A programozás alapjai 2.

Házi Feladat Dokumentáció

**Geometriai alakzatok**

Készítette: Illényi Marcell, UTBYVT

illenyimarci@gmail.com

2021. 05. 11.

Tartalomjegyzék

[Felhasználói dokumentáció 3](#_Toc71645534)

[Osztályok statikus leírása 3](#_Toc71645535)

[Line 3](#_Toc71645536)

[Felelőssége 3](#_Toc71645537)

[Ősosztályok 3](#_Toc71645538)

[Attribútumok 3](#_Toc71645539)

[Metódusok 3](#_Toc71645540)

[Circle 4](#_Toc71645541)

[Felelőssége 4](#_Toc71645542)

[Ősosztályok 4](#_Toc71645543)

[Attribútumok 4](#_Toc71645544)

[Metódusok 4](#_Toc71645545)

[Parabola 5](#_Toc71645546)

[Felelőssége 5](#_Toc71645547)

[Ősosztályok 5](#_Toc71645548)

[Attribútumok 5](#_Toc71645549)

[Metódusok 5](#_Toc71645550)

[Coordinate 5](#_Toc71645551)

[Felelőssége 5](#_Toc71645552)

[Ősosztályok 5](#_Toc71645553)

[Attribútumok 5](#_Toc71645554)

[Metódusok 5](#_Toc71645555)

[Intersection 6](#_Toc71645556)

[Felelőssége 6](#_Toc71645557)

[Ősosztályok 6](#_Toc71645558)

[Attribútumok 6](#_Toc71645559)

[Metódusok 6](#_Toc71645560)

[UML osztálydiagramm 7](#_Toc71645561)

[Összegzés 7](#_Toc71645562)

[Mit sikerült és mit nem sikerült megvalósítani a specifikációból? 7](#_Toc71645563)

[Mit tanultál a megvalósítás során? 7](#_Toc71645564)

[Továbbfejlesztési lehetőségek 7](#_Toc71645565)

[Képernyőképek a futó alkalmazásról 8](#_Toc71645566)

# Felhasználói dokumentáció

A program egyszerű metszésfeladatok megoldására lett elkészítve. Bemenetnek egy fájlba vár kettő darab egyenest, mindkét egyenest két-két pontjával kell megadni. Ezután kettő darab kört vár, a köröknek a középpontjuk koordinátáit, valamint a sugarukat kell megadni. Ezen kívül vár még 1 darab parabolát négyzetes képletének alakjában (y = 0.5\*p\*(x-u)\*(x-u)+v). A fájlba a kívánt geometriai alakzatokat a következő formátumban kell beírni: egyenes:(x0;y0) (x1;y1) (x2;y2) (x3;y3) kor:(u1;v1) r1 kor:(w;z) r2 parabola:y=0.5\*p\*(x-u2)(x-3)+v2, ahol xn, yn tetszőleges valós szám(az egyenes pontjainak koordinátái), u1, v1, w, z a körök középpontjainak koordinátái (tetszőleges valós számok), r1, r2 a körök sugarai (nemnegatív valós számok) és p, u2, v2, ahol p a parabola paramétere, u2 és v2 pedig a tengelypontja. A paraméterek fájlba történő írása után meg kell adni a fájl nevét a programnak, az pedig a továbbiakban teszi a dolgát, megadja a beírt alakzatok lehetséges metszéspontjait. Ha esetleg két alakzatnak nincs metszéspontja, akkor azt jelzi a program. A bemenetre példa: egyenes:(2.0;0.0) (1.0;-1.0) (3.0;3.0) (4.5;-5.5) kor:(1.0;1.0) 2.0 kor:(3.0;3.0) 2.0 parabola:y=0.5\*2\*(x-3)(x-3)+5, amire az elvárt kimenet:

Metszespontok:

Kor és egyenes : (3; 1), (1; -1)

Egyenes es parabola: (…), (…)

Egyenes es masik egyenes: (3.3; 1.3)

Kor es masik kor: (3; 1), (1; 3)

Helytelen fájlnév esetén a program addig kéri újra, ameddig helyeset nem kap. Hasonlóan helytelen sugár esetén is.

# Osztályok statikus leírása

## Line

### Felelőssége

Egy geometriai egyenest megvalósító osztály. A metszési feladatokban gyakran szereplő alakzat, ezért elengedhetetlen volt a program helyes működéséhez. Az osztály segítségével lehetséges meghatározni egy adott körrel, egy adott másik egyenessel és egy adott parabolával lehetséges metszéspontokat.

### Ősosztályok

A Line osztály publikusan öröklődik az Intersection ősosztályból. Erre azért van szükség, hogy képes legyen kiszámolni a többi alakzattal a metszéspontokat. Az Intesection ősosztálytól metszőfüggvényeket örököl.

### Attribútumok

#### Privát

* Point1: Az egyenesen szereplő egyik pont koordinátái, Coordinate típusú, tehát van x és y koordinátája is. (A Coordinate egy általam létrehozott osztály, melynek két darab privát double tagváltozója van-x és y)
* Point2: Az egyenesen szereplő másik pont koordinátái, hasonlóan a Point1-hez ez is Coordinate típusú.

### Metódusok

.

#### Publikus

- Line(): Default konstruktor, új obektum példányosításakor az azonnali inicializálás elkerülése végett hoztam létre

- Line(double, double): Paraméteres konstruktor, inicializálás miatt hoztam létre

- nVector(): Normálvektort megadó függvény, az egyenes egyenletéhez, valamint meredekségéhez szükségesek a normálvektor koordinátái, ezért hoztam létre

- getPoint1(), setPoint1(), getPoint2(), setPoint2(): getterek, setterek, elérhetőség és utólagos értékadás miatt lettek létrehozva

- Lineintersection(Line&): Paraméterként egyenes objektumot váró metszőfüggvény, kiszámolja az egyenes és a megadott egyenes lehetséges metszéspontját

- CircleIntersection(): Paraméterként egy kör típusú objektumot vár, majd meghatározza az egyenes és kör lehetséges metszéspontját(jait), és visszatér a kapott koordinátákkal

- ParabolaIntersection(Parabola&): Paraméterként parabola objektumot váró metszőfüggvény, kiszámolja az egyenes és parabola lehetséges metszéspontjait, majd visszatér a kapott koordinátákkal

## Circle

### Felelőssége

Egy geometriai kört megvalósító osztály. A metszési feladatokban gyakran szereplő alakzat, ezért elengedhetetlen volt a program helyes működéséhez. Az osztály segítségével lehetséges meghatározni egy adott egyenessel és egy adott körrel lehetséges metszéspontokat.

### Ősosztályok

A Circle osztály publikusan öröklődik az Intersection ősosztályból. Erre azért van szükség, hogy képes legyen kiszámolni a többi alakzattal a metszéspontokat. Az Intesection ősosztálytól metszőfüggvényeket örököl.

### Attribútumok

#### Privát

* centerpoint: A kör középpontja, Coordinate típusú.
* range: A kör sugara.

### Metódusok

- Circle(): Default konstruktor, új obektum példányosításakor az azonnali inicializálás elkerülése végett hoztam létre

- Circle(Coordinate, double): Paraméteres konstruktor: Inicializálás miatt hoztam létre

- getCenterpoint(), getRange(), setCenterpoint(), setRange(): Getterek, setterek, elérhetőség és utólagos értékadás miatt lettek létrehozva

- Lineintersection(Line&): Paraméterként egyenes objektumot váró metszőfüggvény, kiszámolja a kör és a megadott egyenes lehetséges metszéspontjait

-CircleIntersection(Circle&): Paraméterként kört váró metszőfüggvény, a két kör metszéspontjainak kiszámítása a feladata, visszaadja a kiszámított pontokat

## Parabola

### Felelőssége

Egy geometriai parabolát megvalósító osztály. A metszési feladatokban gyakran szereplő alakzat, ezért elengedhetetlen volt a program helyes működéséhez. Az osztály segítségével lehetséges meghatározni egy adott egyenessel és egy adott másik parabolával lehetséges metszéspontokat.

### Ősosztályok

A Parabola osztály publikusan öröklődik az Intersection ősosztályból. Erre azért van szükség, hogy képes legyen kiszámolni a többi alakzattal a metszéspontokat. Az Intesection ősosztálytól metszőfüggvényeket örököl.

### Attribútumok

#### Privát

* p: Double típusú, a parabola paramétere a másodfokú egyenletében
* z: Coordinate típusú, a parabola tengelypontja

### Metódusok

- Parabola(): Default konstruktor új obektum példányosításakor az azonnali inicializálás elkerülése végett hoztam létre

- Parabola(double, Coordinate): Paraméteres konstruktor, inicializálás miatt hoztam létre

- double getP(), Coordinate getz(), void setP(double), void setz(): Getterek, setterek elérhetőség és utólagos értékadás miatt lettek létrehozva

- Lineintersection(Line&): Paraméterként egyenes objektumot váró metszőfüggvény, kiszámolja a parabola és a megadott egyenes lehetséges metszéspontjait, majd visszatér a kapott pontokkal.

- ParabolaIntersection(Parabola&): Paraméterként parabola objektumot váró metszőfüggvény, kiszámolja a két parabola lehetséges metszéspontjait, majd visszatér a kapott pontokkal.

## Coordinate

### Felelőssége

Egy adott koordináta-pontot megvalósító osztály. Ennek az osztálynak a segítségével képesek a metszéspontokkal visszatérni a metszőfügvvények

### Ősosztályok

A Coordinate osztálynak nincs ősosztálya.

### Attribútumok

#### Privát

* x: double típusú, a pont x koordinátája
* y: double típusú, a pont y koordinátája

### Metódusok

- Coordinate(): Default konstruktor új obektum példányosításakor az azonnali inicializálás elkerülése végett hoztam létre

- Coordinate(double, double): Paraméteres konstruktor, inicializálás miatt hoztam létre

- getX(), getY(), setX(), setY(): Getterek, setterek elérhetőség és utólagos értékadás miatt lettek létrehozva

## Intersection

### Felelőssége

Ősosztály, belőle származnak le az alakzatok. Tisztán virtuális metszőfüggvényeket tartalmaz, ezeket öröklik a leszármazottak. Az osztály egy interface.

### Ősosztályok

Az Intersection osztálynak nincs ősosztálya.

### Attribútumok

Az Intersection osztálynak nincsenek attribútumai

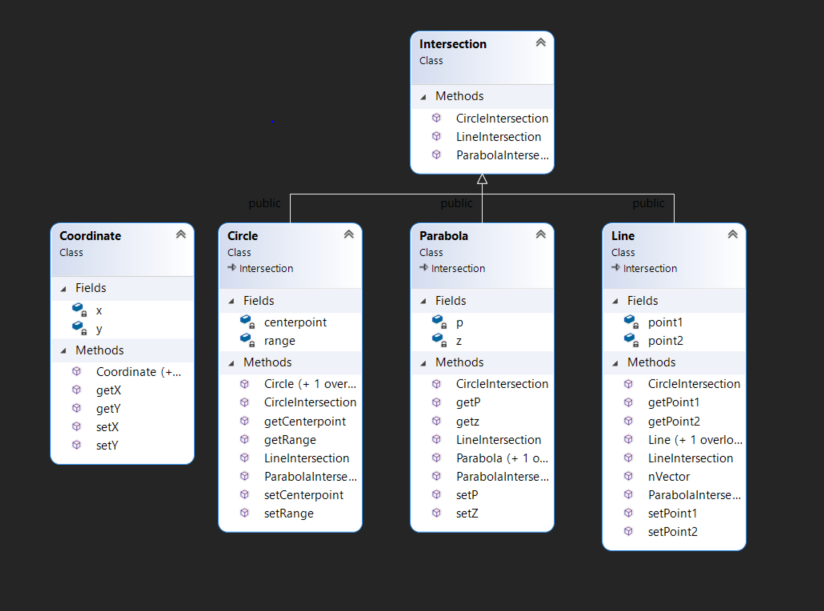
### Metódusok

- virtual LineIntersection(Line&)= 0: az egyenessel való metszéspontokat visszaadó függvény

- virtual CircleIntersection(Circle&)= 0: a körrel való metszéspontokat visszaadó függvény

- virtual ParabolaIntersection(Parabola&)=0: a parabolával való metszéspontokat visszaadó függvény

# UML osztálydiagramm



# Összegzés

## Mit sikerült és mit nem sikerült megvalósítani a specifikációból?

A specifikációhoz képest változott, hogy a program nem a fájlba írja vissza a kapott eredményeket, hanem a standard outputra. Ezt a változtatást azért csináltam, mert több értelmét láttam annak, hogy a pontok rögtön megjelennek a standard outputon, mint annak, hogy újra meg kell nyitni a fájlt minden alkalommal, amikor a program elvégez egy feladatot. A kör és parabola metszéspontjait számító függvényt nem tudtam elkészíteni, túl bonyolultnak bizonyult, így azon a részen hiányos a program.

## Mit tanultál a megvalósítás során?

Tanulságnak le tudtam vonni, hogy az interneten szinte minden megtalálható angolul c++ nyelven történő programozás témájában. A legnagyobb nehézséget a matematikai egyenletek kódba írása okozta.

## Továbbfejlesztési lehetőségek

Az alkalmazást a parabola és kör metszéspontjainak kiszámolására alkalmas függvénnyel, valamint további geometriai alakzatokkal lehetne bővíteni. A célközönség továbbra is a középiskolások és tanárok maradnának.

# Képernyőképek a futó alkalmazásról

