



T.C

KOCaeli SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİęİ PROGRAMI

ÖDEV KONUSU

GEOMETRİK PROBLEMLER

Hazırlayanlar

Zeynep GÜNEŞ

220501018

Esmanur AYDIN

220501005

DERS SORUMLUSU

Prof. Dr. HÜSEYİN TARIK DURU

02/01/2024

[Zeynepgunss · GitHub](#)

[esm4ydn \(Esmanur AYDIN\) · GitHub](#)

İÇİNDEKİLER

1. ÖZET	3
2. GİRİŞ	3
3. YÖNTEM	3
4. SONUÇ VE ÖĞRENİLEN DERSLER	9
5. KAYNAKÇA	10

Ödev No:3	Tarih 02.01.2024	2/10
-----------	------------------	------

1. ÖZET

Bu ödevdeki C++ programı, Nokta, Doğru parçası, Daire, Üçgen sınıfları içermektedir. Bu sınıflar, geometrik şekiller ve noktaları temsil etmek için kullanılır ve bu sayede Geometri Programı oluşturulur. Bu program, geometrik şekillerin temsil edilmesi için faydalı sınıflar sağlar. Her sınıf, farklı geometrik hesaplamalar için uygun metotlara sahiptir.

2. GİRİŞ

Projemizin ana hedefleri şunlardır:

Geometrik şekilleri ve noktaları temsil eden sınıfları oluşturmak ve bunları kullanarak bir C++ programı ile belirli geometrik problemleri çözmektir. Sınıflar, geometrik şekillerin özelliklerini saklamamızı, hesaplamamızı ve bu şekiller arasında etkileşimleri kullanarak sonuçları çıktı almamızı sağlar.

3. YÖNTEM

Simülasyon, Python programlama dili kullanılarak obje tabanlı programlama prensipleri doğrultusunda tasarlanmıştır. Temel olarak, TIR ve gemi nesneleri oluşturulmuş ve bu nesneler arasında belirli kısıtlamalara ve sıralamalara uygun işlemler gerçekleştirilmiştir. Senaryoya özgü sıralama ve kısıtlamalar, program içinde kontrol yapıları ve algoritmalar aracılığıyla uygulanmıştır.

1. Nokta Sınıfı: X ve Y koordinat bilgilerini tutan bir noktayı temsil eder.
2. Doğru Parçası Sınıfı: İki noktayı birleştiren doğru parçasını temsil eder.
3. Daire Sınıfı: Merkez noktası ve yarıçap gibi bilgileri içeren direyi temsil eder.
4. Üçgen Sınıfı: Üç noktayı birleştirene bilgilerini içeren üçgeni temsil eder.

Bu sınıfların hepsi geometrik şekilleri için uygun bilgileri tutacak metotlarla yazılmıştır. Program, bu sınıfları kullanarak belirli geometri problemlerini çözmeyi amaçlar. Bu problemler şekillerin alanı, doğruluklarını kontrol etme, çevre hesabı ve şekiller arası ilişkileri belirleme gibi görevlere sahiptir. Bu özelliklerin sağlanması için kullanılan kütüphaneler:

1. **iostream**: Standart giriş ve çıkış işlemleri içindir.
2. **cmath**: Matematiksel fonksiyonlar içindir. ('sqrt', 'pow', 'cos', 'sin')
3. **string**: String işlemleri için kullanılır.
4. **vector**: Dinamik boyutta veri depolamak için kullanılır. ('std::vector')

Ödev No:3	Tarih 02.01.2024	3/10
-----------	------------------	------

Bunlar C++ için geometrik şekilleri ve noktaları temsil etmek ve matematiksel işlemleri gerçekleştirmek için yaygın olarak kullanılan kütüphanelerdir. Özellikle ‘cmath’ kütüphanesindeki trigonometrik fonksiyonlar veya geometrik hesaplamalar için matematiksel metotlar çok kolaylık sağlar.

KODLARIN AÇIKLAMASI:

Nokta(double x_val, double y_val); - Belirli değerlerle başlatan kurucu fonksiyon:

- Bu fonksiyon, sınıfın bir nesnesini oluşturur ve x koordinatını **x_val** ile, y koordinatını ise **y_val** ile başlatır.

Nokta(double val); - Varsayılan kurucu fonksiyon:

- Bu fonksiyon, sınıfın bir nesnesini oluşturur ve hem x hem de y koordinatlarını **val** değeri ile başlatır.

Nokta(const Nokta& other); - Kopya kurucu fonksiyon:

- Bu fonksiyon, başka bir **Nokta** nesnesinin kopyasını alır ve bu nesnenin x ve y koordinatlarını kullanarak yeni bir **Nokta** nesnesi oluşturur.

void setX(double x_val); - x koordinatını ayarlayan fonksiyon:

- Bu fonksiyon, sınıfın x koordinatını **x_val** değeri ile ayarlar.

double getX() const; - x koordinatını döndüren fonksiyon:

- Bu fonksiyon, sınıfın x koordinatını döndürür. Fonksiyonun sonundaki **const** ifadesi, bu fonksiyonun sınıfın durumunu değiştirmeyeceğini belirtir.

void setY(double y_val); - y koordinatını ayarlayan fonksiyon:

- Bu fonksiyon, sınıfın y koordinatını **y_val** değeri ile ayarlar.

double getY() const; - y koordinatını döndüren fonksiyon:

- Bu fonksiyon, sınıfın y koordinatını döndürür. Fonksiyonun sonundaki **const** ifadesi, bu fonksiyonun sınıfın durumunu değiştirmeyeceğini belirtir.

void set(double x_val, double y_val); - x ve y koordinatlarını belirli değerlerle ayarlayan fonksiyon:

- Bu fonksiyon, sınıfın hem x hem de y koordinatlarını sırasıyla **x_val** ve **y_val** değerleri ile ayarlar.

- **void yazdir() const;** - Koordinatları ekrana yazdıran fonksiyon:

- Bu fonksiyon, sınıfın x ve y koordinatlarını kullanarak bir string temsil oluşturur ve bu temsili ekrana yazdırır.

Ucgen(const Nokta& n1, const Nokta& n2, const Nokta& n3) : nokta1(n1),

Ödev No:3	Tarih 02.01.2024	4/10
-----------	------------------	------

nokta2(n2), nokta3(n3) {} - Üçgen Kurucu Fonksiyonu:

- Bu kurucu fonksiyon, üç noktayı (**nokta1, nokta2, nokta3**) alan bir **Ucgen** nesnesi oluşturur ve üçgenin kenar noktalarını bu noktalara ayarlar.

void setNokta1(const Nokta& nokta) { nokta1 = nokta; } ve Nokta getNokta1() const { return nokta1; } - Nokta1 Ayarlama ve Alama Fonksiyonları:

- Bu fonksiyonlar, **nokta1** üye değişkenini ayarlamak ve bu değeri almak için kullanılır.

std::string toString() const - Üçgenin String Temsilini Döndüren Fonksiyon:

- Bu fonksiyon, üçgenin string temsilini oluşturarak geri döndürür. Noktaların string temsilleri **Nokta** sınıfının **toString** fonksiyonları aracılığıyla alınır.

double alan() const - Üçgenin Alanını Hesaplayan Fonksiyon:

- Bu fonksiyon, üçgenin alanını hesaplamak için kenar uzunluklarını kullanır. **Nokta** sınıfının **mesafe** fonksiyonu ile üçgenin kenar uzunlukları elde edilir, ardından bu uzunluklar kullanılarak üçgenin alanı hesaplanır.

double cevre() const - Üçgenin Çevresini Hesaplayan Fonksiyon:

- Bu fonksiyon, üçgenin çevresini hesaplamak için **DogruParcasi** sınıfının **uzunluk** fonksiyonunu kullanır. Üçgenin her kenarının uzunluğunu alarak toplar ve bu şekilde üçgenin çevresini elde eder.

double* acilar() const - Üçgenin Açılarını Hesaplayan Fonksiyon:

- Bu fonksiyon, üçgenin açılarını hesaplamak için trigonometrik fonksiyonları kullanır. İlk önce üçgenin kenar uzunlukları alınır, ardından üçgenin açıları için cosinüs teoremi uygulanır. Sonuçlar derece olarak hesaplanır ve bir dinamik dizi içinde döndürülür.

DogruParcasi::DogruParcasi(const Nokta& nokta1, const Nokta& nokta2) : p1(nokta1), p2(nokta2) {} - Parametrel Kurucu Fonksiyon:

- Bu fonksiyon, **DogruParcasi** sınıfının parametrel kurucu fonksiyonudur. İki noktayı parametre olarak alır ve bu noktaları **p1** ve **p2** üye değişkenlerine atar.

DogruParcasi::DogruParcasi(const DogruParcasi& other) : p1(other.p1), p2(other.p2) {} - Kopya Kurucu Fonksiyon:

Ödev No:3	Tarih 02.01.2024	5/10
-----------	------------------	------

-
- Bu fonksiyon, **DogruParcasi** sınıfının kopya kurucu fonksiyonudur. Bir başka **DogruParcasi** nesnesinin kopyasını alır.

DogruParcasi::DogruParcasi(const Nokta& ortaNokta, double uzunluk, double egim) - Orta Noktadan ve Uzunluktan Dogru Parçası Oluşturan Kurucu Fonksiyon:

- Bu fonksiyon, bir nokta (**ortaNokta**), uzunluk (**uzunluk**) ve eğim (**egim**) parametrelerini alarak bir doğru parçası oluşturur. Bu doğru parçası, orta nokta etrafında belirtilen uzunluk ve eğim ile yerleştirilir.

void DogruParcasi::setP1(const Nokta& nokta) { p1 = nokta; } ve Nokta DogruParcasi::getP1() const { return p1; } - P1 Ayarlama ve Alma Fonksiyonları:

- Bu fonksiyonlar, **p1** üye değişkenini ayarlamak ve bu değeri almak için kullanılır.

double DogruParcasi::uzunluk() const - Dogru Parçasının Uzunluğunu Hesaplayan Fonksiyon:

- Bu fonksiyon, **p1** ve **p2** noktaları arasındaki doğru parçasının uzunluğunu hesaplar ve bu uzunluğu döndürür.

Nokta DogruParcasi::kesisimNoktasi(const Nokta& nokta) const - Dogru Parçasının Bir Noktayla Kesim Noktasını Hesaplayan Fonksiyon:

- Bu fonksiyon, **DogruParcasi** nesnesi ile verilen bir noktanın dik doğrusu arasındaki kesişim noktasını hesaplar ve bu noktayı döndürür.

Nokta DogruParcasi::ortaNokta() const - Dogru Parçasının Orta Noktasını Hesaplayan Fonksiyon:

- Bu fonksiyon, **p1** ve **p2** noktalarının orta noktasını hesaplar ve bu orta noktayı döndürür.

void DogruParcasi::yazdir() const - Dogru Parçasını Ekrana Yazdıran Fonksiyon:

- Bu fonksiyon, **toString** fonksiyonunu kullanarak doğru parçasının string temsilini oluşturur ve ekrana yazdırır.

Daire::Daire() : merkez(Nokta()), yaricap(0) {} - Varsayılan Kurucu Fonksiyon:

Ödev No:3	Tarih 02.01.2024	6/10
-----------	------------------	------

-
- Bu fonksiyon, **Daire** sınıfının varsayılan kurucu fonksiyonudur. Merkezi (**merkez**) varsayılan bir **Nokta** nesnesi olarak, yarıçapı (**yaricap**) ise 0 olarak başlatır.

Daire::Daire(const Nokta& merkez, double yaricap) : merkez(merkez), yaricap(yaricap) {} - Parametrelili Kurucu Fonksiyon:

- Bu fonksiyon, **Daire** sınıfının parametrelili kurucu fonksiyonudur. Merkezi ve yarıçapı parametre olarak alır ve bu değerlerle bir **Daire** nesnesi oluşturur.

Daire::Daire(const Daire& other) : merkez(other.merkez), yaricap(other.yaricap) {} - Kopya Kurucu Fonksiyon:

- Bu fonksiyon, **Daire** sınıfının kopya kurucu fonksiyonudur. Başka bir **Daire** nesnesinin kopyasını alır.

double Daire::cevre() const - Dairenin Çevresini Hesaplayan Fonksiyon:

- Bu fonksiyon, dairenin çevresini hesaplamak için 2π (pi) sayısını kullanarak **yaricap** ile çarpar ve sonucu döndürür.

std::string Daire::toString() const - Dairenin String Temsilini Döndüren Fonksiyon:

- Bu fonksiyon, dairenin merkezini ve yarıçapını içeren bir string temsil oluşturarak geri döndürür.

void Daire::yazdir() const - Daireyi Ekrana Yazdıran Fonksiyon:

- Bu fonksiyon, **toString** fonksiyonunu kullanarak dairenin string temsilini oluşturur ve ekrana yazdırır.

#ifndef DOGRUPARCASI_H ve #define DOGRUPARCASI_H:

- Bu iki satır, ön işlemci komutlarıdır. İlk satır, **DOGRUPARCASI_H** sembolünün tanımlanıp tanımlanmadığını kontrol eder. Eğer tanımlanmamışsa, bu ifade doğru olur ve dosyanın içeriği işlenir. Eğer tanımlanmışsa, dosyanın içeriği atlanır. İkinci satır ise, sembolü tanımlar, böylece bir sonraki dahil etme işleminden önce bu sembol zaten tanımlı olacaktır.

#include "Nokta.h":

- Bu satır, **Nokta** sınıfının tanımlanmış olduğu başka bir dosyanın (**Nokta.h**) içeriğini bu dosyaya ekler. Bu, **DogruParcasi** sınıfının **Nokta** sınıfını kullanabilmesini sağlar.

Nokta p1; ve Nokta p2;:

Ödev No:3	Tarih 02.01.2024	7/10
-----------	------------------	------

- Bu iki satır, **DogruParcasi** sınıfının özel (**private**) üye değişkenlerini tanımlar. **p1** ve **p2**, **Nokta** sınıfından nesneleri temsil eder.

DogruParcasi(); ve **DogruParcasi(const Nokta& nokta1, const Nokta& nokta2);**;

- Bu iki satır, sırasıyla varsayılan ve parametrelili kurucu fonksiyonları tanımlar. İlk fonksiyon bir doğru parçasını varsayılan değerlerle oluşturur, ikincisi ise iki nokta alarak bir doğru parçasını oluşturur.

void setP1(const Nokta& nokta);, **Nokta getP1() const;**, **void setP2(const Nokta& nokta);**, ve **Nokta getP2() const;**;

- Bu dört satır, **p1** ve **p2** üye değişkenlerini ayarlamak ve almak için olan fonksiyonları tanımlar.

double uzunluk() const;, **Nokta kesisimNoktasi(const Nokta& nokta) const;**, **Nokta ortaNokta() const;**;

- Bu üç satır, sırasıyla doğru parçasının uzunluğunu, bir nokta ile kesişim noktasını, ve doğru parçasının orta noktasını hesaplayan fonksiyonları tanımlar.

```
NOKTA SINIFI İÇİN TEST ÇIKTILARI:
Oluşturulan 5 noktanın x ve y değerleri:
Koordinatlar: (0,000000, 0,000000)
Koordinatlar: (5,000000, 5,000000)
Koordinatlar: (3,000000, 4,000000)
Koordinatlar: (3,000000, 4,000000)
Koordinatlar: (4,000000, 3,000000)
1. ve 2. noktaların güncel x ve y değerleri:
Koordinatlar: (10,000000, 0,000000)
Koordinatlar: (25,000000, 30,000000)

DOGRUPARCASI SINIFI İÇİN TEST SONUÇLARI:
Oluşturulan 3 doğru parçasının başlangıç ve bitiş noktaları:
P1: (0,000000, 0,000000), P2: (3,000000, 4,000000)
P1: (0,000000, 0,000000), P2: (3,000000, 4,000000)
P1: (-2,253220, -4,509035), P2: (8,253220, 12,509035)
2. doğru parçasının başlangıç ve bitiş noktaları:
P1: (0,000000, 0,000000), P2: (3,000000, 4,000000)
1. doğru parçası için uzunluk değeri: 5
1. doğru parçası için orta nokta: (1,500000, 2,000000)

DAIRE SINIFI İÇİN TEST SONUÇLARI:
Daire 1: Merkez: (0,000000, 0,000000), Yarıçap: 5,000000
Alan: 78.5398, Çevre: 31.4159
Daire 2: Merkez: (0,000000, 0,000000), Yarıçap: 5,000000
Alan: 78.5398, Çevre: 31.4159
Daire 3: Merkez: (0,000000, 0,000000), Yarıçap: 15,000000
Alan: 706.858, Çevre: 94.2478
Daire 1 ve Daire 2 Kesişim Durumu: 1
```

Ödev No:3	Tarih 02.01.2024	8/10
-----------	------------------	------

4. SONUÇ VE ÖĞRENİLEN DERSLER

Bu ödev, C++ programlama dili kullanarak nesne yönelimli programlama prensiplerini ve geometrik problemleri ele almayı hedefler. Bu ödevi tamamlarken öğrendiklerimiz:

- **Nesneye Yönelik Programlama (OOP):** Sınıflar ve nesneler arasındaki ilişkileri anlama ve uygulama, Sınıflarda erişim metotları kullanmanın kolaylıkları ve şekil, nokta gibi gerçek dünya nesnelerinin kullanımıyla sınıfların nasıl oluşturulacağını pekiştirdik.
- **Sınıfların Oluşturulması ve Kullanımı:** Sınıf yapıcıları, metotları, özellikleri(nesne), get ve set fonksiyonları ile çalışmayı, bu sınıfları başlatmayı, verileri değiştirmeyi ve işlemeyi pekiştirdik.
- **Geometrik Hesaplamalar:** Dairelerin çevresi ve alan gibi geometrik şekillerle ilgili temel hesaplamaları gerçekleştirme bununla birlikte, doğruların uzunluğu, kesişim noktaları gibi geometrik problemleri öğrenmemizi sağladı.
- **Kütüphane Kullanımı:** Standart C++ kütüphanelerinin kullanımının yanı sıra ‘cmath’ ve ‘vector’ kütüphanelerinin kullanımı, matematiksel fonksiyonların ve string işlemlerinin nasıl kullanılacağını anladık.
- **Metot kullanımı:** Metotları kullanarak sınıflar arasında etkileşim sağlamayı, bir nesnenin diğer nesneler ile ilişkilendirilerek nasıl kullanılacağını öğrendik.

Ödev No:3	Tarih 02.01.2024	9/10
-----------	------------------	------

5. KAYNAKÇA

- ❖ <https://chat.openai.com>
- ❖ <https://www.algoritmaornekleri.com/c-plus-plus/c-math-kutuphanesi/>
- ❖ <https://cod3xblog.wordpress.com/2015/01/24/c-hazir-sayisal-ve-sozel-fonksiyonlar/>
- ❖ <https://kerimfirat.medium.com/c-vektör-vector-kullanımı-d81378e8ed8>
- ❖ <https://mguludag.github.io/2020/07/16/cpp-vector-e-bakis/>
- ❖ <https://turkmuhendis.net/cppprogramlama/string-kutuphanesi-fonksiyonlari/>
- ❖ <https://slideplayer.biz.tr/slide/3627075/>

Ödev No:3	Tarih 02.01.2024	10/10
-----------	------------------	-------