

# Programozási technológia

## I. Beadandó feladat

Boda Bálint  
KDHPNI

2022. 10. 10.

### 1. Feladat

Töltsön fel egy gyűjteményt különféle szabályos (kör, szabályos háromszög, négyzet, szabályos hatszög) síkidomokkal! **Adja meg melyik síkidom befoglaló téglalapja a legnagyobb területű!** Egy síkidom befoglaló téglalapja lefedi a síkidomot, oldalai párhuzamosak a tengelyekkel. 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!

## 2. Terv

### 2.1. Típusok

Minden osztályban felüldefiniáljuk `java.lang.Object` osztályból megörökölt `toString()` metódust.

#### 2.1.1. Point

A síkidomok középpontjának reprezentálásához bevezetjük a két dimenziós pontokat ábrázoló `Point` osztályt, mely két adattaggal (`double x,y`) rendelkezik. Az osztály egyetlen egy az adattagokat beállító konstruktorral rendelkezik.

#### 2.1.2. Shape

A különböző síkidomokat a `Shape` absztrakt ősosztályból származtatjuk, mely egy adattagot tartalmaz, a középpontot (`Point center`). Annak érdekében, hogy egységesen tudjuk kezelni a síkidom objektumokat itt kerül deklarálásra a befoglaló téglalap területét kiszámoló `boundingArea()` metódus. A feladatban szereplő síkidomokat két csoportra osztatjuk:

1. Szabályos sokszögek (pl. négyzet, szabályos háromszög és hatszög)
2. Nem sokszögek (pl. kör)

#### 2.1.3. RegularAbstractPolygon

A szabályos sokszögeket egy köztes absztrakt osztályból a `Shape` osztályból származtatott `AbstractRegularPolygon` osztályból származtatjuk. Ebben az osztályban kerül bevezetésre az oldalhossz mező.

#### 2.1.4. Konkrét síkidomok

A kör osztályt a `Shape` osztályból származtatjuk, így fel tudunk venni egy sugár nevű mezőt anélkül, hogy lenne egy felesleges megörökölt oldalhossz mezőnk (ezért nem vettük fel az oldalhossz mezőt a `Shape` osztályban).

A többi (`EquilateralTriangle`, `Square`, `RegularHexagon`) síkidomot az `RegularAbstractPolygon` osztályból származtatjuk.

Minden nem absztrakt síkidom osztályban (`EquilateralTriangle`, `Square`, `RegularHexagon`, `Circle`) felüldefiniáljuk a `boundingArea()` metódust.

A bementi fájlokban a síkidomokat a következő karakterek azonosítják:

Karakter	Síkidom
c	kör
s	négyszet
t	szabályos háromszög
h	szabályos hatszög

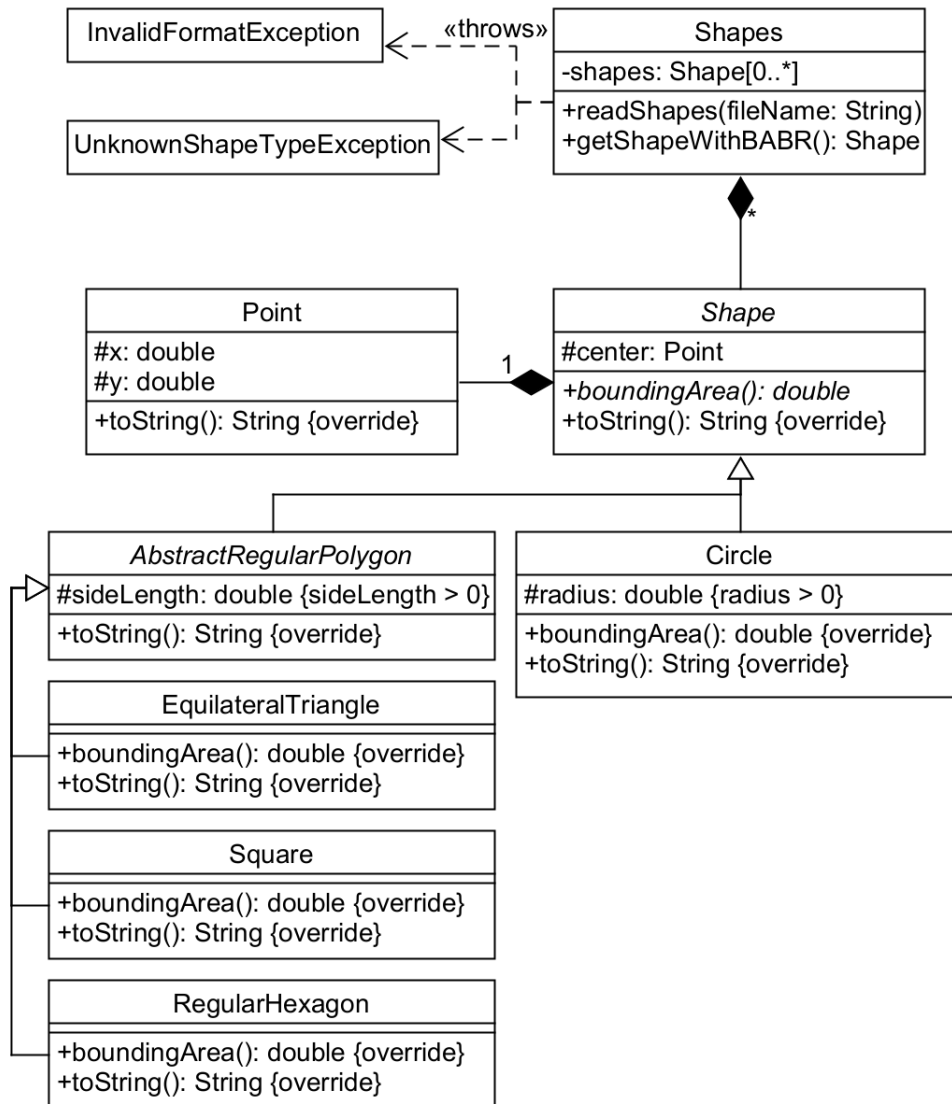
#### 2.1.5. Kivételek

Bevezetjük a `InvalidFormatException` és a `UnknownShapeTypeException` kivételeket, melyek akkor lépnek fel, ha nem megfelelő a bemeneti fájlban szereplő adatok formátuma, valamint, ha olyan karaktert tartalmaz a fájl, melyhez nincs rendelve semmilyen síkidom.

#### 2.1.6. Shapes

A `Shapes` osztályban, egy síkidomokat tároló adatszerkezetet veszünk fel és itt definiáljuk a beolvasást végző `readShapes()` függvényt. A feladat tényleges kérdését a `getShapeWithBABR()` metódus válaszolja meg, mely kiválasztja az adott `Shapos` objektum legnagyobb területű befoglaló téglalappal rendelkező síkidomot.

## 2.2. Osztálydiagramm



## 2.3. Implementálás

A terv implementálását a **Point** osztállyal érdemes kezdeni (mivel ez más osztálytól nem függ). Ezt követően az általánosabb osztályoktól a specifikusabb osztályok felé érdemes haladni.

Használt Java verzió: 17.0.4.1

### 3. Tesztelés

#### 3.1. Fehérdobozos tesztesetek

Leírás	Bemenet	Elvárt eredmény
Nem létező fájl	shapesN.txt	FileNotFoundException
Üres fájl	shapes00.txt	InvalidFormatException
Rossz formátumú fájl #1	shapesW1.txt	InvalidFormatException
Rossz formátumú fájl #2	shapesW2.txt	InvalidFormatException
Ismeretlen síkidomtípust tartalmazó fájl	shapesU.txt	UnknownShapeTypeException
Illegális argumentumú konstruktor hívást eredményező fájltartalom	shapesI.txt	IllegalArgumentException

#### 3.2. Feketedobozos tesztesetek

Leírás	Bemenet	Elvárt eredmény
0 síkidom	shapes0.txt	null
Egy síkidom	shapes1.txt	Circle(radius: 5.0, center: (2.0, 5.0))
Több síkidom	shapes2.txt	Circle(radius: 6.0, center: (2.0, 5.0))
Extra adatok valahány sorban	shapesE.txt	Circle(radius: 6.0, center: (2.0, 5.0))
Több (kb. azonos oldalhosszú) síkidom	shapes3.txt	RegularHexagon(sideLength: 5.0, center: (0.0, 0.0))
Azonos méretű befoglaló téglalap	shapes4.txt	Square(sideLength: 10.0, center: (0.0, 0.0))

#### 3.3. Automatikus tesztelés

Az utóbbi tesztesetek a **BBRTest.java** fájlban szereplő JUnit 5-öt használó tesztkörnyezet részét képezik.