

Konu: Nesneye Dayalı Programlamanın Temel Yapıları ve JUnit Kullanımı

Soru:

Bu proje kapsamında aşağıda temel gereksinimleri verilen bir metin editörü (text editor) uygulamasının Java dili ile gerçekleştirimi yapılacaktır.

Temel Gereksinimler:

İkinci aşaması ileride Proje-2 kapsamında tanımlanacak olan uygulamanın Proje-1 kapsamında yalnızca aşağıdaki özelliklere sahip olması yeterlidir:

- * **Dosya oluşturma, açma, kapatma, kaydetme;**
- * **Verilen bir kelimeyi bulmak ve değiştirmek;**
- * **Yazılan harfleri geri almak (harf harf (tek tek) geri alınabilir);**
- * **Yazım Hatası Kontrolü ve düzeltilmesi:** Aşağıdaki açıklamaları okuyunuz!

Metin (İngilizce dilinde olacaktır) içinde kelimeler nokta, virgül, noktalı virgül, boşluk gibi ayraçlarla birbirinden ayrılmış olabilir. Metin içinde sayısal değer geçerse (28 gibi) kontrol etmeye gerek yoktur, doğru kabul edilecektir. Metin içindeki bir kelime – ile ayrılmış iki parçadan (full-text gibi) oluşabilir.

Metin içindeki bir kelime eğer size verilen "words.txt" dosyasında yer alan sözlükte var ise doğru olarak kabul edilecektir. Metin içindeki bir kelime eğer size verilen sözlükte yok ise hatalı kabul edilecektir. Bu durumda **sadece** "single transposition" olarak adlandırılan ve iki harfin yer değiştirmesinden kaynaklı yazım hatalarının düzeltilmesi istenmektedir. Bunun için tüm "single transposition" ları bularak sözlükte olup olmadığını kontrol edebilirsiniz. Örneğin; default kelimesi için "single transposition" lar, e ve d yer değiştirmişse edfault, e ve f yer değiştirmişse dfeault, f ve a yer değiştirmişse deafult, vb. gibi hatalı yazımlar olmuş olabilir. Yukarıda belirtildiği gibi sadece bir çeşit (single transposition) yazım hatası düzeltme özelliği bu proje için yeterlidir ve "words.txt" dosyasını da ödev klasöründe bulabilirsiniz.

Geliştireceğiniz editör Microsoft Windows ortamındaki "Notepad" uygulamasına benzemekle birlikte daha az fonksiyona sahiptir ve daha sadedir. Örneğin, yazıcı çıktısı alma gibi detaylı işlevlere gerek yoktur.

Geliştirdiğiniz uygulamaya ilişkin bir işlevselliği (veya bir senaryoyu) gerçekleştirirken, JUnit kullanarak birim testlere dayalı geliştiriniz. Her ne kadar "Extreme Programlama" nın gerçek hayat uygulamalarında tüm uygulamanın birim testlere dayalı olarak geliştirilmesi gerekse de, nesneye dayalı programlamanın diğer taraflarına daha çok odaklanabilmeniz için bu proje kapsamında geliştirdiğiniz uygulamanın tümünü birim testler ile gerçekleştirmenize gerek yoktur.

Konu: Tasarım Desenleri**Soru:**

Birinci projede geliřtirmiş olduđunuz metin editörü (text editor) uygulamasının tasarım desenleri (design patterns) kullanılarak yeniden yapılandırılması (yeni bir versiyon geliřtirilmesi) beklenmektedir.

Temel Gereksinimler:

- 1) Geri alma işlevselliđini "Command" tasarım deseni kullanarak gerçekteřtirez.
- 2) Koleksiyon dolařma gereken yerlerde Java ortamında hazır olarak bulunan "Iterator" deseni gerçekteřtirimini kullanınız.
- 3) "Singleton" dıřında iki adet tasarım deseni daha kullanınız. Bu iki adet desen; ders kitabında olanlardan, ders kitabı dıřında derste anlatılanlardan veya bunların dıřındaki desenlerden (GoF desenleri katalogunda olması ve ne için nasıl kullandığınızın raporda anlatılması řartı ile) seçilebilir.
- 4) JUnit kullanımına gerek yoktur.