

NESENEYE DAYALI PROGRAMLAMA

Proje 2

Egemen Aydın 05170000078

Sercan Bayram 05170000038

Oğuz Türk 05170000079

Gerçekleştirilen Platform ve Sürüm Adı:

Netbeans 12.2 ve UML için Netbeans 8.1

Programın Tanımı:

Bu projede birinci projede geliştirmiş olduğumuz metin editörü (text editor) uygulamasının tasarım desenleri (design patterns) kullanılarak yeniden yapılandırılması (yeni bir versiyon geliştirilmesi) beklenmektedir.

Temel Gereksinimler:

1) Geri alma işlevselliğinin "Command" tasarım deseni kullanarak gerçekleştirilmesi.

2) Koleksiyon dolaşma gereken yerlerde Java ortamında hazır olarak bulunan "Iterator" deseni gerçekleştiriminin kullanımı.

3) "Singleton" dışında iki adet tasarım desenin daha kullanılması. Bu iki adet desen; ders kitabında olanlardan, ders kitabı dışında derste anlatılanlardan veya bunların dışındaki desenlerden (GoF desenleri kataloğunda olması ve ne için nasıl kullandığının raporda anlatılması şartı ile) seçilebilir.

Bu rapor Proje-1’de oluşturulmuş olan tüm sınıfları, örnekleri vs. kapsamaktadır.

SINIFLAR

Content Sınıfı

Content sınıfında oluşturulan “words” arraylisti bize verilmiş olan “words.txt” dosyasındaki kelimeleri sözlük gibi tutar, “metin” stringi de programın ana ekranında bulunan Text Area içeriğini tutar. Ardından content sınıfında bir “add” metodu bulunur. Bu metotta "kelime" değişkeninin noktalama işaretlerinden arındırılmış string olması beklenmektedir. Bu string ile word nesnesi oluşturulduktan sonra bu nesne metine eklenir. Content sınıfında bulunan bir başka metod olan “wordVer” metodu programın geliştirilmesi sırasında sözlüğü görselleştirmek amacıyla oluşturulmuştur.

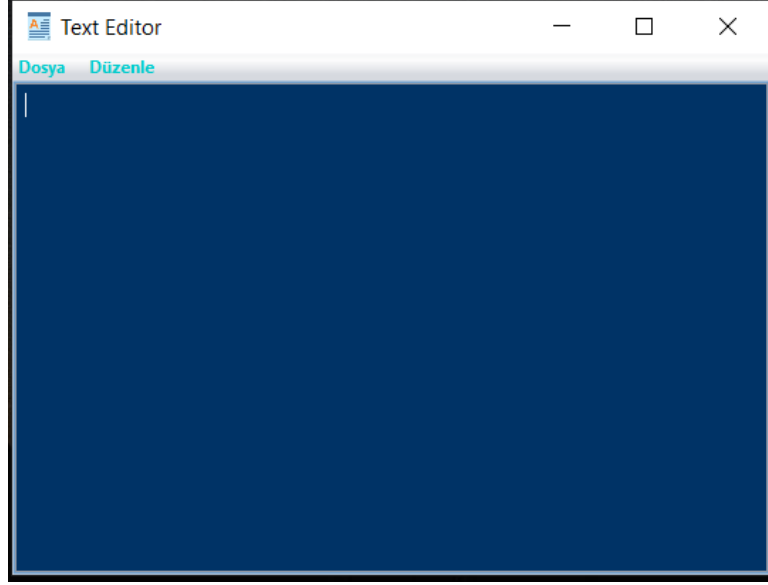
Content sınıfının içinde bir inner sınıf olan “Word” sınıfı bulunur. Word sınıfı string girdisinin word olmadan önce belirli kontrollerden geçmesini sağlar.

“SetWord” metodunda bu kontroller gerçekleşir. Kelime words'de (sözlükte) bulunursa word nesnesi oluşturulur bulunamaz ise iki harf değiştirerek elde edilen türevlerinden words içinde olanı aranır. Hiçbir türevi sözlükte yoksa ilk hâli ile nesne oluşturulur.”SetWord” metodundaki kontrolleri gerçekleştiren “ikiHarfDegistir” ve “varMi” metotları da yine bu sınıfta yer alır. “ikiHarfDegistir” metodunda her harfin bir sağındakiyle yer değiştirmesi ile elde edilen tüm kelimeler arama fonksiyonuna (varMi) gönderilir. Bir arama fonksiyonu olan “varMi” metodunda da .txt dosyasındaki kelimeler alfabetik sırada olduğu için ikili arama(binary search) kullanımı uygun görülmüştür.

NewJFrame Sınıfı

Programın görselleşmesinin gerçekleştiği temel JFrame sınıfıdır. Content sınıfında da bahsi geçen Text Area, menü ve butonlar bu sınıfta bulunmaktadır. “NewJFrame” sınıfında “kelimeParcala” adında bir metot bulunur. Test Area'daki tüm metin bu fonksiyona gelmektedir. Metin boşluk, noktalama işaretleri veya /n ile karşılaşılıncaya kadar okunur. Karakterlerden biri bulunduğunda okunmuş olan kısım ve karşılaşılan karakter birbirinden bağımsız şekilde Content nesnesine eklenir. Henüz okunmamış olan kısım tekrar bu fonksiyona gönderilerek 'recursive' şekilde 'split' işlemi gerçekleştirilmiş olur. Bu sınıfın devamında dosya açmak, dosya kaydetmek, bir kelimeyi başka bir kelimeye dönüştürmek için bulunan JFrame’ler oluşturulur.

Ardından gelen dosya düzenleme özelliğinde de düzenleme fonksiyonu kullanıldıktan sonra Text Area'daki tüm metin “kelimeParcala” fonksiyonundan geçtikten sonra tekrar Text Area'ya yazılır. Son olarak da Text Area'daki metnin en sonundaki karakter silinerek son harfi silme özelliği de programa eklenmiş olur.



NewJFrame'de oluşturulan program arayüzü

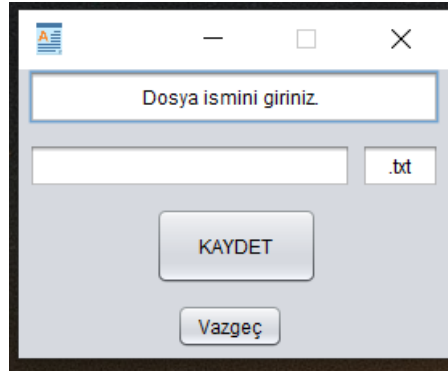
degistirmeForm Sınıfı

Ana menüde bulunan kelime değiştirme özelliğinin kullanılabilmesi için kullanılan formdur. Form üzerinden kaynak kelime, hedef kelime ve büyük harf duyarlılığı için boolean bilgileri alınır. Text Area'dan okunan değerde gerekli değişiklik yapıp tekrar Text Area'ya yazılır.

Kelime Değiştirme Ekrani

kayitForm Sınıfı

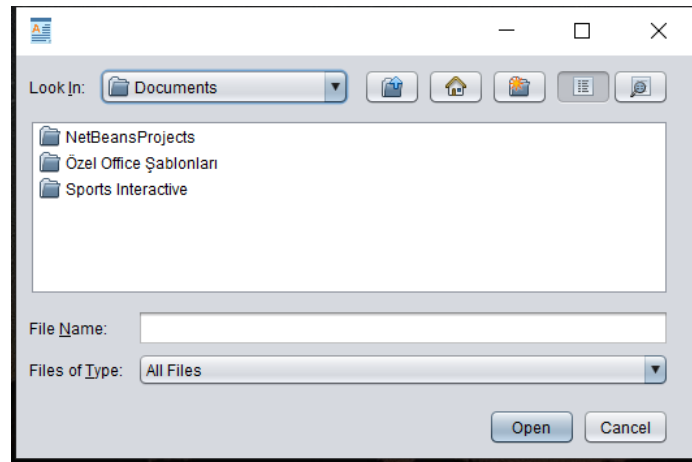
Ana menüde bulunan dosyayı kaydet özelliğinin kullanılabilmesi için kullanılan formdur. Kaydet butonuna basılmasının ardından Text Area'dan okunan path'e '.txt' uzantısı ile dosya yazma işlemi yapılır. Dosyanın hangi isimle kaydedileceği de kullanıcı girdisiyle belirlenir.



Dosya Kaydetme Ekranı

txtAcmaFormu Sınıfı

Ana menüde bulunan dosya açma özelliğinin kullanılabilmesi için kullanılan formdur. Kullanıcı kendi bilgisayarından açmak istediği dosyayı seçer. Okunan bir dosya varsa tüm içeriğini Text Area'ya yazar.



Dosya Açma Ekranı

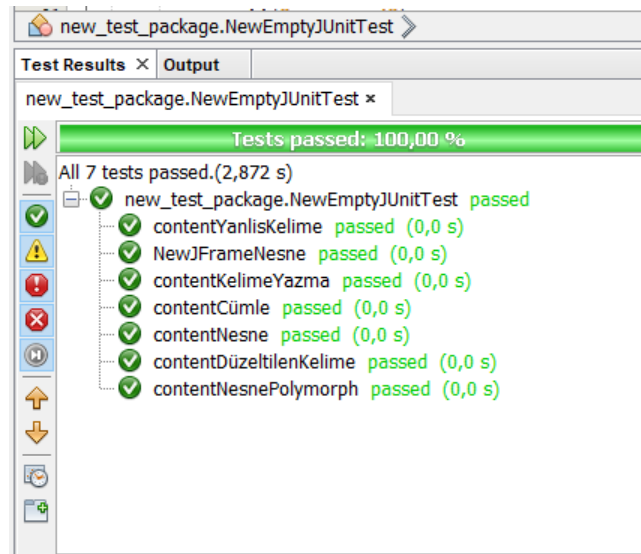
NewMain Sınıfı

Uygulamanın çalıştırılacağı main sınıftır. Proje dizinindeki words dosyası okunarak bir arrayList'e yazılır. Bu array list "Content" nesnesi üzerindeki words'e yazılır. Ana sayfa görevi gören JFrame oluşturulur.

NewEmptyJUnitTest Sınıfı

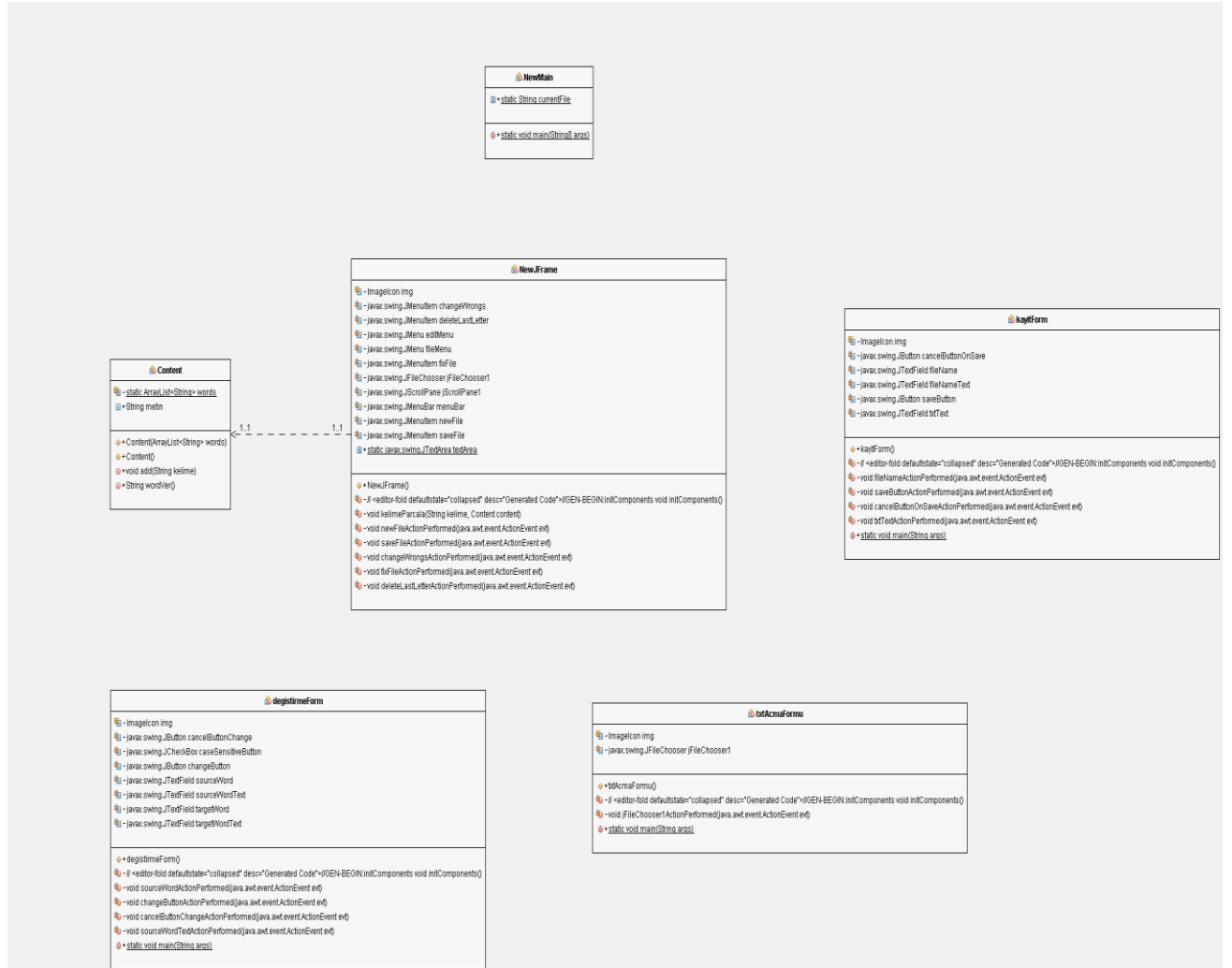
Uygulama geliştirilirken kullanılan test sınıfıdır. Bu sınıfta 7 adet test metodu bulunmaktadır.

- “NewJFrameNesne” test metodu NewJFrame sınıfından oluşturulan bir nesnenin oluşup oluşmadığını test eder.
- “contentNesne” test metodu Content sınıfından oluşturulan bir nesnenin oluşturulup oluşturulmadığını test eder.
- “contentNesnePolymorph” test metodu Content sınıfından oluşturulan bir arraylist nesnesinin oluşturulup oluşturulmadığını test eder.
- “contentKelimeYazma” test metodu Text Area’ya eklenen bir metnin doğru yazıldığı durumu test eder.
- “contentYanlisKelime” test metodu Text Area’ya eklenen bir metnin yanlış yazıldığı durumu test eder.
- “contentDüzeltilenKelime” test metodu Text Area’da bulunan bir yanlış kelimenin düzeltilme durumunu test eder.
- “contentCümle” test metodu Text Area’ya eklenen kelimeler bütünü aralarında boşluklar bulunduğu için bir cümle gibi test eder.



Oluşturulan JUnit Testlerinin Sonucu

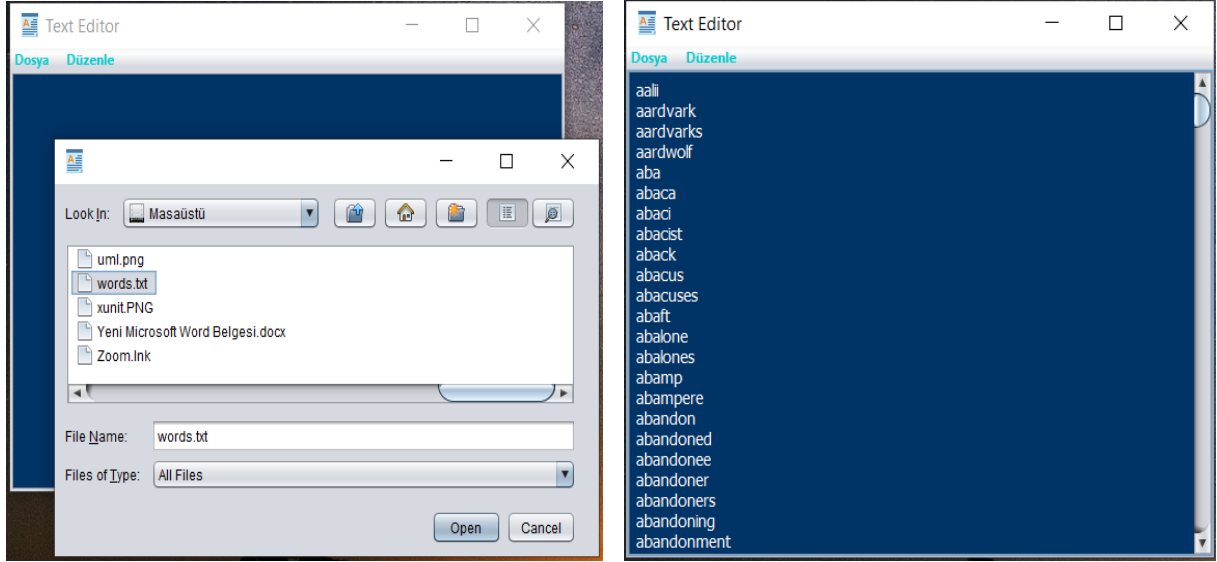
UML Diyagramı



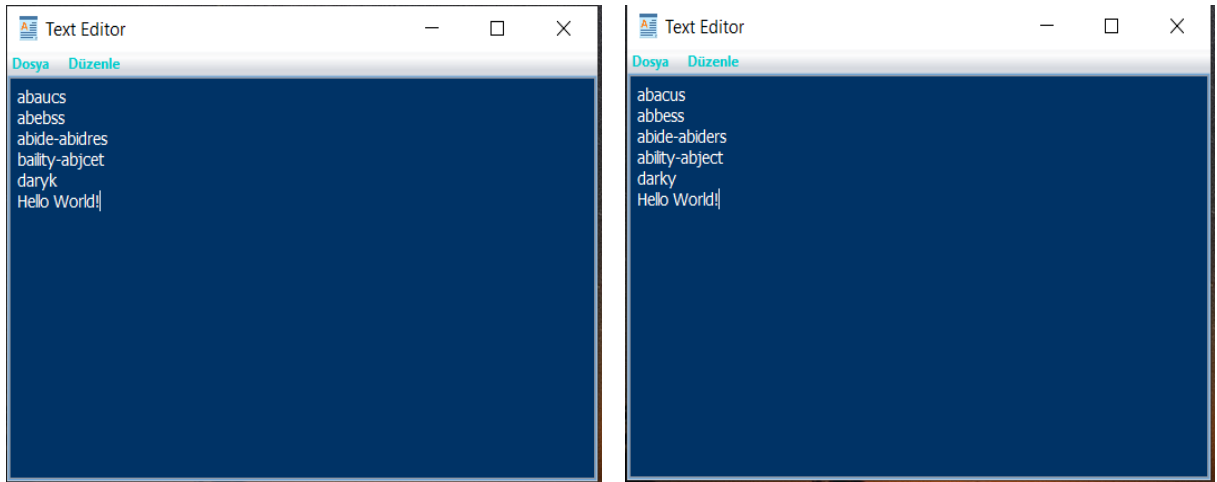
Netbeans ortamında bulunan easyUML plugini ile oluşturulan UML diyagramı

UML diyagramında projede bulunan tüm sınıflar, değişkenler ve metotlar görünmektedir. Ayrıca “NewJFrame” ve “Content” sınıfları arasındaki bağıllık ilişkisi de şekilde gösterilmiştir.

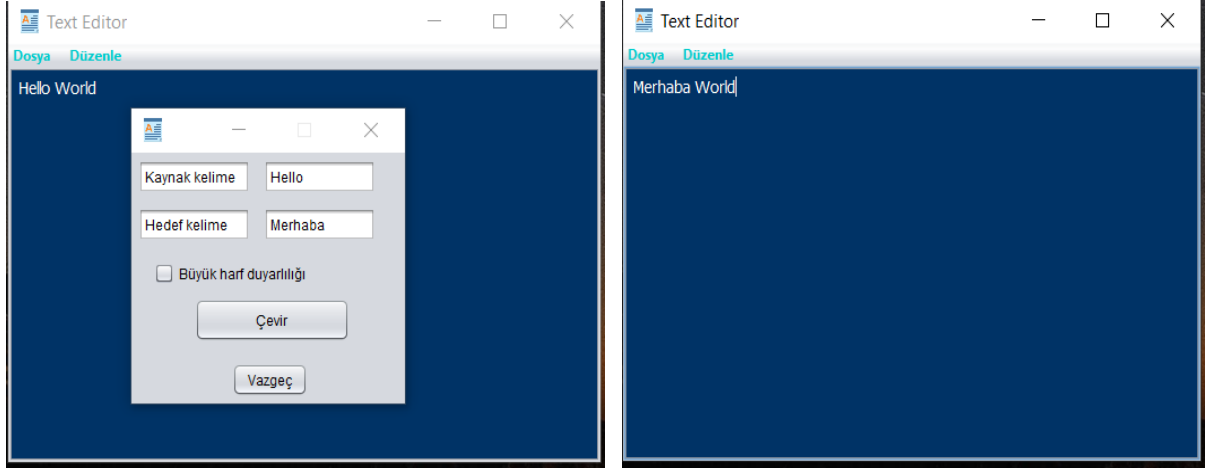
Programın Kullanımıyla İlgili Bazı Örnekler



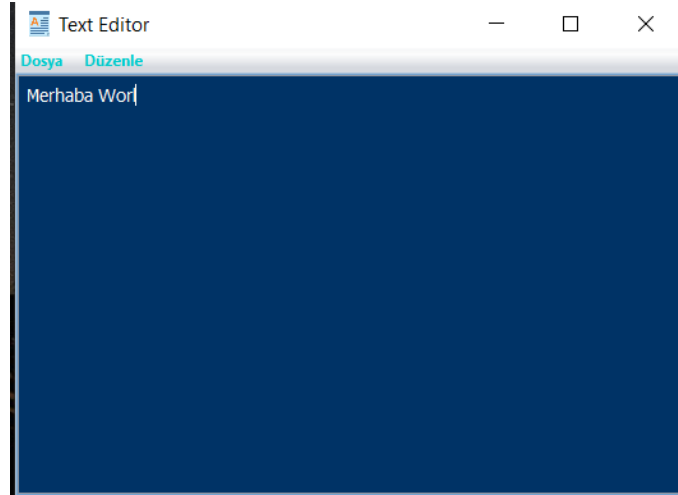
Bir dosya açma örneği



Bir yanlışları düzeltme örneği



Metinde bulunan bir kelimenin değiştirilme örneği



Son resimde bulunan metinden son harfi silme örneği

Proje-2 ile Eklenenler

Command Arayüzü

Geri alma işleminde command pattern'i kullanılmıştır.

GeriAll Sınıfı

Geri alma fonksiyonudur. Metindeki son karakter silinir.

GeriAlmaCommand Sınıfı

Geri alma işleminde command pattern kullanılmıştır. Bu class'ta geri alma işlemi için Command interface'i override edilmiştir.

Invoker Sınıfı

Tema değiştirmek için kullanılan decorator pattern'deki invoker görevini üstlenen class Command fonksiyonunu çalıştırır.

Context Sınıfı

Strategy tasarım desinindeki context görevinde kullanılan sınıftır.

Strategy Sınıfı

Yanlış yazılmış kelimeleri birden fazla yöntem ile düzeltebileceğimiz için bu işlemlerde strategy tasarım deseninin kullanımı uygun görülmüştür.

excessKeysStrategy Sınıfı

Bu sınıfta kelimeleri düzeltmek için kelime yazımı sırasında bir harfe iki kere tıklanmış olabileceği kontrol edilir. Çift olarak bulunan karakterleri silme stratejisi uygulanır. Bu sınıfın içinde “changeWord” adında bir fonksiyon bulunur. Bu fonksiyona gelen kelimenin her karakteri için döngü kurulmuştur. Her karakter bir sonraki karakter ile karşılaştırılır. Eşleşme varsa sonraki karakter kelimedenden çıkarılarak arama yapılır. Arama başarılı olursa yeni kelime döndürülür. Hiçbir karakter için eşleşme bulunamazsa kelime ilk hâli ile döndürülür.

nearKeysStrategy Sınıfı

Bu sınıfta kelimeleri düzeltmek için kelime yazımı sırasında bir harfe basılmak istenirken komşu bir harfe tıklanmış olabileceği üzerinde durulur. Her harfi komşuları ile değiştirme stratejisi uygulanır.

swappingStrategy Sınıfı

Bu sınıfta kelimeleri düzeltmek için kelime yazımı sırasında iki harfin yer değiştirmiş olma ihtimali kontrol edilir. Her harf çifti için 'swapping' stratejisi uygulanır. İçinde bulunan changeWord fonksiyonunda bulunan döngüde her karakterin bir sonraki ile yer değiştirilmesiyle elde edilen kelime arama algoritmasına gönderilir. Kelime düzeltilemez ise ilk hâli döndürülür.

Theme Arayüzü

Decorator tasarım deseninde kullanılan tema arayüzü.

Dark Sınıfı

Theme arayüzünü implemente eder ve içinde tema metotunu bulundurur.

Light Sınıfı

Theme arayüzünü implemente eder ve içinde tema metotunu bulundurur.

ThemeDecorator Sınıfı

Decorator tasarım deseni için thema'yı override eden decorator sınıfıdır.

DarkThemeDecorator Sınıfı

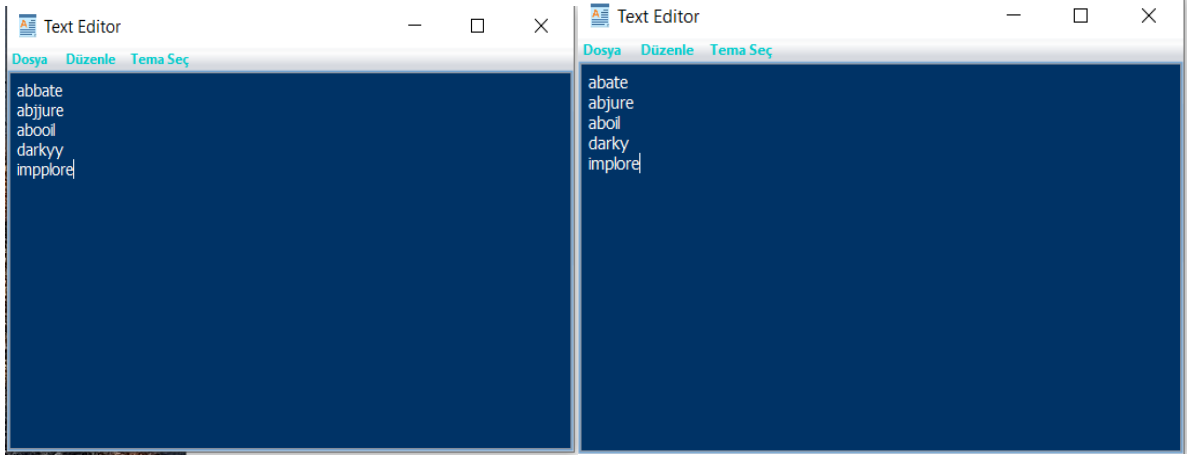
Karanlık tema için görsel düzenlemeleri yapan sınıftır.

LightThemeDecorator Sınıfı

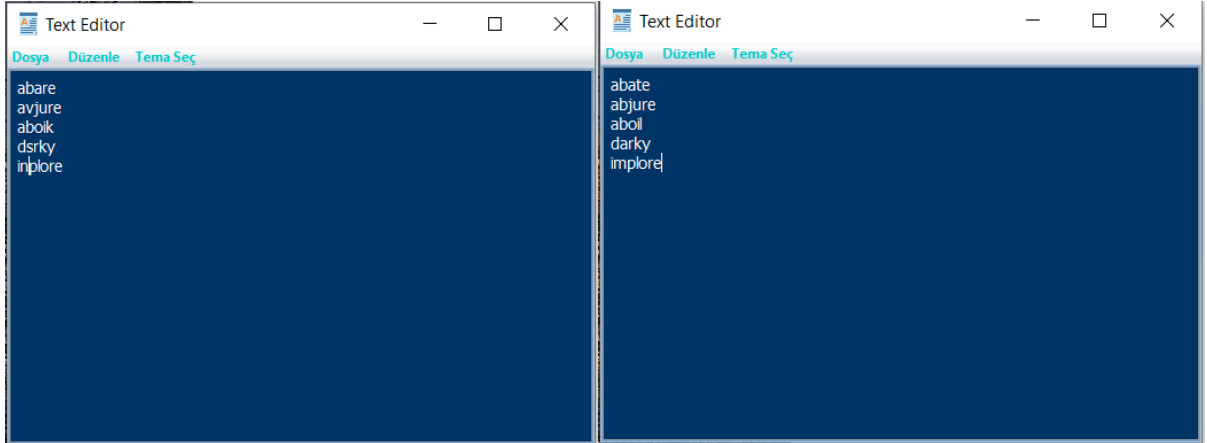
Aydınlık tema için görsel düzenlemeleri yapan sınıftır.

Not: Tema seçimi için uygulamanın arayüzüne Tema Seçimi özelliği eklenmiştir ve karanlık veya aydınlık tema seçildiğinde gerekli objeler `NewJFrame`'de oluşturulur.

Koleksiyon dolaşma gereken yerlerde Java ortamında hazır olarak bulunan "Iterator" deseni kullanılmıştır.



Bir harfe yanlışlıkla iki kere tıklanma örneği



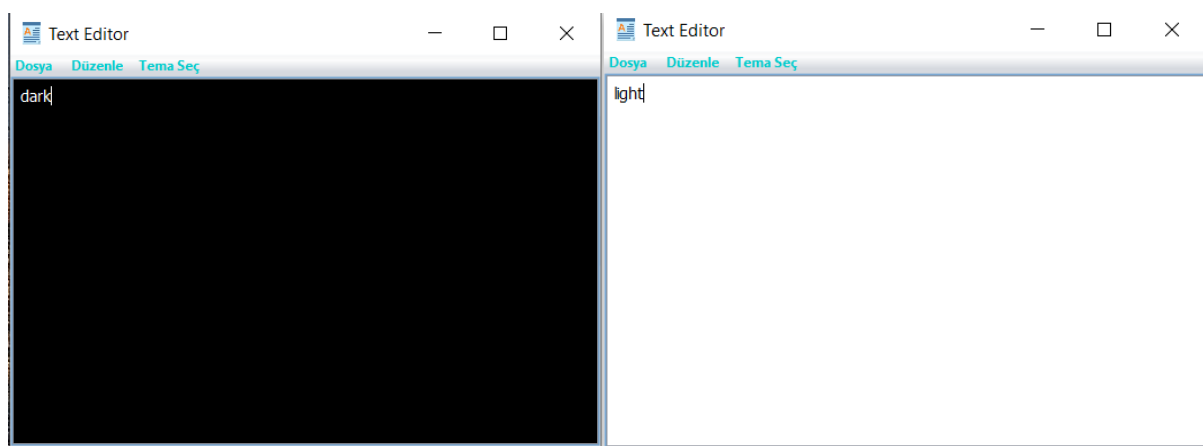
Bir harfe basılmak istenirken komşu bir harfe tıklanma örneği

```

String temp=word;
Content searcher = new Content();
for(CharacterIterator it = new StringCharacterIterator(word);it.getIndex()<word.length()-1;it.next()){
    if(word.charAt(it.getIndex())==word.charAt(it.getIndex()+1)){
        temp=word.substring(0,it.getIndex())+word.substring(it.getIndex()+1);
        if(searcher.varMi(temp)) return temp;
    } else {
        temp = word;
        for(int x=it.getIndex()+1;x<word.length()-1;x++){
            if(word.charAt(x)==word.charAt(x+1)){
                temp=word.substring(0,x)+word.substring(x+1);
                if(searcher.varMi(temp)) return temp;
            }
        }
        word = temp;
    }
}
return word;

```

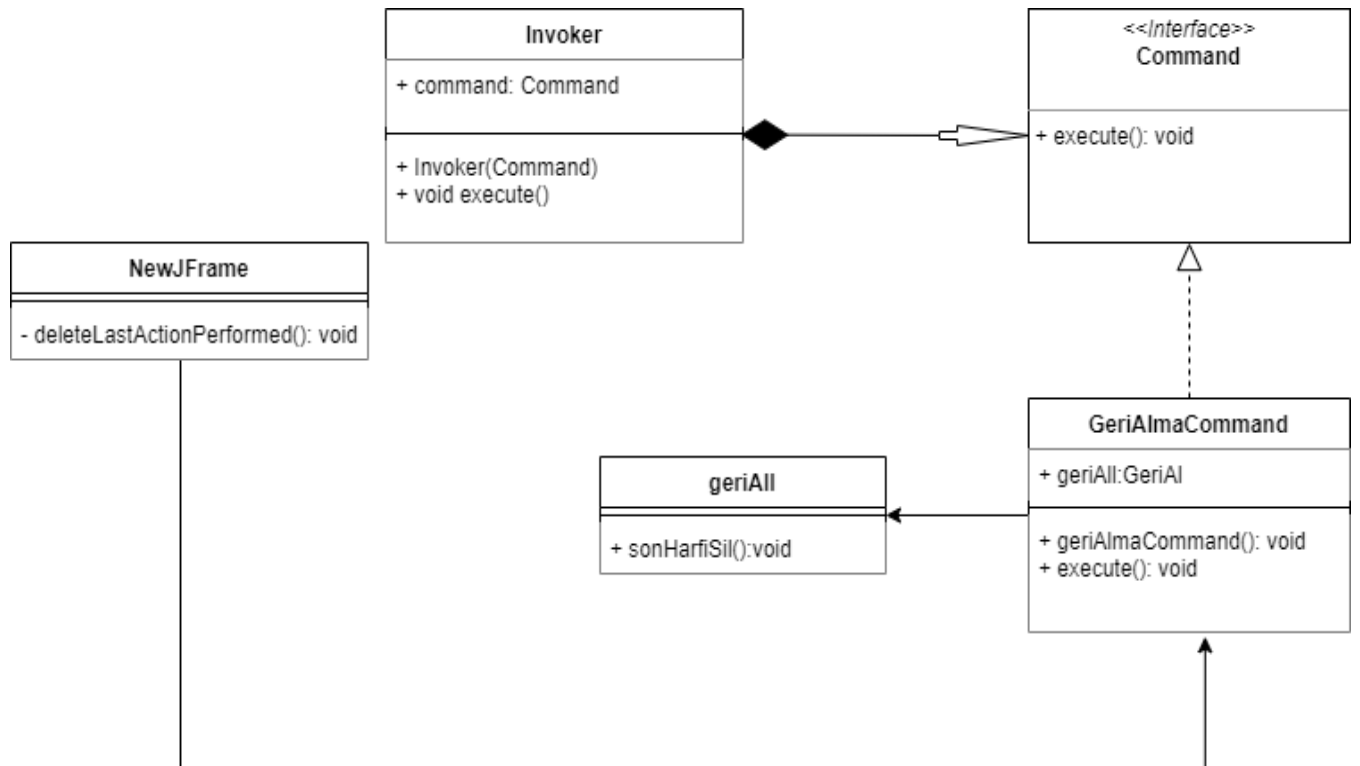
İteratör kullanımına bir örnek



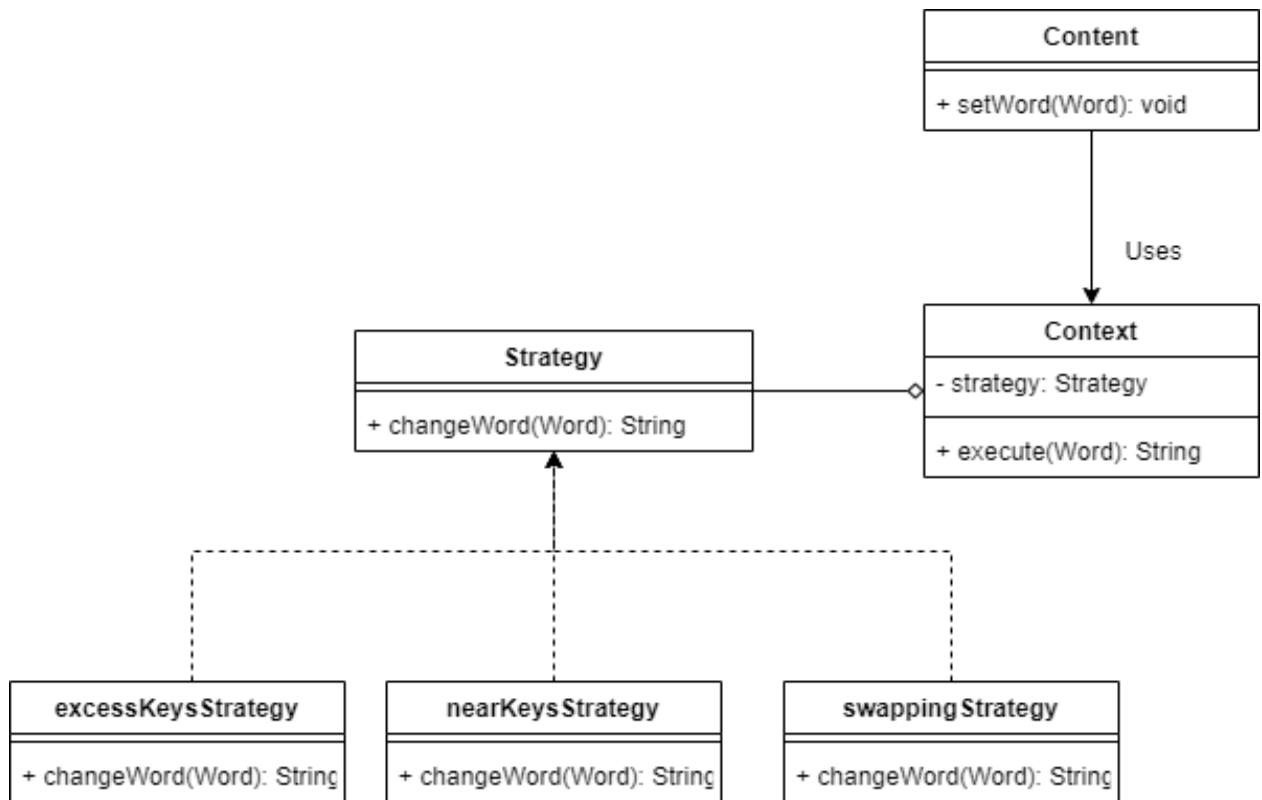
Karanlık ve aydınlık tema

Tasarım Desenleri için UML Diyagramları

Command Tasarım Deseni için UML



Strategy Tasarım Deseni için UML



Decorator Tasarım Deseni için UML

