



T.C

**KOCaeli SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ**  
**MÜHENDİSLİK VE DOęA BİLİMLERİ FAKÜLTESİ**  
**YAZILIM MÜHENDİSLİęİ PROGRAMI**

## **ÖDEV KONUSU**

# **GEOMETRİK PROBLEMLER**

**Hazırlayan**

**ÖęRENCİ ADI :ŞEYMA LAÇİN**  
**ÖęRENCİ NUMARASI :220502035**  
**ÖęRENCİ ADI:ZEHRA YARDIMCI**  
**ÖęRENCİ NUMARASI:220502038**

**DERS SORUMLUSU**

**Prof.Dr. HÜSEYİN TARIK DURU**

**01.01.2024**

---

## İÇİNDEKİLER

1.	<u>ÖZET (ABSTRACT)</u> .....	3
2.	<u>GİRİŞ (INTRODUCTION)</u> .....	3
	2.1 Temel Amacımız.....	3
	2.2 Kullanılan Temel Yöntemler ve Araçlar.....	3
	2.3 Beklenen işlemler.....	4
3.	<u>YÖNTEM (METHOD)</u> .....	5
3.1	<u>Nokta sınıfı Metodları</u> .....	5
3.2	<u>DogruParcasi sınıfı Metodları</u> .....	6
3.3	<u>Daire sınıfı Metodları</u> .....	7
3.4	<u>Ucgen sınıfı Metodları</u> .....	8
4.	<u>SONUÇ VE ÖĞRENİLEN DERSLER</u> .....	9
5.	<u>GİTHUB HESAPLARI</u> .....	10
6.	<u>KAYNAKÇA</u> .....	10

Ödev No: 3	Tarih 01.01.2024	2/10
------------	------------------	------

## 1. ÖZET

Bu C++ programı, geometrik problemleri ele alarak Nokta, DoğruParçası, Daire ve Üçgen sınıflarını içermektedir. Her bir sınıfın özel yapıları ve işlevleri, programın modüler ve etkili bir şekilde çalışmasını sağlar. Ayrıca, sınıflar arasındaki ilişkiler ve metodların kullanımı, geometrik problemleri çözmek için olanaklar sunar. Program, matematiksel hesaplamaları gerçekleştirmek ve geometrik nesneleri temsil etmek için sınıf ve metotları başarıyla kullanmaktadır.

## 2. GİRİŞ

Bu C++ programı, geometrik nesnelerin (Nokta, Doğru Parçası, Üçgen) temsilini ve bu nesneler üzerinde yapılabilecek temel geometrik hesaplamaları sağlayan bir yapıyı içermektedir. Program, geometrik nesnelerin özelliklerini ve birbirleriyle etkileşimini modelleyerek temel geometrik hesaplamaları gerçekleştirir. Her bir sınıf, nesnesinin durumunu yöneten işlevselliği sağlar ve bu sayede farklı geometrik nesneler oluşturulabilir ve manipüle edilebilir. Programın kullanımı, özellikle geometrik nesneler üzerinde çeşitli hesaplamalar yapmak isteyen kullanıcılar için uygundur.

### 2.1 Temel Amacımız

Bu C++ programının temel amacı, geometrik nesnelerin (Nokta, Doğru Parçası, Üçgen, Daire) temsilini sağlamak ve bu nesneler üzerinde bazı temel geometrik hesaplamaları gerçekleştirmektir. Her bir geometrik nesne belirli özelliklere (örneğin, Nokta sınıfı x ve y koordinatları gibi) sahiptir ve bu nesneler arasındaki ilişkiler ve hesaplamalar program içerisinde modellenmiştir..

### 2.2 Kullanılan Temel Yöntemler ve Araçlar

Aşağıdaki temel yöntemler ve araçlar kullanılarak gerçekleştirilmiştir:

#### a. Sınıf ve Nesne kavramları:

Her geometrik nesne (daire ,üçgen ,nokta,doğru parçası) bir sınıf olarak temsil edilmiştir

#### b. Encapsulation(Kapsülleme):

Sınıfların içindeki özellikler veri üyeleri) özel olarak ve bu özelliklere sadece belirli

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

---

fonksiyonlarla erişilebilir.

**c. Inheritance(miras alma):**

Bazı sınıflar, diğer sınıflardan türetilmiştir. Örneğin, **DogruParcasi** sınıfı **Nokta** sınıfından türetilmiştir. Miras alma, sınıflar arasında bir hiyerarşi oluşturarak kodun düzenli ve tekrar kullanılabilir olmasını sağlar.

**d. Pointer Kullanımı:**

Dinamik bellek yönetimi için pointer'lar kullanılmıştır. Örneğin, `double* Ucgen::acilar() const` fonksiyonunda dinamik bellekten alan ayrılmıştır.

**e. Matematiksel Hesaplamalar:**

Geometrik hesaplamalarda trigonometrik fonksiyonlar (örneğin, `tan`, `acos`), karekök alma (`sqrt`), üs alma (`pow`) gibi matematik fonksiyonları kullanılmıştır

**f. String İşlemleri:**

`toString` fonksiyonları ile geometrik nesnelerin string temsili oluşturulmuştur.

String birleştirme ve dönüştürme işlemleri için C++ string sınıfı fonksiyonları kullanılmıştır

**g. Koşullu İfadeler ve Döngüler:**

İf-else ifadeleri ve döngüler (`for`, `while`) kullanılarak çeşitli durumlar için kontrol yapıları oluşturulmuştur

**h. Standart Giriş/Çıkış (I/O) İşlemleri:**

`cout` ve `cin` gibi standart giriş/çıkış işlemleri kullanılarak programın kullanıcı ile etkileşimi sağlanmıştır

**i. Standart Giriş/Çıkış (I/O) İşlemleri:**

`cout` ve `cin` gibi standart giriş/çıkış işlemleri kullanılarak programın kullanıcı ile etkileşimi sağlanmıştır

## 2.3 Beklenen işlemler

### Nokta Sınıfı İşlemleri:

- Koordinatları değiştirme
- İki nokta arasındaki uzaklığı hesaplama

### Üçgen Sınıfı İşlemleri:

Ödev No: 3	Tarih 01.01.2024	4/10
------------	------------------	------

- 
- Üçgenin alanını hesaplama
  - Üçgenin çevresini hesaplama
  - Üçgenin açılarını hesaplama

#### **Daire Sınıfı İşlemleri:**

- Dairenin alanını hesaplama
- Dairenin çevresini hesaplama
- İki daire arasındaki kesişimi kontrol etme

#### **Doğru Parçası Sınıfı İşlemleri:**

- Doğru parçasının uzunluğunu hesaplama
- İki noktanın kesişim noktasını bulma
- Doğru parçasının orta noktasını hesaplama

#### **Genel İşlemler:**

- Nesnelerin string temsilini oluşturma
- Nesneleri ekrana yazdırma
- Temel matematiksel hesaplamalar (trigonometrik fonksiyonlar, karekök alma, üs alma)
- Pointer kullanımı ve bellek yönetimi
- Header dosyaları ve sınıflar arasındaki ilişkiler

### **3. YÖNTEM**

Bu bölümde, geometrik problemleri çözmek için kullanılan ana metotlar ve bu metotların detaylı adımları açıklanmaktadır.

#### **3.1 Nokta sınıfı Metodları**

Ödev No: 3	Tarih 01.01.2024	5/10
------------	------------------	------

```
1 #pragma once
2
3 #include<iostream>
4 #include <cmath>
5 #include <string>
6 #define M_PI 3.14159265358979323846
7
8 using namespace std;
9
10 class Nokta {
11 private:
12     double x;
13     double y;
14
15 public:
16     Nokta(); // Parametresiz yapıcı
17     Nokta(double value); // Tek parametrelili yapıcı
18     Nokta(double xCoord, double yCoord); // İki parametrelili yapıcı
19     Nokta(const Nokta& other); // Kopya yapıcı
20     Nokta(const Nokta& other, double offset_x, double offset_y); // Ofsetli yapıcı
21     double getX() const;
22     double getY() const;
23     void setX(double value);
24     void setY(double value);
25     void setCoordinates(double xCoord, double yCoord);
26     string toString() const;
27     void yazdir() const;
28 };
29
30
```

- **Parametresiz Yapıcı (Nokta()):** Noktanın x ve y koordinatlarını sıfıra ayarlar.
- **Tek Parametrelili Yapıcı (Nokta(double value)):** İki koordinata da aynı değeri atar.
- **İki Parametrelili Yapıcı (Nokta(double xCoord, double yCoord)):** x ve y koordinatları için sırasıyla iki double değişken alır.
- **Kopya Yapıcı (Nokta(const Nokta& other)):** Başka bir noktanın kopyasını oluşturan yapıcı.
- **Ofsetli Yapıcı (Nokta(const Nokta& other, double offset\_x, double offset\_y)):** Başka bir nokta ve iki double değişken olarak yeni bir nokta üreten yapıcı; ofset değişkenlerini orijinal noktanın koordinatlarına ekler.
- **getX() ve getY():** x ve y koordinatlarını get etmek için kullanılan metotlar.
- **setX(double value) ve setY(double value):** x ve y koordinatlarını set etmek için kullanılan metotlar.
- **setCoordinates(double xCoord, double yCoord):** Hem x hem de y koordinatlarını aynı anda set etmek için kullanılan metot.
- **toString():** Noktanın koordinatlarının string gösterimini döndüren metot.
- **yazdir():** toString metodunu kullanarak noktanın koordinatlarını ekrana yazdıran metot.

Ödev No: 3	Tarih 01.01.2024	6/10
------------	------------------	------

### 3.2 DogruParcasi sınıfı Metodları

```
1  #pragma once
2  #include "Nokta.h"
3  #include <iostream>
4  #include <cmath>
5  #include <string>
6  #define M_PI 3.14159265358979323846
7
8  using namespace std;
9
10 class DogruParcasi {
11 private:
12     Nokta baslangicNoktasi;
13     Nokta bitisNoktasi;
14
15 public:
16     // İki uç noktayı Nokta nesnesi olarak alan yapıcı
17     DogruParcasi(const Nokta& baslangic, const Nokta& bitis);
18     // Başka bir DogruParcasi nesnesi alıp onun bir kopyasını yeni bir DogruParcasi nesnesi olarak oluşturan yapıcı.
19     DogruParcasi(const DogruParcasi& other);
20     // Bir Nokta nesnesi, uzunluk ve eğim alarak DogruParcasi nesnesi oluşturan yapıcı
21     DogruParcasi(const Nokta& ortaNokta, double uzunluk, double egim);
22     const Nokta& getBaslangicNoktasi() const;
23     const Nokta& getBitisNoktasi() const;
24     // Uzunluğu hesaplayan metot
25     double uzunluk() const;
26     // Bir Nokta nesnesini parametre olarak alarak kesişim noktasını hesaplayan metot
27     Nokta kesisimNoktasi(const Nokta& digerNokta) const;
28     // Dogru parçasının orta noktasını hesaplayan metot
29     Nokta ortaNokta() const;
30     // Nesneyi string olarak temsil eden metot
31     string toString() const;
32     void yazdir() const;
33     void setP1(const Nokta& newP1);
34     void setP2(const Nokta& newP2);
35 };
36
```

- **İki Uç Noktayı Alarak Başlatan Yapıcı (DogruParcasi(const Nokta& baslangic, const Nokta& bitis)):** İki uç noktayı Nokta nesnesi olarak alan yapıcı.
- **Kopya Yapıcı (DogruParcasi(const DogruParcasi& other)):** Başka bir DogruParcasi nesnesini alıp onun bir kopyasını yeni bir DogruParcasi nesnesi olarak oluşturan yapıcı.
- **Nokta, Uzunluk ve Eğim Alarak Oluşturan Yapıcı (DogruParcasi(const Nokta& ortaNokta, double uzunluk, double egim)):** Bir Nokta nesnesi, uzunluk ve eğim alarak DogruParcasi nesnesi oluşturan yapıcı.
- **getBaslangicNoktasi() ve getBitisNoktasi():** Dogru parçasının başlangıç ve bitiş noktalarını get etmek için kullanılan metotlar.
- **uzunluk():** Dogru parçasının uzunluğunu hesaplayan metot.
- **kesisimNoktasi(const Nokta& digerNokta):** Bir Nokta nesnesini parametre olarak alarak kesişim noktasını hesaplayan metot.
- **ortaNokta():** Dogru parçasının orta noktasını hesaplayan metot.
- **toString():** Dogru parçasının string gösterimini döndüren metot.
- **yazdir():** toString metodunu kullanarak dogru parçasının başlangıç ve bitiş noktalarını ekrana yazdıran metot.

### 3.3 Daire sınıfı Metodları

Ödev No: 3	Tarih 01.01.2024	7/10
------------	------------------	------

```
Lab Daire alan() const
1 #pragma once
2 #include "Nokta.h"
3 #include <iostream>
4 #include <cmath>
5 #include <string>
6 #define M_PI 3.14159265358979323846
7
8 using namespace std;
9
10 class Daire {
11 private:
12     Nokta merkez; // Dairenin merkezini temsil eden Nokta nesnesi
13     double yaricap; // Dairenin yarıçapı
14
15 public:
16     // Merkez (Nokta nesnesi olarak) ve yarıçapı parametre olarak alan yapıcı.
17     Daire(const Nokta& m, double r);
18     // Başka bir Daire nesnesi alıp onun bir kopyasını yeni bir Daire nesnesi olarak oluşturan yapıcı.
19     Daire(const Daire& other);
20     // Başka bir Daire nesnesi ve reel bir pozitif x değeri alarak, parametre olarak alınan Daire nesnesini yarıçapı x ile çarpılmış olarak kopyalayan y.
21     Daire(const Daire& other, double x);
22     // Dairenin alanını hesaplayan metot.
23     double alan() const;
24     // Dairenin çevresini hesaplayan metot.
25     double cevre() const;
26     // İki dairenin kesişim durumunu kontrol eder. İç içe ise 0, örtüşüyorsa 1, hiç kesişim yoksa 2 döndürür.
27     int kesisim(const Daire& digerDaire) const;
28     // Dairenin merkezi ve yarıçapını String halinde döndüren metot.
29     string toString() const;
30     void yazdir() const;
31 };
32
33
```

- **Merkezi ve Yarıçapı Parametre Olarak Alan Yapıcı (Daire(const Nokta& m, double r)):** Merkezi bir Nokta nesnesi olarak ve yarıçapı parametre olarak alan yapıcı.
- **Kopya Yapıcı (Daire(const Daire& other)):** Başka bir Daire nesnesini alıp onun bir kopyasını yeni bir Daire nesnesi olarak oluşturan yapıcı.
- **Başka Daire ve Pozitif X Değeri Alarak Oluşturan Yapıcı (Daire(const Daire& other, 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():** Dairenin alanını hesaplayan metot.
- **cevre():** Dairenin çevresini hesaplayan metot.
- **kesisim(const Daire& digerDaire):** İki dairenin kesişim durumunu kontrol eden metot. İç içe ise 0, örtüşüyorsa 1, hiç kesişim yoksa 2 döndürür.
- **toString():** Dairenin merkezi ve yarıçapını string halinde döndüren metot.
- **yazdir():** toString metodunu kullanarak dairenin bilgilerini ekrana yazdıran metot.

### 3.4 Ucgen sınıfı Metodları

Ödev No: 3	Tarih 01.01.2024	8/10
------------	------------------	------



```
Lab Uçgen
1 #pragma once
2 #include "Nokta.h"
3 #include "DogruParcasi.h"
4 #include <iostream>
5 #include <cmath>
6 #include <string>
7 #define M_PI 3.14159265358979323846
8
9 using namespace std;
10
11 class Uçgen {
12 private:
13     Nokta nokta1;
14     Nokta nokta2;
15     Nokta nokta3;
16
17 public:
18     // Üç tane Nokta nesnesi alan yapıcı.
19     Uçgen(const Nokta& n1, const Nokta& n2, const Nokta& n3);
20     const Nokta& getNokta1() const;
21     const Nokta& getNokta2() const;
22     const Nokta& getNokta3() const;
23     void setNokta1(const Nokta& n1);
24     void setNokta2(const Nokta& n2);
25     void setNokta3(const Nokta& n3);
26     // Üçgenin String temsili döndüren metod.
27     string toString() const;
28     // Üçgenin alanını hesaplayan metod.
29     double alan() const;
30     // Üçgenin çevresini hesaplayan metod. Üç doğru parçası oluşturarak DogruParcasi sınıfının uzunluk metodunu kullanır.
31     double cevre() const;
32     // Üçgenin açılarını hesaplayan ve üç ögeli double dizisi olarak döndüren metod.
33     double* acilar() const;
34 };
```

- **Üç Noktayı Alan Yapıcı (Uçgen(const Nokta& n1, const Nokta& n2, const Nokta& n3))**: Üç tane Nokta nesnesi alan yapıcı.
- **alan()**: Üçgenin alanını hesaplayan metod.
- **cevre()**: Üçgenin çevresini hesaplayan metod. Üç doğru parçası oluşturarak DogruParcasi sınıfının uzunluk metodunu kullanır.
- **acilar()**: Üçgenin açılarını hesaplayan ve üç ögeli double dizisi olarak döndüren metod.
- **toString()**: Üçgenin string temsili döndüren metod.

#### 4. SONUÇ VE ÖĞRENİLEN DERSLER

Öğrenilen Dersler:

Nesne tabanlı Programlama Prensipleri

C++ dilinde String işlemleri

Geometrik Problemlerin programlamayla çözümü

Her geometrik sınıf için başlık (**header**) ve implementasyon (**cpp**) dosyalarının ayrı olarak tutulmuştur.

Bu ödev, hem nesne tabanlı programlama konusundaki temel yetenekleri pekiştirmeye hem de matematiksel problemleri programlamayla çözebilme becerisini geliştirmeye yönelik önemli bir deneyim sağlamıştır.

Ödev No: 3	Tarih 01.01.2024	9/10
------------	------------------	------

---

Projede karşılaştığımız zorluklar ise, bizimle paylaşılan Test sınıfına uyarlama yapmak oldu. Test sınıfına uyarlamada, sınıfların içerdiği hesaplamaların doğruluğunu sağlamak ve hata ayıklama sürecini yürütmek zorlu oldu. Projenin büyümesiyle birlikte, kodun okunabilirliğini sürdürmek zorlu hale geldi. Bu zorluğun üstesinden gelmek için düzenli yorum satırları eklenerek, her sınıfın fonksiyonlarının ve önemli noktalarının açıkça belirtilmesi sağlandı.

## 5. GİTHUB HESAPLARI

<https://github.com/Seyma-lacin>

<https://github.com/Zehrayardimci>

## 6. KAYNAKÇA

<https://www.matematiktutkusu.com/forum/geometri-formulleri/698-tum-analitik-geometri-formulleri.html>

<https://www.geeksforgeeks.org/c-plus-plus/>

<https://cplusplus.com/forum/beginner/285698/>

<https://www.youtube.com/playlist?list=PLlrATfBNZ98dudnM48yfGUldqGD0S4FFb>

<https://www.youtube.com/watch?v=rWC7pEcixcg&list=PL8prT7s2xXGEmzP1Crdp7N4OOvzvW4Kds&index=35>

<https://www.youtube.com/watch?v=8A1csC5Y-G8>

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