

T.C
KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
LİSANS PROGRAMI

GEOMETRİK PROBLEMLER

Hazırlayan
EREN YALAZ 220502007
AHMET BURAK ÜNVER 220502004

DERS SORUMLUSU
Prof.DR.HÜSEYİN TARIK DURU

1.01.2024

İÇİNDEKİLER

1. ÖZET (ABSTRACT)
2. GİRİŞ (INTRODUCTION)
3. YÖNTEM (METHOD)
4. SONUÇ VE ÖĞRENİLEN DERSLER
5. KAYNAKÇA

1. ÖZET

Bu rapor, C++ programlama dili kullanılarak geliştirilen Nokta, DogruParcasi, Daire ve Ucgen sınıflarını içermektedir. Her bir sınıf, geometrik yapıları temsil eder ve farklı özelliklere sahiptir: Nokta sınıfı koordinatları tutar, DogruParcasi sınıfı doğru parçalarını temsil eder, Daire sınıfı daireleri ve Ucgen sınıfı üçgenleri yönetir. Her sınıfın yapısı, içerdiği özellikler ve bu özelliklerin nasıl kullanılabileceği raporda detaylı olarak incelenmiştir. Ayrıca her sınıf için örneklerle desteklenmiş kullanım senaryoları sunulmuştur. Bu rapor, geometriyle ilgili temel yapıların C++ programlama dili kullanılarak nasıl oluşturulabileceğini göstermektedir.

2. GİRİŞ

Bu rapor, geometrik yapıları temsil eden Nokta, DogruParcasi, Daire ve Ucgen sınıflarının C++ programlama dili kullanılarak nasıl oluşturulduğunu ve kullanıldığını ele almaktadır. Her bir sınıfın özellikleri ve işlevleri detaylı olarak incelenmiş ve örneklerle desteklenmiştir. Rapor, geometriyle ilgilenenler için pratik bir kaynak niteliği taşımaktadır.

3. YÖNTEM

Nokta Sınıfı Metotları:

Nokta(): Parametresiz yapıcı metot, x ve y koordinatlarını varsayılan olarak 0.0 olarak ayarlar.

Nokta(double deger): Tek parametrelili yapıcı, iki koordinata aynı değeri atar.

Nokta(double x_, double y_): İki parametrelili yapıcı, x ve y koordinatları için sırasıyla iki double değişken alır.

Nokta(const Nokta& digeri): Başka bir noktanın kopyasını alıp yeni bir nokta oluşturan yapıcı.

Nokta(const Nokta& digeri, double ofset_x, double ofset_y): Başka bir nokta ve iki double değişken alarak yeni bir nokta oluşturur ve ofset değişkenlerini orijinal noktanın x ve y koordinatlarına ekler.

DogruParcasi Sınıfı Metotları:

DogruParcasi(const Nokta& baslangic, const Nokta& bitis): İki noktayı alarak doğru parçası oluşturan yapıcı.

DogruParcasi(const DogruParcasi& digeri): Başka bir DogruParcasi nesnesinin kopyasını oluşturan yapıcı.

DogruParcasi(const Nokta& ortaNokta, double uzunluk, double egim): Nokta, uzunluk ve

eğimi olarak doğru parçası oluşturan yapıcı.

uzunluk() const: Doğru parçasının uzunluğunu hesaplayan metot.

kesisimNoktasi(const Nokta& digerNokta) const: Verilen bir noktadan doğru parçasına dik olarak çizilecek doğru parçasının kesişme noktasını hesaplayan metot.

ortaNokta() const: Doğru parçasının orta noktasını hesaplayan metot.

toString() const: Doğru parçasının başlangıç ve bitiş noktalarını string olarak döndüren metot.

setP1(const Nokta& nokta): Doğru parçasının başlangıç noktasını değiştiren metot.

Daire Sınıfı Metotları:

Daire(const Nokta& merkez, double r): Merkez ve yarıçapı olarak daire oluşturan yapıcı.

Daire(const Daire& digeri): Başka bir Daire nesnesinin kopyasını oluşturan yapıcı.

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

alan() const: Dairenin alanını hesaplayan metot.

cevre() const: Dairenin çevresini hesaplayan metot.

kesisim(const Daire& digeriDaire) const: Verilen bir Daire nesnesi ile kesişme durumunu kontrol eden metot.

toString() const: Dairenin merkezini ve yarıçapını string olarak döndüren metot.

Üçgen Sınıfı Metotları:

Üçgen(const Nokta& n1, const Nokta& n2, const Nokta& n3): Üç noktayı olarak üçgen oluşturan yapıcı.

toString() const: Üçgenin üç noktasını string olarak döndüren metot.

alan() const: Üçgenin alanını hesaplayan metot.

cevre() const: Üçgenin çevresini hesaplayan metot.

acilar() const: Üçgenin açılarını hesaplayıp dizi olarak döndüren metot.

4. SONUÇ VE ÖĞRENİLEN DERSLER

Bu rapor, C++ programlama diliyle Nokta, DogruParcasi, Daire ve Ucgen sınıflarını oluşturarak geometrik yapıları temsil etmeyi amaçladı. Öğrendiğimiz noktalar şunlar:

Nesne Yönelimli Programlama Uygulaması: Sınıfların oluşturulması ve bu yapıların geometrik hesaplamalar için nasıl kullanılabileceği.

Yapıcı Fonksiyonların (Constructor) Önemi: Farklı başlatma yöntemlerinin ve sınıflar arası iletişimin nasıl sağlanabileceğini anlamak.

Metotların Kullanımı: Her sınıfın özelliklerine yönelik metotların tanımlanması ve kullanılması.

Bu rapor, geometri ve nesne yönelimli programlamanın birleştirilerek, programlama prensiplerinin pratikte nasıl uygulanabileceğini gösterdi.

5. KAYNAKÇA

<https://www.yusufsezer.com.tr/cpp-class-ornekleri/>

<https://chat.openai.com>