

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Ders : **Veri Yapıları**
Dönem : **2021-2022 Güz Dönemi**

Ad ve Soyad : **Muhammet Kemal Güvenç**

Okul No : **B181210076**

Konu : **2. Ödev**

Şube : **1. Öğretim A Grubu**

Yazılan kodda değişkenler, metot adları ve sınıf adları kodun evrensel olması için İngilizce olarak yazılmıştır. Fakat kodları size en iyi şekilde anlatabilmek için yorum satırları Türkçe olarak yazılmıştır. Değişkenler küçük harflerle yazılmış eğer birden fazla kelimeden oluşuyorsa diğer kelimelerin baş harfleri büyük harfli olacak şekilde adlandırılmıştır. Sınıf ve fonksiyon adları ise bu kuraldan biraz farklı olarak baş harfleri de büyük olarak adlandırılmıştır.

Projede “PointNode”, “LineQueue”, “LineNode” ve “AVLTree” olmak üzere toplam 4 sınıf bulunmaktadır. Sınıflar, başlık dosyası ve kaynak dosyası olmak üzere 2 dosyaya yazılmıştır. Başlık dosyalarında sınıfların üye değişkenleri ve metot bildirimleri, kaynak dosyalarında ise metotların gövdeleri bulunmaktadır. Kod yazılırken elden geldiğince gereksiz bellek tüketiminden kaçınılmıştır.

PointNode sınıfı, içinde tuttuğu 3 kordinat bilgisi, noktanın orijine olan uzaklığı (priority) ve ilerisinde olan düğümün adresi olmak üzere 5 üye değişkene sahiptir. Bir de noktanın orijine olan uzaklığını hesaplayan bir metot ve yapıcı metot bulundurmaktadır. Ayrıca “<<” ve “>” opertörleri bu sınıf için aşırı yüklenmiştir. “<<” operatörü ile ekrana o düğümün orijine olan uzaklığı yazılmaktadır. “>” operatörü ile ise iki noktanın orijine olan uzaklıkları kıyaslanmaktadır. PointNode sınıfından bir nesne oluşturulurken kordinat bilgilerinin verilmesi şart koşulmuştur. Düğümün ilerisi ise otomatik olarak NULL’a eşitlenir ve orijine olan uzaklığı hesaplanıp saklanır.

LineQueue sınıfı, biri doğru kuyruğunun başını diğeri sonunu tutan 2 adet gösterici üye değişkenine sahiptir. Bir doğru kuyruğu oluşturulurken kuyruğun başı ve sonu otomatik olarak NULL’a eşitlenir. Buna karşılık yapıcı ve yıkıcı metotları hariç olmak üzere 8 adet metodu bulunmaktadır. Bu metotlar şu şekildedir:

PreviousPriorityNode: Doğru kuyruğunda kendisine gönderilen noktanın önceliğinden bir tık daha fazla önceliğe sahip düğümü bulur ve döndürür.

DistancePoints: Kendisine gönderilen 2 noktanın arasındaki uzaklığı hesaplayıp döndürür.

Enqueue: Kendisine gönderilen koordinatların değerine sahip bir nokta oluşturur. Sonra oluşturduğu bu noktanın orijine olan uzaklığıyla ters orantılı olan önceliğine göre kuyruğa ekler.

Dequeue: Doğru kuyruğundan önceliği en yüksek olan (orijine en yakın olan) noktayı çıkarır.

Peek: Doğru kuyruğunda önceliği en yüksek olan (orijine en yakın olan) noktayı döndürür.

IsEmpty: Doğru kuyruğunun boş olup olmadığını döndürür.

TotalLength: Doğru kuyruğunun toplam uzunluğunu hesaplayıp döndürür.

Clear: Doğru kuyruğundaki tüm noktaları temizler.

Bunların dışında “<<”, “>” ve “<=” operatörleri bu sınıf için aşırı yüklenmiştir. “<<” operatörü ile doğru kuyruğundaki tüm noktaların orijine olan uzaklıkları ekrana yazılır. “>” ve “<=” operatörleri ile ise iki doğru kuyruğunun toplam uzunlukları karşılaştırılır.

LineNode sınıfı ağacın düğümlerini temsil eder. Bu sınıfta 3 tane üye değişkeni bulunmaktadır. Bunlar düğümün içinde tuttuğu doğru kuyruğu ile bu düğümün sol ve sağ çocuklarının göstericileridir. Bu sınıftan yeni nesne oluşturulduğunda içine yerleştirilecek olan doğru kuyruğunun verimesi şart koşulmuştur. Çocuk düğümleri ise otomatik olarak NULL’a eşitlenir. Yapıcı ve yıkıcı metotları dışında metodu bulunmamaktadır.

AVLTree sınıfı, ağacın sadece kökünü gösteren bir gösterici üye değişkenine sahiptir. Ayrıca yapıcı ve yıkıcı metotları dışında toplam 7 adet metodu vardır. Bu metotlar şu şekildedir:

TurnRight: Ağaca eleman eklenirken ağacın dengesi sola doğru bozulursa ağacı sağa döndürmeye yarar.

TurnLeft: Ağaca eleman eklenirken ağacın dengesi sağa doğru bozulursa ağacı sola döndürmeye yarar.

Add: Aslında bu metottan 2 tane var. Bunlardan sadece biri kullanıcıya açılmıştır. Asıl işi kullanıcıya açılmayan metot yapmaktadır. Bu metoda ağaca eklenmesi istenen doğru kuyruğu ile ağacın kök düğümü verilir. Bu metot recursive özelliğini kullanarak verilen doğru kuyruğunu kuyruğun toplam uzunluğuna göre ağacın uygun yerine yerleştirir. Ayrıca bunu yaparken de eğer ağacın dengesi bozulursa onu da düzeltir.

Height: Verilen düğümün yüksekliğini hesaplayıp döndürür.

Clear: Ağacı postorder olarak gezip bütün düğümlerini siler.

PostOrder: Ağacı postorder olarak gezip her bir düğümün içindeki doğru kuyruğunu ekrana yazar. Doğru kuyruğunun ekrana yazılması ise içlerindeki noktaların orijine olan uzaklıklarının ekrana yazılmasıdır.

Bunların dışında “<<” operatörü aşırı yüklenmiştir. PostOrder metodu kullanıcıdan saklanıp bu metodun kullanılabilmesi için “<<” operatörü aşırı yüklenilmiştir. Aslında bu operatör kullanılırken arkada PostOrder metodu çağırılıyor.