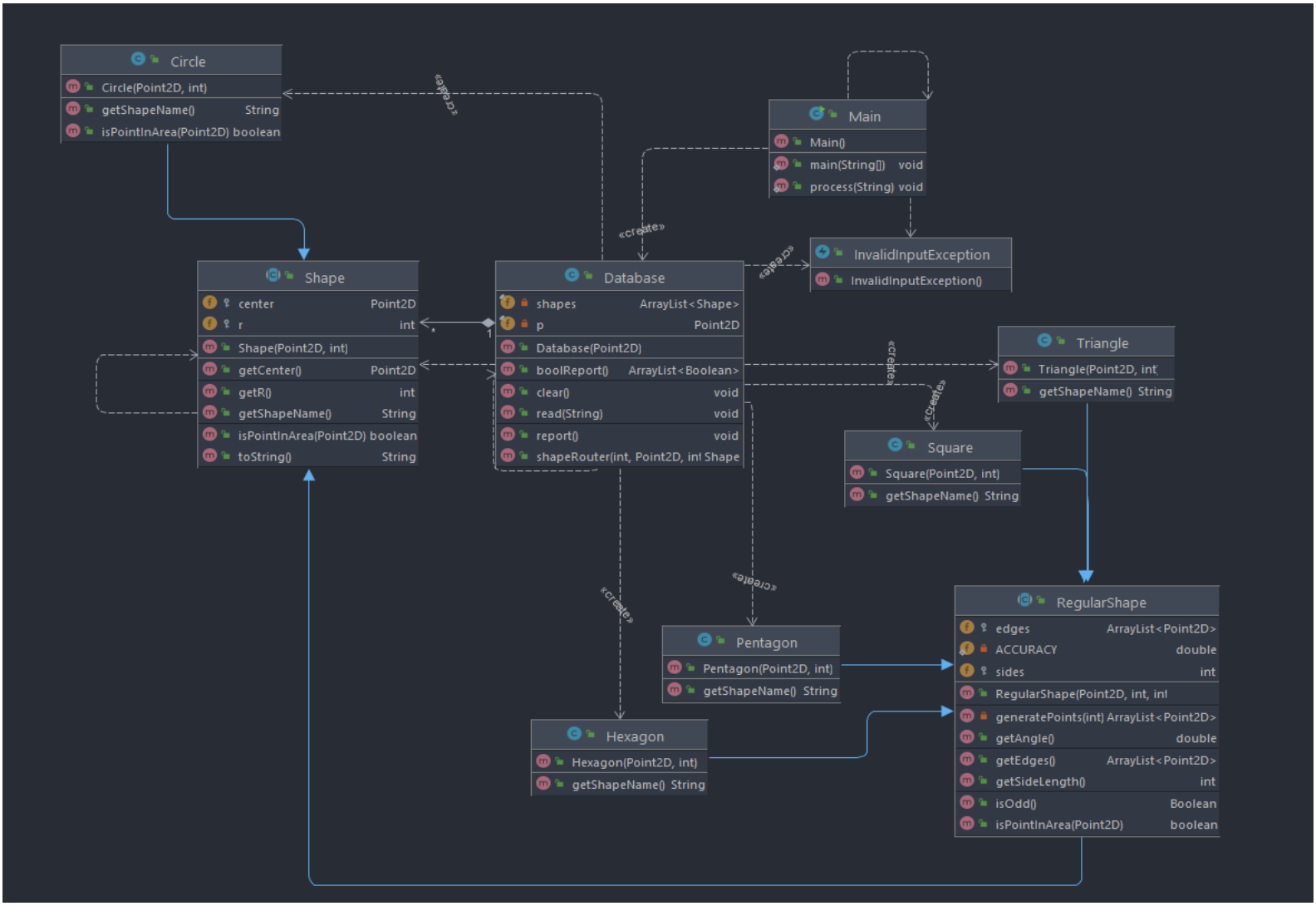
ProgTech // 1. Beadandó // Dokumentáció
Tatai Áron Péter // G07ZOE

6. feladat

6. 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!

Közös elvárás a megoldásoknál, hogy gyűjteményben tároljuk az azonos ősosztályból származtatott osztályok objektumait.
Az objektumok feldolgozása során használjunk foreach szerkezetet. Hibás adatok megadása esetén a program dobjon kivételt, amit kezeljünk is le.
A dokumentációban szerepeljen a feladat leírása,
az osztálydiagram,
illetve a metódusok rövid leírása,
valamint a tesztelés.

Osztálydiagram



Metódusok rövid leírása.

A Main::Process példányosít egy Database instance-et egy p ponttal, ami lesz a vizsgált pont. Utána a read(String) metódus egy filenevet kap, amit után beolvas. Ez a metódus tud exceptiont-dobni, ezt lezekeljük a main-ben.

Database

A read() metódus beolvassa a file,t és a shapeRouter(type, center, r) segítségével feltölti a shapes(ArrayList<Shape>) listát.

A report() metódus stdout-ra kiírja a jelenlegi bebtöltött file részleteid és a belső adatokat, majd a végén kiírja, hogy hány sokszögben van benne a pontunk.

A boolReport() hasonló a report-hoz, tesztléshez van használva és egy ArrayList-ben adja vissza azt, hogy az egyes szokszögetkben benne van-e a pont.

Shape

Az összes 'forma' ősosztálya.
Absztrakt osztály, közös információ a sugár/oldalhossz [r] , középpont [center], ezeknek a getterei, illetve a felülírandó isPointInarea.
Felülírja illetve a toString metódust a szebb kiírásért.

RegularShape

Az összes szabályos szokszögnek a közös kódját artalmazza.
A konstruktor meghívásakor 'legenerálja' az oldalhossz és középpont koorindátájából az összes oldalát a shape-nek. A megadott file-ban int pontosságban voltak a koordináták, itt double pontosságban, illetve ACCURACY int konstans pontosságra kerekítve (ami alapesetben 1000).
Számít még természetesen az is, hogy mivel a sokszögeknek a 'talpukon' kell állniuk, ezért a páratlan oldalúakat el kell egy oldalfélnyivel forgatni.

A isPointInArea függvény felülírása a sokszög (nem feltétlenül szabályos) 'ray-trace' vagyis fénnyaláb módszerrel állapítja meg, hogy a pont a sokszögen belül helyezkedik el. A sokszög határvonala az ebben az implementációban már a sokszögön kívülinek számít.

Triangle, Square, Pentagon, Hexagon

Ezek a függvények mind a RegularShape osztályból származnak, és már nem absztraktak. Itt már csak a sokszög nevét felülíró getShapeName-et és a pontos oldalak számát kell megadni.

Circle

Közvetlenül a Shape osztályból származik, és implementálja a isPointInArea metódust, illetve felülírja a nevet.

Tesztelés

JUnit teszteléssel van megoldva, két fő tesztcsoport:

- File-olvasással kapcsolatos
    - hibás argumentumérték
    - hibás forma típus
    - üres file / nincs file
    - alap konstruktur használata
- geometriai ellenőrzése az alkalmazásnak
    - az összes különböző forma bennevan
    - egyik sem
    - négyszög és kör határpontjai
    - más középponttal rendelkező formák
    - komplex tesztcase; különböző középpontokkal

EOF