

A PROGRAMOZÁS ALAPJAI 2.

HÁZI FELADAT DOKUMENTÁCIÓ

NÉGYSZÖGEK

KÉSZÍTETTE: NYITRAI BENCE, QEM00R
bence.nyitrai@gmail.com

KÉSZÍTÉS FÉLÉVE: 2022/15/5

TARTALOMJEGYZÉK

Felhasználói dokumentáció	3
Osztályok statikus leírása.....	4
Point.....	4
Felelőssége.....	4
Ősosztályok	4
Attribútumok	4
Metódusok.....	4
SetofPoints.....	5
Felelőssége.....	5
Ősosztályok	5
Attribútumok	5
Metódusok.....	5
Quadrilateral.....	6
Felelőssége.....	6
Ősosztályok	6
Attribútumok	6
Metódusok.....	6
Trapezoid, Parallelogram, Rectangle, Rhombus, Square.....	7
Felelőssége.....	7
Ősosztályok	7
Attribútumok	7
Metódusok.....	7
UML osztálydiagramm	8
Összegzés	9
Mit sikerült és mit nem sikerült megvalósítani a specifikációból?	9
Mit tanultál a megvalósítás során?.....	9
Továbbfejlesztési lehetőségek.....	9
Képernyőképek a futó alkalmazásról.....	10

Felhasználói dokumentáció

Az alkalmazás a síkon elhelyezkedő négyszögek tárolására és manipulálására készült.

A program bemenetét egy fájlból olvassa be (bemenet.txt), a fájlban valós számok szerepelnek szóközzel, új sorral vagy tabulátorral elválasztva, emellett még fontos megjegyezni, hogy a csúcsok az óramutató járásával ellentétes sorrendben kell, hogy beérkezzenek. Példa a bemenetre: **-1 -3 5 -1 3 5 -3 3** (ez egy Négyzet)

Erre a bemenetre a program elvégzi az osztályba sorolást, és példányosít egy Square objektumot ezekkel a koordinátákkal. Ennek a megtörténte után kér a usertől egy számot, amely a nagyítás mértékét fogja megadni, majd egy szöget (fokban!), amely a forgatás mértékét fogja meghatározni. Példa input: **10 720**

Ezt követően a program a standard kimenetre kiírja a különböző transzformációk után a Square új koordinátáit. Példa:

```
Az Y tengelyre valo tukrozes utan:  
Negyzet koordinatai:  
a: 1 -3  
b: -5 -1  
c: -3 5  
d: 3 3  
kozeppontja: 1 1
```

Amikor az összes transzformáció végbe ment az összes négyszögen, akkor átrendezi a heterogén kollekciót az origótól vett távolságuk alapján, és abban a sorrendben írja ki őket. Példa:

```
Az origotol vett tavolsag alapjan sorba rendezett adathalmaz:  
Negyzet: 14.1421
```

Végezetül kimentí a végső transzformáció utáni adatokat egy fájlba (kimenet.txt), és felszabadítja a dinamikusan foglalt memóriaterületeket. Példa:

```
kimenet.txt – Jegyzettömb  
Fájl Szerkesztés Formátum Nézet Súgó  
-10 -30  
50 -10  
30 50  
-30 30
```

Osztályok statikus leírása

Point

Felelőssége

Egy síkbeli pont x és y koordinátáját tárolja tagváltozókként, tagfüggvényeit használok fájlból beolvasásra és a síkbeli pontok által alkotott négyszög típusának meghatározására is.

Össz osztályok

Nincs.

Attribútumok

Privát

- double x: egy síkbeli pont x koordinátáját tárolja
- double y: egy síkbeli pont y koordinátáját tárolja

Védett

Nincs.

Publikus

Nincs.

Metódusok

Privát

Nincs.

Védett

Nincs.

Publikus

- Point(): paraméter nélküli konstruktor
- Point(double , double): paraméteres konstruktor
- Point(const Point&): másoló konstruktor
- void setPoint(double, double) : setterek és getterek
- double getX()const
- double getY()const
- void setx(double)
- void sety(double)
- double distance(const Point&)const: kiszámítja két pont távolságát
- bool iscollinear(const Point& , const Point&)const: eldönti három pontról, hogy egy egyenesre esnek-e
- bool isrightangle(const Point&, const Point&)const: eldönti három pontról, hogy derékszögű háromszöget zárnak-e be
- double slope(const Point&)const: meghatározza a két ponton átmenő egyenes meredekségét
- void reflecttoY(): Y tengelyre tükrözi a pontot
- void reflecttoX(): X tengelyre tükrözi a pontot
- void reflecttotheOrigin(): origóra középpontosan tükröz
- Point operator=(const Point&): értékadás operátor
- bool operator==(const Point&)const: összehasonlítás operátor
- int readfromfile(std::istream&): kiolvassa az adatokat a fájlból

SetofPoints

Felelőssége

A beolvasott pontokat egy SetofPoints objektumban tárolom, majd meghívom a categorize nevű függvényt, amelynek a segédfüggvényei ebben az osztályban találhatók, és ez a függvény visszatér egy olyan típusú négyszög objektumra mutató pointerrel, amelyet a négy csúcs definiál.

Ősosztályok

Nincs.

Attribútumok

Privát

Nincs.

Védett

- Point a: a beolvasott 4 pontot tárolja ezekben a tagváltozóknban
- Point b
- Point c
- Point d

Publikus

Nincs.

Metódusok

Privát

Nincs.

Védett

Nincs.

Publikus

- SetofPoints(): paraméter nélküli konstruktor
- SetofPoints(const Point& , const Point& , const Point& , const Point&): paraméteres konstruktor
- const Point& geta() const: getterek és setterek
- const Point& getb() const
- const Point& getc() const
- const Point& getd() const
- void seta(const Point&)
- void setb(const Point&)
- void setc(const Point&)
- void setd(const Point&)
- bool isQuadrilateral()const: eldönti négy pontról, hogy négyszög-e
- bool isTrapezoid()const: eldönti négy pontról, hogy trapéz-e
- bool isParallelogram()const: eldönti négy pontról, hogy paralelogramma-e
- bool isRectangle()const: eldönti négy pontról, hogy téglalap-e
- bool isRhombus()const: eldönti négy pontról, hogy rombusz-e
- bool isSquare()const: eldönti négy pontról, hogy négyzet-e

Quadrilateral

Felelőssége

Általános négyszögeket tároló osztály, emellett a többi négyszög ősosztálya. Tároláson kívül rendelkezik még a különböző transzformációkat megvalósító metódusokkal.

Ősosztályok

A SetofPoints osztályból publikusan öröklődik, mert a négyszögek ponthalmazt alkotnak a síkon. A SetofPoints a, b, c, d tagváltozói a négyszög négy csúcsa.

Attribútumok

Privát

Nincs.

Védett

- Point middle: a négyszög középpontja
- double distancefromorigin: a négyszög középpontjának az origótól mért távolsága

Publikus

Nincs.

Metódusok

Privát

Nincs.

Védett

Nincs.

Publikus

- Quadrilateral(const Point& pa, const Point& pb, const Point& pc, const Point& pd): paraméteres konstruktor
- Quadrilateral(const Quadrilateral&): másoló konstruktor
- Quadrilateral(): paraméter nélküli konstruktor
- const Point& getmiddle()const: getterek
- double getdistancefromorigin()const
- virtual std::string type()const: visszaadja az objektum típusát(pl.: tégalapnál téglalap a visszatérési értéke)
- void axialreflectiontoY(): Y tengelyre tengelyesen tükrözi a négyszöget
- void axialreflectiontoX(): X tengelyre tengelyesen tükrözi a négyszöget
- void reflectiontotheOrigin(): középpontos tükrözi a négyszöget az origóra
- void centralmagnification(double): a paraméterként kapott számszorosára nagyítás az origóra véve
- void rotation(double): paraméterként kapott szöggel forgatás origó középponttal
- Point calculatemiddle(const Point& ,const Point& ,const Point& ,const Point&): négyszög középpontjának kiszámítása akkor, amikor az objektum keletkezik
- void calculatemiddle():ha megváltoznak a négyszög koordinátái, akkor a megváltozott középpont beállítására szolgál

Trapezoid, Parallelogram, Rectangle, Rhombus, Square

Felelőssége

Azért vettem egybe ezt az 5 osztályt, mert ami különbség köztük az a nevük, és a type függvényük. A feladatuk a nevüknek megfelelő típusú négyszögek példányosítása.

Ősosztályok

Az UML diagrammnak megfelelően származnak le egymásból. Megjegyzendő, hogy a Square osztály virtuálisan öröklődik a Rectangle és a Rhombus osztályból, mivel rendelkezik mindkét osztály tulajdonságaival.

Attribútumok

Privát

Nincs.

Védett

Nincs.

Publikus

Nincs.

Metódusok

Privát

Nincs.

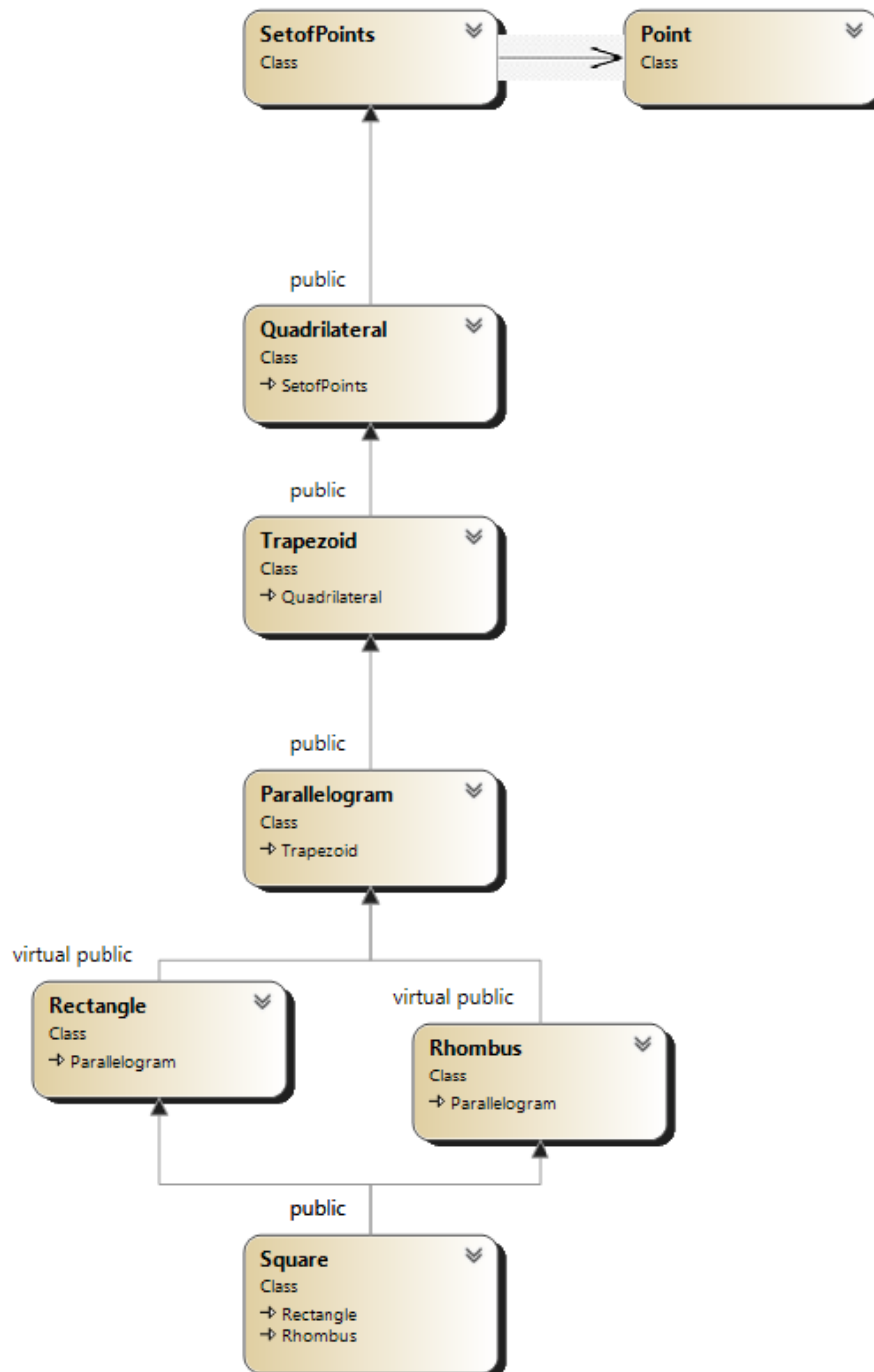
Védett

Nincs.

Publikus

- mindegyik osztály rendelkezik paraméteres és paraméter nélküli konstruktorral
- mindegyik osztály rendelkezik egy virtuálisan felüldefiniált type metódussal, amely a hívó objektum osztályát adja vissza

UML osztálydiagramm



Összegzés

Mit sikerült és mit nem sikerült megvalósítani a specifikációból?

A specifikációból mindent sikerült megvalósítani.

Mit tanultál a megvalósítás során?

Programozási szempontból a csatlakozás és az ez által keletkező memória szivárgás megoldása jelentette a legnagyobb problémát, emellett számos akadályba ütköztem, amikor a program inicializációs függvényét írtam.

A feladat legnehezebb része számomra a matematikai koordinátageometriás ismeretek megfelelő használata volt.

A házfeladat megmutatta számomra, hogy elsőre bármennyire is komplexnek tűnik egy probléma, ha felbontjuk kisebb részekre, akkor sokkal jobban kezelhetővé, megoldhatóvá válik.

Továbbfejlesztési lehetőségek

Az alkalmazásba be lehetne vezetni újdonságként a háromszögek és körök osztályát is.

Egy másik lehetőség a továbbfejlesztésre az, hogy a különböző alakzatokat a program a koordinátaiknak megfelelően kirajzolja egy Descartes-féle derékszögű koordinátarendszerben.

Képernyőképek a futó alkalmazásról

```
Altalanos negyszog koordinatai:  
a: 0.3333 -4.44444  
b: 63333 -2.00101  
c: 333333 72  
d: -34444 511  
kozeppontja: 90555.6 144.139
```

Transzformaciok megkezdodnek:

Adja meg a nagyitas merteket!

■

```
c: -333333 -72  
d: 34444 -511  
kozeppontja: -90555.6 -144.139
```

Az origora valo kozeppontos tukrozes utan:

```
Altalanos negyszog koordinatai:  
a: 0.3333 -4.44444  
b: 63333 -2.00101  
c: 333333 72  
d: -34444 511  
kozeppontja: 90555.6 144.139
```

Az origora vett 1-szeres nagyitas utan :

```
Altalanos negyszog koordinatai:  
a: 0.3333 -4.44444  
b: 63333 -2.00101  
c: 333333 72  
d: -34444 511  
kozeppontja: 90555.6 144.139
```

Az origora vett -720 fokos forgatas utan :

```
Altalanos negyszog koordinatai:  
a: 0.3333 -4.44444  
b: 63333 -2.00101  
c: 333333 72  
d: -34444 511  
kozeppontja: 90555.6 144.139
```

Képernyőképek a futó alkalmazásról

```
Az origotól vett távolság alapján sorba rendezett adathalmaz:
```

```
Altalanos negyszog: 75.889
```

```
Negyzet: 141.421
```

```
Teglalap: 212.132
```

```
Trapez: 269.258
```

```
Rombusz: 565.685
```

```
Paralelogramma: 838.153
```

```
Ezek a koordinatak nem alkotnak negyszoget:
```

```
a: 0.3333 0.3333
```

```
b: 0.3333 0.3333
```

```
c: 333333 72
```

```
d: -34444 511
```

```
Nem érkezett negyszog a bemeneten.
```

```
Transzformaciok megkezdodnek:
```

```
Adja meg a nagyitas merteket!
```

```
12
```

```
Adja meg a forgatas szoget!
```

```
rsfscbsvsvs x
```

```
Hibas user input!
```