

ÖDEV#2 + ÖDEV#3 - METİN YAZARI BELİRLEME
(KELİME KULLANIM SIKLIKLARINI HESAPLAMA)

Senaryo

Metnin yazarının belli olmadığı veya bir başkasına atfedildiği durumlarda metnin yazarını belirlemenin bir yolu kelime sıklıklarını kullanmaktır. Bir X yazarının A metnini yazdığı biliniyorsa ve incelenen bir B metnindeki kelime sıklıklarının dağılımı A'daki sıklıklara çok yakınsa, o zaman B'nin yazar X tarafından yazılmış olması muhtemeldir. Bu yöntemin edebi çalışmalar için ne kadar güvenilir olup olmadığına bakmaksızın, bu yöntemi kullanan programı yazmak istiyoruz. Bu programın bir metin dosyasını taraması ve bu dosyadaki kelimelerin sıklığını hesaplaması beklenmektedir.

Gereksinimler

C# programlama dili ile geliştirilecek Windows Forms uygulaması ile aşağıdaki işlevlerin yerine getirilmesi istenmektedir:

- Form üzerinde bulunan “Metin Yükle” butonu tıklandığında “OpenFileDialog” sınıfı kullanılarak UTF8 kodlama formatı ile oluşturulduğu varsayılan bir “.txt” uzantılı dosyanın seçilmesi sağlanacaktır. Dosya yolu (*path*) form üzerinde bir metin kutusunda gösterilmelidir.
- Form üzerinde bulunan “Stack’e Oku” butonunun tıklanması ile seçilen bu dosyadaki her bir cümlelerin bağlı liste ile yığıt (*stack*) veri yapısına alınması sağlanmalıdır. Tüm cümleleri ile bir yığıt dizisinde tutabilirsiniz.
- Okuma işleminden sonra da dosyaya ait birtakım özellikler (içerdiği toplam cümle sayısı, her bir cümledeki kelime sayısı ve ortalama kelime sayısı) hakkında bir form kontrolü üzerinde bilgi verilmelidir.
- Her bir cümlelerin alt alta satırlarda bulunduğu, bir cümle içerisindeki her bir kelimenin boşluk karakteri ile diğer kelimelerde ayrıldığı varsayılmaktadır.
- Kelimenin orjinal biçimi, pozisyon bilgisi (metin içerisinde kaçınıcı cümle, cümle içerisinde kaçınıcı kelime), kelimenin ön işleme (*preprocessed*) yapılmış biçimi ve kullanım sıklığı (*frequency*) bilgileri bilinmelidir.
- “Ağaca Aktar” butonu ile de kelimeler tutulan yığıt dizisinden ağaç veri yapısına aktarılmalıdır. Ağaçtaki her bir düğümde kelime ve kelime kullanım sayısı ile birlikte

kelimenin diğer bilgileri uygun bir veri yapısı kullanılarak tutulabilir. Ağacın heap ağacı olması beklenmektedir.

- “En Sık Kullanılan Kelimeleri Getir” butonu ile de heap ağacı üzerinde heap sort uygulanmalı ve butonun yanındaki metin kutusunda yazan kelime sayısı kadar en sık kullanılan kelimeler ekranda uygun bir form kontrolünde gösterilmelidir.
- “Hash Tablo Oluştur” butonu ile kelimeleri ya da kelimeye ait uygun bir bilgiyi anahtar olarak hash’leyerek hash tablo oluşturunuz. Ayrık zincirleme yöntemi ile çakışmaları engelleyiniz.
- Kelimenin başında ve sonundaki noktalama işaretleri (*punctuation characters*) dikkate alınmamalıdır. C#’ta `Char.IsPunctuation` metodu kullanılabilir. Dosyanın yalnızca harf içeren sözcüklerden oluştuğu varsayılmalıdır.
- Kelime içerisinde yer alan noktalama işaretleri (kesme, tırnak vs.) ve bu işaretler sonrası karakter dizilimleri dikkate alınmamalıdır. Örn: Okan’dan -> Okan
- Metin içerisindeki noktalama işaretleri dikkate alınmaz ve büyük/küçük harf duyarlılığı devre dışı bırakılır. Her bir kelimeyi küçük harfe dönüştürerek normalize edebilirsiniz.
- Formun tasarımı, problem için önerilen çözüm senaryonuza uygun bir şekilde tasarlanmalıdır. Kontroller rastgele form üzerinde dağıtılmamalıdır.

Diğer Bilgiler

- Uygulamada bağlı listeler ile yığıt gerçekleştirimi, ağaç veri yapısı kullanımı, heap sort sıralama algoritması ve hash tablo veri yapısı kullanmak gerekmele birlikte **bunların yanında** istenilen bir veri yapısı de ek olarak kullanılabilir.
- Yardımcı kütüphane ve fonksiyonlar kullanılabilir.
- Bir kaynak kullanılmışsa ilgili kod bloğunun üstüne ya da uygun bir yere kaynağın referansını yorum satırı olarak ekleyip alıntılایınız.
- Kod içerisinde önemli fonksiyonların üstüne yorum satırı olarak fonksiyon işlevini açıklayan bilgilendirmeler eklemelisiniz.
- Nesne yönelimli programlama özellikleri kullanılmalıdır.
- Ödev bireysel olarak yapılmalıdır.

Değerlendirme (200 puan, ders notuna etkisi 18 puandır)

- Dosya okuma ve dosya bilgileri gösterme 20 puan
- Bağlı Liste ile Yığıt veri yapısı gerçekleştirimi 30 puan
- Ağaç veri yapısı kullanımı 40 puan
- Heap sort sıralama algoritması kullanımı 30 puan
- Hash tablo veri yapısı ile oluşturma ve çıkarma önleme 50 puan
- Nesne yönelimli programlama ve programın çalışması 30 puan

Teslim Zamanı

- **27 Mayıs 2022 Cuma, 23.59**
- Geç teslim mümkün DEĞİLDİR.

Teslim Ortamı

- Yüklemeler Github ortamında gerçekleştirilecektir. GitHub'a çözümünüzü **sıkıştırarak yüklemeyiniz (sıkıştırırsanız -20puan)**. Git komutlarını kullanınız.
<https://classroom.github.com/a/FDBzE1Lh>
- UBYs'ye yükleme yapmayınız.

Dürüstlük (İntihal)

Yazılım benzerliği tespitinde benzerlik eşiğini aşan tüm teslimler sıfır alacaktır. Dürüst olmayan bir teslim yerine ödevinizi teslim etmemenizi öneririm.

Sorular

Ödevle ilgili sorularınız için lütfen Teams platformu altında yapılmış ödev duyurusu ile ilgili gönderilen iletinin altındaki yorumlar bölümünü kullanınız. Sorunuzu sormadan önce, lütfen önceki soruları ve cevapları dikkatlice kontrol ediniz.

E-posta ya da başka bir platformdan gönderilen sorulara cevap verilmeyecektir!

BELİRSİZ bir şey bulursanız, lütfen bunu AYDINLATMAK için yardım isteyin!

Başarılar!!!