

Sözcük Çözümleyici Uygulama Yazılımı Test Planı



Ders Adı : Yazılım Kalite ve Test Süreci

Yürütücü İsmi : Prof. Dr. Oya Kalıpsız

Öğrenci Numara, İsim ve Soy İsimleri :

15011069 – Ömer Muhammed Demir

16011044 – İbrahim Furkan Erçelebi

16011105 – Buğra Karaca

16011702 – Mustafa Aydın

16011706 – Duygu Erduran

İçindekiler

1. Giriş.....	
1.1. Amaç.....	
1.2. Kapsam.....	
2. Kaynaklar.....	
3. Proje Tanımı.....	
4. Test Faaliyetleri.....	
4.1. Test Stratejisi.....	
4.2. Test Edilecek Öğeler.....	
4.3. Test Kriterleri.....	
4.4. Hata Raporlama.....	
4.5. Sonuç Raporlama.....	
4.6. Test Araçları ve ortamı.....	
5. Test Çıktıları.....	
6. Paydaşlar.....	

1. Giriş

Bu belge, sözcük çözümlemek isteyen yazılım geliştiricilerinin, sözdizimlerini kontrol altında tutması, tüm dış davranışlarını, gereksinimlerini tanımlamak için hazırlanmıştır.

1.1. Amaç

Amaç, sözdizimi takibini yönetmek ve yazılımları daha iyi inceleyebilmektir.

1.2. Kapsam

Bu belge, içerisinde bahsedilen “Sözcük Çözümleyici Uygulama Yazılımı” projesinde geliştirilecek olan yazılımın test eylemlerini kapsamaktadır.

2. Kaynaklar

Bu planın hazırlanmasında aşağıdaki kaynaklardan faydalanılmıştır.

- Depo Takip Yönetim Sistemi Test Planı
- İnsan Kaynakları Sistemi Projesi Yazılım Test Planı

3. Proje Tanımı

Üst düzey kodu, yani programlama dillerini, bilgisayarın anlayabileceği bir biçime - ikili koda - çevirme görevi, bir derleyicinin ana işidir. Basit bir şekilde konuşursak, derleyici 3 bölüme ayrılabilir:

- Lexical Analyzer – LA (Sözcük Çözümleyicisi)
- Syntax Analyzer – SA (Sözdizimi Çözümleyicisi)
- Semantic Analyzer – SMA (Semantik Çözümleyici)

Sözcük Analizcisi, kaynak kodu, kodu oluşturan sözcükler olan lexeme'lere ayırmaktan sorumludur.

Tüm lexeme'leri ayırdıktan sonra, LA onları Token sınıflandırmasını kullanarak sınıflandırır.

****Keyword'ler****, ****Özel Semboller****, ****Identifier'lar**** ve ****Operatörler**** token örnekleridir.

Derlenmiş kodun beyaz boşluklarını ve yorumlarını kaldırmak da Lexical Analyzer tarafından oynanan bir roldür. Bu işlemin çıktısı, lexeme'leri ve token sınıflandırmalarını içeren bir tablodur. Lexeme birimlerinin geçersiz yapıları olarak sözcük hataları, *ör. '12variableName'*, *'na;;me'*, ayrıca LA tarafından yakalanır.

Bu proje, Java'da yapılmış ****basit**** bir Sözcük Analizcisi'nin bir uygulamasıdır. Kullanıcının kodu yazabileceği ve belirteçlerini alabileceği bir GUI sağlar. Kodu bir dosyadan yüklemek ve analiz yapmak da mümkündür.

Bu projenin Sözcük Çözümleyicisi, aşağıdaki token sınıflarını tanır:

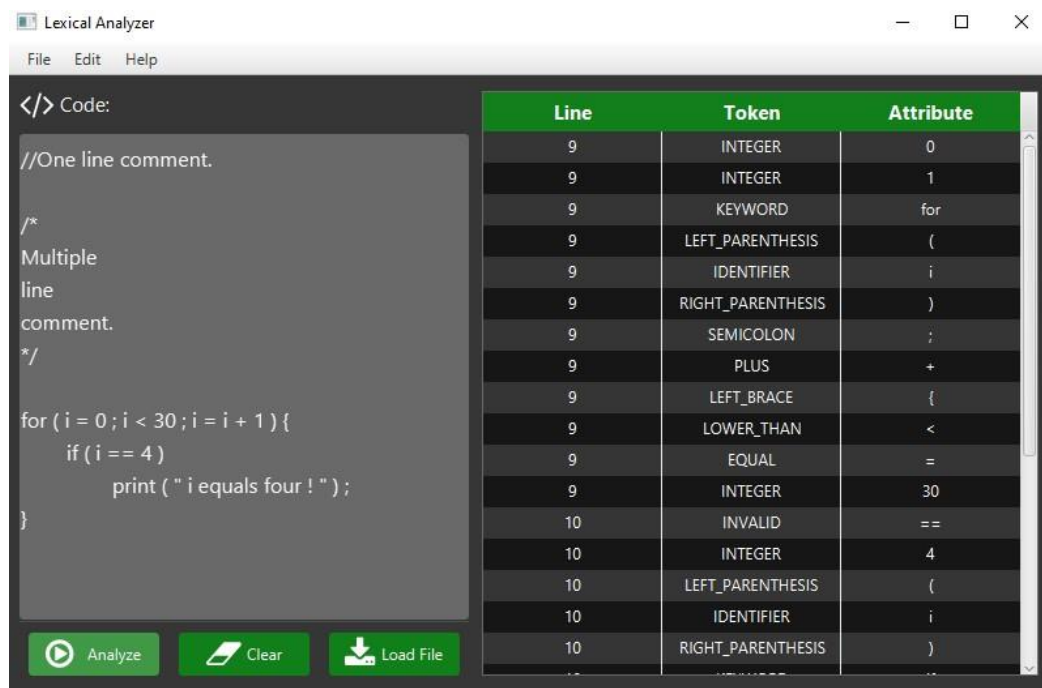
- ****IDENTIFIER**** - Değişken adları;
- ****STRING**** - Çift tırnak arasındaki kelimeler "";
- ****INTEGER**** - Noktasız sayı (.);
- ****FLOAT**** - Kayan nokta sayıları;
- ****PLUS**** - (+);
- ****MINUS**** - (-);
- ****TIMES**** - (*);
- ****DIVIDE**** - (/);
- ****KEYWORD**** - for, while, do, if, else, print, switch, case, default ve null;
- ****INVALID****;
- ****ASSIGN_OP**** - Atama operatörü (=);
- ****SEMICOLON**** - (;);
- ****LEFT_PARENTHESIS**** - '(';
- ****RIGHT_PARENTHESIS**** - ')';
- ****LEFT_BRACE**** - ({);
- ****RIGHT_BRACE**** - (});
- ****COMMA**** - (,);
- ****DOT**** - (.);
- ****DOTDOT**** - (..);
- ****COLON**** - (:);
- ****EQUAL**** - (==);
- ****LOWER_OR_EQUALS**** - (<=);
- ****GREATER_OR_EQUALS**** - (>=);
- ****NOT_EQUALS**** - (<>);
- ****GREATER_THAN**** - (>);

- ** LOWER_THAN ** - (<);

- ** AT_SIGN ** - (@).

***Not 1**:// ile başlayan cümleler veya /* */ arasındaki cümle parçaları yorum olarak kabul edilir ve çıktıda belirtilmez.*

***Not 2**:// Lexeme'ler, ayrılmış şeyler olarak tanınmak için en az bir boşluk (' ') ile ayrılmalıdır.*



Bu, Lexical Analyzer'ın nasıl uygulanabileceğini gösteren çok basit bir örnektir. Bu proje aynı zamanda çok güçlü ve kullanışlı bir araç olan Sonlu Durum Otomata'nın bir kullanım örneğidir.

4. Test Faaliyetleri

4.1. Test Stratejisi

Sözcük Çözümleyici Uygulama Yazılımı projesinde yazılım testleri faaliyetleri süresince gerçekleştirilecek test türleri şunlardır :

Test Türü	Test Kapsamı	Test Tipi	Hata Alımı	Test Numarası	Testi yapan
Fonksiyonel	Birim	Saydam Kutu	Hayır	#01	Mustafa Aydın
Fonksiyonel	Birim	Saydam Kutu	Evet	#02	Mustafa Aydın
Fonksiyonel	Birim	Saydam Kutu	Hayır	#03	Mustafa Aydın
Fonksiyonel	Birim	Saydam Kutu	Evet	#04	Mustafa Aydın
Fonksiyonel	Birim	Saydam Kutu	Hayır	#05	Mustafa Aydın
Fonksiyonel	Birim	Saydam Kutu	Evet	#06	Mustafa Aydın
Fonksiyonel	Birim	Saydam Kutu	Hayır	#07	Duygu Erduran
Fonksiyonel	Birim	Saydam Kutu	Evet	#08	Duygu Erduran
Fonksiyonel	Birim	Saydam Kutu	Hayır	#09	Duygu Erduran
Fonksiyonel	Birim	Saydam Kutu	Evet	#10	Duygu Erduran
Fonksiyonel	Bütünleme	Saydam Kutu	Evet	#11	Furkan Erçelebi
Fonksiyonel	Bütünleme	Saydam Kutu	Evet	#12	Furkan Erçelebi
Fonksiyonel	Bütünleme	Saydam Kutu	Evet	#13	Furkan Erçelebi
Fonksiyonel	Sistem	Kara Kutu	Hayır	#14	Ömer Demir
Fonksiyonel	Onaylama	Saydam Kutu	Hayır	#15	Furkan Erçelebi
Fonksiyonel	Onaylama	Saydam Kutu	Hayır	#16	Furkan Erçelebi
Fonksiyonel	Onaylama	Saydam Kutu	Hayır	#17	Furkan Erçelebi
Arayüz	Birim	Saydam Kutu	Hayır	#18	Buğra Karaca
Arayüz	Birim	Saydam Kutu	Evet	#19	Buğra Karaca
Arayüz	Birim	Saydam Kutu	Hayır	#20	Buğra Karaca
Arayüz	Birim	Saydam Kutu	Evet	#21	Buğra Karaca
Arayüz	Birim	Saydam Kutu	Hayır	#22	Buğra Karaca
Arayüz	Birim	Saydam Kutu	Evet	#23	Buğra Karaca
Arayüz	Sistem	Kara Kutu	Hayır	#24	Ömer Demir
Performans	Birim	Saydam Kutu	Hayır	#25	Furkan Erçelebi
Performans	Birim	Saydam Kutu	Hayır	#26	Furkan Erçelebi
Performans	Birim	Saydam Kutu	Hayır	#27	Furkan Erçelebi
Performans	Sistem	Kara Kutu	Hayır	#28	Ömer Demir
Uyumluluk	Sistem	Kara Kutu	Hayır	#29	Ömer Demir
Uyumluluk	Sistem	Kara Kutu	Hayır	#30	Ömer Demir
Uyumluluk	Sistem	Kara Kutu	Hayır	#31	Ömer Demir
Güvenlik	Sistem	Kara Kutu	Hayır	#32	Ömer Demir
Güvenlik	Sistem	Kara Kutu	Hayır	#33	Ömer Demir

Gereksinim spesifikasyonu dikkate alınarak oluşturulan modüllerin testleri gerçekleştirilecektir.

4.2. Test Edilecek Öğeler

4.3. Test Kriterleri

Sözcük Çözümleyici Uygulama Yazılımı projesinde genel test faaliyetleri kriterleri şu şekildedir :

1. Planlanmış test faaliyetleri gerçekleştirilmiş olmalıdır.
2. Test senaryoları adım adım gerçekleştirilmiş olmalıdır.
3. Çıkan hataların çözümü olmalı; hatalar, bütünleme testlerinde ise çözümler, onaylama testi içinde yer almalıdır.
4. Her test mutlaka raporlanmalıdır.

4.4. Hata Raporlama

Sözcük Çözümleyici Uygulama Yazılımı projesi test aşamasında karşılaşılan hatalar, aşağıdaki gibi sınıflandırılmıştır.

Kritik derece hatalar: Sistemi çalışmasını etkileyen hatalardır.

Yüksek derece hatalar: Sistemin çalışmasını aksatmayacak fakat gereksinimleri engelleyecek hatalardır.

Orta derece hatalar: Modül arayüz hatalarıdır.

Düşük derece hatalar : Kullanımı aksatmayan ve gereksinimlerde bulunmayan hatalardır.

4.5. Sonuç Raporlama

Sözcük Çözümleyici Uygulama Yazılımı projesi kapsamında gerçekleştirilen tüm testlerin sonuçları raporlanarak saklanacaktır. Testlerin her biri için test durumu yazılacak ve test sonuç raporuna eklenecektir.

4.6. Test Araçları

Sözcük Çözümleyici Yazılım Uygulaması projesi birim testleri, Visual Studio Code üzerinde gerçekleştirildi. Projenin yazılım testleri için testçiler kendi bilgisayarlı üzerinde yazılımın geliştirme ortamını kurdular. Testçiler farklı platformlardaki (Windows ve Linux) kendi bilgisayarları üzerinden tanımlanan testleri gerçekleştirdiler.

5. Test Çıktıları

Sözcük Çözümleyici Uygulama Yazılımı projesi kapsamındaki her test için alınacak çıktılar :

- Durum önceliği
- Test türü
- Yürütme türü
- Açıklama
- Test adımları
- Beklenen sonuç

Sözcük Çözümleyici Uygulama Yazılımı projesi kapsamındaki her hata için alınacak çıktılar :

- Önem derecesi
- Açıklama
- Test adımları
- Test verisi
- Beklenen sonuç
- Gerçek sonuç

Sözcük Çözümleyici Uygulama Yazılımı projesi kapsamında alınacak çıktılar :

- Gereksinim spesifikasyon belgesi
- Yazılım test planı
- Test sonuç raporu

6. Paydaşlar

Mustafa Aydın (Proje Yöneticisi)

İbrahim Furkan Erçelebi (Birim Test Geliştirici)

Duygu Erduran (Sistem Analisti)

Ömer Muhammed Demir (Kara Kutu Test Geliştirici)

Buğra Karaca (Arayüz Test Geliştirici)