



T.C

KOCAELİ SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ
LİSANS EęİTİM ENSTİTÜSÜ
YAZILIM MÜHENDİSLİęİ PROGRAMI

ÖDEV KONUSU

Programlama Lab 1 - Ödev 3

Hazırlayan

ÖęRENCİ ADI: Serhat Arslaner

ÖęRENCİ NUMARASI: 220502043

GITHUB: <https://github.com/serhatarslaner>

Hazırlayan

ÖęRENCİ ADI: Emir Dursun

ÖęRENCİ NUMARASI: 220502001

GITHUB: <https://github.com/emirdrs>

DERS SORUMLUSU

Prof. Dr. H. Tarık DURU

TARİH:01.01.2024

İÇİNDEKİLER

1. ÖZET.....	3
2. Tasarım (H) Dosyaları.....	3
3. CPP Dosyaları.....	7
3.1 Nokta.cpp.....	8
3.2 DogruParcasi.cpp.....	8
3.3 Daire.cpp.....	10
3.4 Ucgen.cpp.....	12
4. KAYNAKÇA.....	14

1. ÖZET

Programlama Lab 1 – 3. Ödevimizi yaparken kütüphane , fonksiyon, sınıf ve koşul ifadeleri kullandık.

Gerekli matematik işlemlerini gerçekleştirdik.

Tasarım ve gerçekleştirme dosyaları kullandık.

Ödevde belirtilen koşulları yerine getirdik.

Ödev No: 3	Tarih 01.01.2024	3/15
------------	------------------	------

2. Tasarım (H) Dosyaları

2.1 Nokta.h

```
1  #include <iostream>
2  #include <string>
3  #ifndef NOKTA_H
4  #define NOKTA_H
5  using namespace std;
6  class Nokta {
7
8  public:
9      Nokta();
10     Nokta(double xy);
11     Nokta(double x1, double y1);
12     Nokta(const Nokta &temp);
13     Nokta(const Nokta &temp, double offset_x, double offset_y);
14
15     double getX() const;
16     void setX(double value);
17
18     double getY() const;
19     void setY(double value);
20
21     void set(double x, double y);
22
23     std::string toString();
24     void yazdir();
25
26 private:
27     double x;
28     double y;
29 };
30 #endif
```

Nokta.h tasarım dosyası içerisinde Nokta adında bir sınıf tutar bu sınıfın public erişime açık fonksiyonları Nokta(), Nokta(double xy), Nokta(double x1,double y1), Nokta(const Nokta &temp), Nokta(const Nokta &temp, double offset_x, double offset_y), double getX() const, void setX(double value), double getY() const, void setY(double value), void set(double x, double y), std::string toString(), void yazdir(). Veri kapsülleme private x ve y.

Nokta(): Parametresiz yapıcı, Nokta(double xy) : tek parametrelili yapıcı tek değere iki koordinat atar, Nokta(double x1,double y1): iki parametrelili yapıcı x ve y koordinatları için sırasıyla iki double değişken alır, Nokta(const Nokta &temp): Başka bir noktayı alıp o noktanın bir kopyasını yeni nokta olarak üreten yapıcı, Nokta(const Nokta &temp, double offset_x, double offset_y): Başka bir nokta (Nokta nesnesi) ve iki double değişken (offset_x ve offset_y) alarak yeni bir nokta (Nokta nesnesi) üretir ve offset değişkenlerini ilk giriş parametresi olarak alınan orijinal noktanın x ve y koordinatlarına ekler.

Ödev No: 3	Tarih 01.01.2024	4/15
------------	------------------	------

2.2 DogruParcasi.h

```
1  #include "Nokta.h"
2  #ifndef DOGRUPARCASI_H
3  #define DOGRUPARCASI_H
4
5  #include <cmath>
6  using namespace std;
7  class DogruParcasi {
8  public:
9      DogruParcasi(const Nokta &temp1, const Nokta &temp2);
10     DogruParcasi(const DogruParcasi &temp);
11     DogruParcasi(const Nokta& temp,double uzunluk, double egim);
12
13     Nokta getP1() const;
14     void setP1(const Nokta& temp1);
15
16     Nokta getP2() const;
17     void setP2(const Nokta& temp1);
18
19
20     Nokta kesisimNokta(const Nokta& temp1);
21
22     double uzunluk();
23
24     Nokta ortaNokta();
25
26     string toString();
27     void yazdir();
28
29 private:
30     Nokta nokta1;
31     Nokta nokta2;
32 };
33 #endif
```

DogruParcasi.h tasarım dosyasi içerisinde DogruParcasi adında bir sınıf tutar bu sınıfın public erişime açık fonksiyonları DogruParcasi(const Nokta &temp1, const Nokta &temp2),DogruParcasi(const DogruParcasi &temp), DogruParcasi(const Nokta& temp,double uzunluk, double egim) Nokta getP1() const, void setP1(const Nokta& temp1), Nokta getP2() const, void setP2(const Nokta& temp1), Nokta kesisimNokta(const Nokta& temp1), double uzunluk(), Nokta ortaNokta(), string toString(), void yazdir() private nokta1 ve nokta2.

DogruParcasi(const Nokta &temp1, const Nokta &temp2): İki uç noktayı Nokta nesnesi olarak alan yapıcı., DogruParcasi(const DogruParcasi &temp): Başka bir DogruParcasi nesnesi alıp onun bir kopyasını yeni bir DogruParcasi nesnesi olarak oluşturan yapıcı.

DogruParcasi(const Nokta& temp,double uzunluk, double egim): Bir Nokta nesnesi (doğru parçasının orta noktası olarak), parçanın uzunluğu (double) ve eğimi(double) değerlerini alarak, doğru parçasının 2 uç noktasının x ve y koordinatlarını hesaplayan yapıcı.

Ödev No: 3	Tarih 01.01.2024	5/15
------------	------------------	------

2.3 Daire.h

```
1  #include <iostream>
2  #include "Nokta.h"
3  #ifndef DAIRE_H
4  #define DAIRE_H
5
6
7  class Daire {
8  public:
9
10
11     Daire(const Nokta& temp1,double r1);
12     Daire(const Daire& temp1);
13     Daire(const Daire& temp1,double x);
14
15     double alan();
16
17     double cevre();
18
19     double kesisim(const Daire& temp1);
20
21     string toString();
22     void yazdir();
23
24
25 private:
26     double r;
27     Nokta merkez;
28
29 };
30
31 #endif
```

Daire.h tasarım dosyası içerisinde Daire adında bir sınıf tutar bu sınıfın public erişime açık fonksiyonları Daire(const Nokta& temp1,double r1), Daire(const Daire& temp1), Daire(const Daire& temp1,double x), double alan(), double cevre(), double kesisim(const Daire& temp1),string toString(), void yazdir(), private r ve merkez.

Daire(const Nokta& temp1,double r1): Merkez (Nokta nesnesi olarak) ve yarıçapı parametre olarak alan yapıcı. Daire(const Daire& temp1): Başka bir Daire nesnesi alıp onun bir kopyasını yeni bir Daire nesnesi olarak oluşturan yapıcı.

Daire(const Daire& temp1,double x): Başka bir Daire nesnesi ve reel bir pozitif x değeri olarak, parametre olarak alınan Daire nesnesini yarıçapı x ile çarpılmış olarak kopyalayan yapıcı.

Double alan(): Dairenin alanını döndürür.

Double cevre(): Dairenin çevresini döndürür.

Double kesisim(const Daire& temp1): Bir Daire nesnesi alır. Parametre olarak gelen daire metodu çağıran dairenin içinde ise 0, daireler birebir örtüşüyorsa 1, hiç kesişim yoksa 2 döndürür.

Ödev No: 3	Tarih 01.01.2024	6/15
------------	------------------	------

2.4 Ucgen.h

```
1  #include "DogruParcasi.h"
2  #include "Nokta.h"
3  #include <cmath>
4  #ifndef UCGEN_H
5  #define UCGEN_H
6
7  using namespace std;
8
9
10
11 class Ucgen
12 {
13 public:
14     Ucgen(Nokta temp1, Nokta temp2, Nokta temp3);
15
16
17     Nokta getP1();
18     void setP1(const Nokta& temp1);
19
20     Nokta getP2();
21     void setP2(const Nokta& temp1);
22
23     Nokta getP3();
24     void setP3(const Nokta& temp1);
25
26
27     string toString();
28     double alan();
29     double cevre();
30     double* acilar();
31
32
33 private:
34     Nokta nokta1;
35     Nokta nokta2;
36     Nokta nokta3;
37 };
38 #endif
39
```

Ucgen.h tasarım dosyası içerisinde Ucgen adında bir sınıf tutar bu sınıfın public fonksiyonları Ucgen(Nokta temp1, Nokta temp2, Nokta temp3), Nokta getP1(), void setP1(const Nokta& temp1) Nokta getP2(), void setP2(const Nokta& temp1), Nokta getP3(), void setP3(const Nokta& temp1), string toString(), double alan(), double cevre(), double* acilar(), private nokta1, nokta2, nokta3.

Ucgen(Nokta temp1, Nokta temp2, Nokta temp3): Üç tane Nokta nesnesi alan yapıcı

Gerekli set-get işlemleri: Nokta getP1(), void setP1(const Nokta& temp1) Nokta getP2(), void setP2(const Nokta& temp1), Nokta getP3(), void setP3(const Nokta& temp1)

double alan(): Bu üç noktanın temsil ettiği üçgenin alanını hesaplar ve döndürür.

double cevre(): DoğruParcasi sınıfının uzunluk metodunu kullanarak ucgen cevresi hesaplar.

double* acilar(): Üçgenin açılarını üç öğeli double dizi olarak hesaplar ve döndürür

Ödev No: 3	Tarih 01.01.2024	7/15
------------	------------------	------

3. CPP Dosyaları

3.1 Nokta.cpp

```
1  #include "Nokta.h"
2
3
4  Nokta::Nokta() {
5      setX(0);
6      setY(0);
7  }
8
9  Nokta::Nokta(double xy) {
10     setX(xy);
11     setY(xy);
12 }
13 Nokta::Nokta(double x1, double y1) {
14     setX(x1);
15     setY(y1);
16 }
17 Nokta::Nokta(const Nokta &temp) {
18     setX(temp.getX());
19     setY(temp.getY());
20 }
21 Nokta::Nokta(const Nokta &temp, double offset_x, double offset_y) {
22     setX(temp.getX() + offset_x);
23     setY(temp.getY() + offset_y);
24 }
25
26 double Nokta::getY() const { return y; }
27
28 void Nokta::setY(double value) { y = value; }
29 double Nokta::getX() const { return x; }
30
31 void Nokta::setX(double value) { x = value; }
32
33 void Nokta::set(double x1, double y1) {
34     setX(x1);
35     setY(y1);
36 }
37
38 string Nokta::toString() {
39     return "(" + std::to_string(x) + "," + std::to_string(y) + ")";
40 }
41
42 void Nokta::yazdir() {
43     cout << "'String' Koordinat : " << toString() << endl;
44 }
```

Nokta.cpp dosyasında Nokta.h dosyasında tasarlanan fonksiyonlar gerçekleşir.

Nokta::Nokta(): parametresiz yapıcı default değerler döndürür.

Nokta::Nokta(double xy): gerekli set yapıları sayesinde tek değişkene 2 nokta atar.

Nokta::Nokta(double x1,double y1): gerekli set yapıları sayesinde x1 ve y1 fonksiyonları 2 farklı double değişkeni alır.Nokta::Nokta(const Nokta &temp): Başka bir noktayı referans alıp o noktanın bir kopyasını oluşturur.

Nokta::Nokta(const Nokta &temp, double offset_x, double offset_y): Başka bir nokta (Nokta nesnesi) ve iki double değişken (ofset_x ve ofset_y) alarak yeni bir nokta (Nokta nesnesi) üretir ve ofset değişkenlerini ilk giriş parametresi olarak alınan orijinal noktanın x ve y koordinatlarına ekler.

Ödev No: 3	Tarih 01.01.2024	8/15
------------	------------------	------

3.2 DogruParcasi.cpp

```
1  #include "DogruParcasi.h"
2
3  using namespace std;
4
5  DogruParcasi::DogruParcasi(const Nokta &temp1, const Nokta &temp2)
6  {
7      setP1(temp1);
8      setP2(temp2);
9  }
10 DogruParcasi::DogruParcasi(const DogruParcasi &temp) {
11     setP1(temp.getP1());
12     setP2(temp.getP2());
13 }
14 DogruParcasi::DogruParcasi(const Nokta &temp, double uzunluk, double egim)
15 {
16     double deltaX = (uzunluk / 2) / sqrt(pow(egim, 2) + 1);
17     double deltaY = deltaX * egim;
18
19     double x1 = temp.getX() - deltaX;
20     double y1 = temp.getY() - deltaY;
21     double x2 = temp.getX() + deltaX;
22     double y2 = temp.getY() + deltaY;
23
24     setP1(Nokta(x1, y1));
25     setP2(Nokta(x2, y2));
26 }
27
28 double DogruParcasi::uzunluk() {
29     double sonuc = sqrt(pow(getP2().getX() - getP1().getX(), 2) +
30                          pow(getP2().getY() - getP1().getY(), 2));
31
32     return sonuc;
33 }
34
35 Nokta DogruParcasi::getP1() const { return nokta1; }
36 void DogruParcasi::setP1(const Nokta &temp1) { nokta1=temp1; }
37 Nokta DogruParcasi::getP2() const { return nokta2; }
38 void DogruParcasi::setP2(const Nokta& temp1) { nokta2 = temp1; }
39
```

```

40  Nokta DogruParcasi::kesisimNokta(const Nokta& temp1) {
41
42      double m1 = (getP2().getY() - getP1().getY()) / (getP2().getX() - getP1().getX());
43
44
45      double m2 = -1 / m1;
46
47      double kesisimX = (m1 * temp1.getX() - m2 * temp1.getX() - getP1().getY() + getP2().getY() / (m1 - m2));
48
49      double kesisimY = m1 * (kesisimX - getP1().getX()) + getP1().getY();
50
51      return Nokta(kesisimX, kesisimY);
52  }
53
54
55  Nokta DogruParcasi::ortaNokta() {
56      double xorta = (getP1().getX() + getP2().getX()) / 2;
57      double yorta = (getP1().getY() + getP2().getY()) / 2;
58      Nokta orta(xorta, yorta);
59
60      return orta;
61  }
62
63  string DogruParcasi::toString() {
64      return getP1().toString() + getP2().toString();
65  }
66
67
68  void DogruParcasi::yazdir() {
69      std::cout << "DogruParcasi 1. ve 2. Noktalar: " << toString() << endl;
70  }
71

```

DogruParcasi.cpp dosyasında DogruParcasi.h dosyasında tasarlanan fonksiyonlar gerçekleştirilir.

DogruParcasi(const Nokta &temp1, const Nokta &temp2): iki noktayı nesne olarak alır

DogruParcasi(const DogruParcasi &temp): başka bir doğru parçası nesnesini kopya olarak üretir

DogruParcasi(const Nokta &temp, double uzunluk, double egim): Bir Nokta nesnesi (doğru parçasının orta noktası olarak), parçanın uzunluğu (double) ve eğimi (double) değerlerini alarak, doğru parçasının 2 uç noktasının x ve y koordinatlarını hesaplayan yapıcı.

Uzunluk(): iki nokta arası uzaklık formülünden yararlanarak doğru uzunluğunu döndürür.

OrtaNokta(): $(x_1+x_2) / 2$ ve $(y_1+y_2) / 2$ orta nokta formülünün koda dökülmüş şeklidir.

Ödev No: 3	Tarih 01.01.2024	10/15
------------	------------------	-------

3.3 Daire.cpp

```
1  #include "Daire.h"
2  #include <iostream>
3  #include <cmath>
4
5
6  Daire::Daire(const Nokta& templ, double r1) {
7      r = r1;
8      merkez = templ;
9  }
10
11
12  Daire::Daire(const Daire& templ) {
13      r = templ.r;
14      merkez = templ.merkez;
15  }
16
17  Daire::Daire(const Daire& templ, double x) {
18      r = templ.r * x;
19      merkez = templ.merkez;
20  }
21
22
23  double Daire::alan() {
24
25      double a = 3.14 * pow(r,2);
26      return a;
27  }
28
29
30  double Daire::cevre() {
31      double c = 2 * 3.14 * r;
32      return c;
33  }
```

```

34
35 double Daire::kesisim(const Daire& temp1) {
36
37     double uzaklik = sqrt(pow(merkez.getX() - temp1.merkez.getX(), 2) +
38         pow(merkez.getY() - temp1.merkez.getY(), 2));
39
40     if ((temp1.r + r) >= uzaklik)
41     {
42         if (temp1.r == r && temp1.merkez.getX() == merkez.getX()
43             && temp1.merkez.getY() == merkez.getY()) {
44             return 1;
45         }
46         return 0;
47     }
48     else { return 2; }
49 }
50
51
52 string Daire::toString() {
53     return "Merkez: (" + std::to_string(merkez.getX()) + ", " + std::to_string(merkez.getY()) + ") " + "Yarıçap: " + std::to_string(r);
54 }
55
56 void Daire::yazdir() {
57     std::cout << toString() << std::endl;
58 }
59 }

```

Daire.cpp dosyasında Daire.h dosyasında tasarlanan fonksiyonlar gerçekleştirilir.

Daire(const Nokta& temp1, double r1): Merkez (Nokta nesnesi olarak) ve yarıçapı parametre olarak alan yapıcı.

Daire(const Daire& temp1): Başka bir Daire nesnesi alıp onun bir kopyasını yeni bir Daire nesnesi olarak oluşturan yapıcı.

Daire(const Daire& temp1, double x): Başka bir Daire nesnesi ve reel bir pozitif x değeri alarak, parametre olarak alınan Daire nesnesini yarıçapı x ile çarpılmış olarak kopyalayan yapıcı.

Alan(): $\pi \cdot r^2$ formülünün kod biçiminde yazılmış halidir.

Cevre(): $2 \cdot \pi \cdot r$ formülünün kod biçiminde yazılmış halidir.

Ödev No: 3	Tarih 01.01.2024	12/15
------------	------------------	-------

3.4 Ucgen.cpp

```
1  #include "Ucgen.h"
2  #include <cmath>
3  Ucgen::Ucgen(Nokta temp1, Nokta temp2, Nokta temp3){
4      setP1(temp1);
5      setP2(temp2);
6      setP3(temp3);
7  }
8
9  Nokta Ucgen::getP1() {return nokta1;}
10 void Ucgen::setP1(const Nokta& temp1) { nokta1 = temp1; }
11
12 Nokta Ucgen::getP2() { return nokta2; }
13 void Ucgen::setP2(const Nokta& temp1) { nokta2 = temp1; }
14
15 Nokta Ucgen::getP3() { return nokta3; }
16 void Ucgen::setP3(const Nokta& temp1) { nokta3 = temp1; }
17
18 string Ucgen::toString() {
19
20
21     return ("Ucgenin noktaları : " + getP1().toString() + ", " + getP2().toString()
22           + ", " + getP3().toString());
23 }
24 double Ucgen::alan() {
25
26     double yaricevre = (cevre() / 2);
27     DogruParcasi kenar1(nokta1, nokta2);
28     DogruParcasi kenar2(nokta1, nokta3);           //Heron formülü
29     DogruParcasi kenar3(nokta2, nokta3);
30     double alan = sqrt(yaricevre * (yaricevre - kenar1.uzunluk()) * (yaricevre - kenar2.uzunluk()) * (yaricevre - kenar3.uzunluk()));
31     return alan;
32 }
33 double Ucgen::cevre() {
34     DogruParcasi kenar1(nokta1, nokta2);
35     DogruParcasi kenar2(nokta1, nokta3);
36     DogruParcasi kenar3(nokta2, nokta3);
37
38     double cevre = kenar1.uzunluk() + kenar2.uzunluk() + kenar3.uzunluk();
39     return cevre;
40 }
41
42 double* Ucgen::acilar() {
43
44     DogruParcasi kenar1(nokta1, nokta2);
45     DogruParcasi kenar2(nokta1, nokta3);
46     DogruParcasi kenar3(nokta2, nokta3);
47     double A = alan();
48     double aci1= (180 / 3.14) * asin(2 * A / (kenar1.uzunluk() * kenar2.uzunluk()));
49     double aci2 = (180 / 3.14) * asin(2 * A / (kenar2.uzunluk()*kenar3.uzunluk()));
50     double aci3 = 180 - (aci1 + aci2);
51
52     static double sonuc[3] = {aci1,aci2,aci3};
53
54
55     return sonuc;
56 }
```

Ucgen.cpp dosyasında Ucgen.h dosyasında tasarlanan fonksiyonlar gerçekleştirir

Ucgen(Nokta temp1, Nokta temp2, Nokta temp3): Üç adet nokta nesnesi alan yapıcı fonksiyon.

Alan(): Heron Formülünün kod biçiminde yazılmış halidir.

Cevre(): Üç adet verilen noktaların birbirine olan uzunluklarının toplamı üçgenin çevresini veriyor.

Acilar(): Açı1 ve Açı2 Hesaplanırken aşağıdaki formül kod biçiminde yazılır.

$$\angle A = \frac{180^\circ}{\pi} \arcsin\left(\frac{2A}{bc}\right)$$

$$\angle B = \frac{180^\circ}{\pi} \arcsin\left(\frac{2A}{ac}\right)$$

$$\angle C = \frac{180^\circ}{\pi} \arcsin\left(\frac{2A}{ab}\right)$$

Açı3 ise açı1 ve açı2 bulunduktan sonra 180'den açı1 ve açı2 çıkarıldığı zaman açı3 bulunur.

Ödev No: 3	Tarih 01.01.2024	14/15
------------	------------------	-------

4. KAYNAKÇA

Metin Tabanlı Yapay Zeka Modeli ChatGPT(GPT-3.5 mimarisi)

W3Schools. <https://www.w3schools.com/>

Wikipedia. https://tr.wikipedia.org/wiki/Heron_formülü

Wikipedia. https://tr.wikipedia.org/wiki/Kosin%C3%BCs_teoremi

Khan Academy. <https://tr.khanacademy.org/math/geometry/hs-geo-analytic-geometry/hs-geo-distance-and-midpoints/a/distance-formula>

Ödev No: 3	Tarih 01.01.2024	15/15
------------	------------------	-------