

Ödev Raporu

Giriş

Bu ödevde, ikili ağaçları bağlı liste içerisinde yönetmek için bir program geliştirdim. Programda, her bir satırda bir ikili ağacın yer aldığı bir dosyadan (agaclar.txt) veriler okundu ve bu verilerle ikili ağaçlar oluşturuldu. Daha sonra, kullanıcıya bağlı liste içerisindeki ağaçlarla çeşitli işlemler yapma imkânı verildi. Bu işlemler arasında ağacı aynalama, düğüm silme, önceki ve sonraki düğüme gitme gibi işlemler bulunmaktadır.

Öğrendiklerim

Bu ödevde, ikili ağaçların ve bağlı listelerin temel işlevlerini öğrendim. Ayrıca, programda yer alan kullanıcı etkileşimi ve menü yapısı ile ilgili deneyim kazandım. Programın temel işlevleri şunlardır:

- **Bağlı Liste:** Bağlı liste kullanarak ağaçları depoladım. Bu liste üzerinde ilerlemek için çeşitli fonksiyonlar (movePrev, moveNext) kullandım.
- **İkili Ağaçlar:** Her bir satırdaki karakterlerden bir ikili ağaç oluşturulmasını sağladım. Her karakter, ikili ağacın düğüme ekleniyor.
- **Dosya Okuma ve Yazma:** Verileri dosyadan okumayı ve ekrana yazmayı öğrendim. Dosya okuma işlevi, her satırdaki verileri işleyerek ikili ağaç oluşturulmasına olanak sağladı.

Yapılanlar

1. **Bağlı Liste Yapısı:** Bağlı listeyi kullanarak, her bir ağaç için bir düğüm ekledim ve bu düğümler üzerinde hareket edebilmek için çeşitli fonksiyonlar geliştirdim.
2. **İkili Ağaç Yapısı:** Her satırdaki karakterleri ikili ağaç yapısına ekledim. Bu, her bir karakterin ağaca eklenmesini sağlayarak ikili ağaçları oluşturdu.
3. **Kullanıcı Etkileşimi:** Kullanıcıya bağlı liste üzerinde işlem yapma imkânı sundum. Kullanıcı ağaca ayna yapma, düğüm silme, önceki veya sonraki düğüme gitme gibi işlemleri gerçekleştirebiliyor.
4. **Dosya Okuma:** Veriler "agaclar.txt" dosyasından okunarak işleme alındı.

Zorlanılan Kısımlar

- **Türkçe Karakter Desteği:** C++ programında Türkçe karakterlerin doğru şekilde yazılması ve görüntülenmesi konusunda bazı zorluklar yaşadım. Bunun için doğru locale ayarlarını yapmam gerekti.
- **Bağlı Liste İşlemleri:** Bağlı liste üzerinde düğüm ekleme, silme ve geçiş yapma işlemleri bazı karmaşıklıklar içeriyordu. Listeyi düzgün bir şekilde güncellemek ve kullanıcı etkileşimini sağlamak zaman alıcı oldu.

Eksik Bırakılan Kısımlar

- **Hata Yönetimi:** Dosya okuma ve kullanıcı etkileşimi sırasında daha kapsamlı hata yönetimi ekleyebilirdim. Örneğin, geçersiz karakter girildiğinde daha kullanıcı dostu hata mesajları verilebilir.
- **Daha Detaylı Çıktı:** Programın çıktısını daha estetik ve anlaşılır hale getirebilirdim. Bu, kullanıcı deneyimini iyileştirebilirdi.

Sonuç

Bu ödev, ikili ağaçlar ve bağlı listeler konusundaki bilgimi pekiştirmemi sağladı. Ayrıca, kullanıcı etkileşimi ile ilgili deneyim kazandım. Programın temel işlevleri çalışıyor olsa da, daha fazla özellik eklenebilir ve hata yönetimi geliştirilebilir. Bu ödev sayesinde C++ dilindeki veri yapılarını daha iyi anlama fırsatı buldum.