**T.C.**

**FIRAT ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**YAZILIM MÜHENDİSLİĞİ**

**YMH 319 Programlama Dilleri**

**Dört İşlem Kodlanabilen Basit Bir Programlama Dili ve Derleyicisi**



**Kerimhan BADUR 180541033**

**Doç. Dr. Fatih ÖZKAYNAK**

**Şubat – 2021**

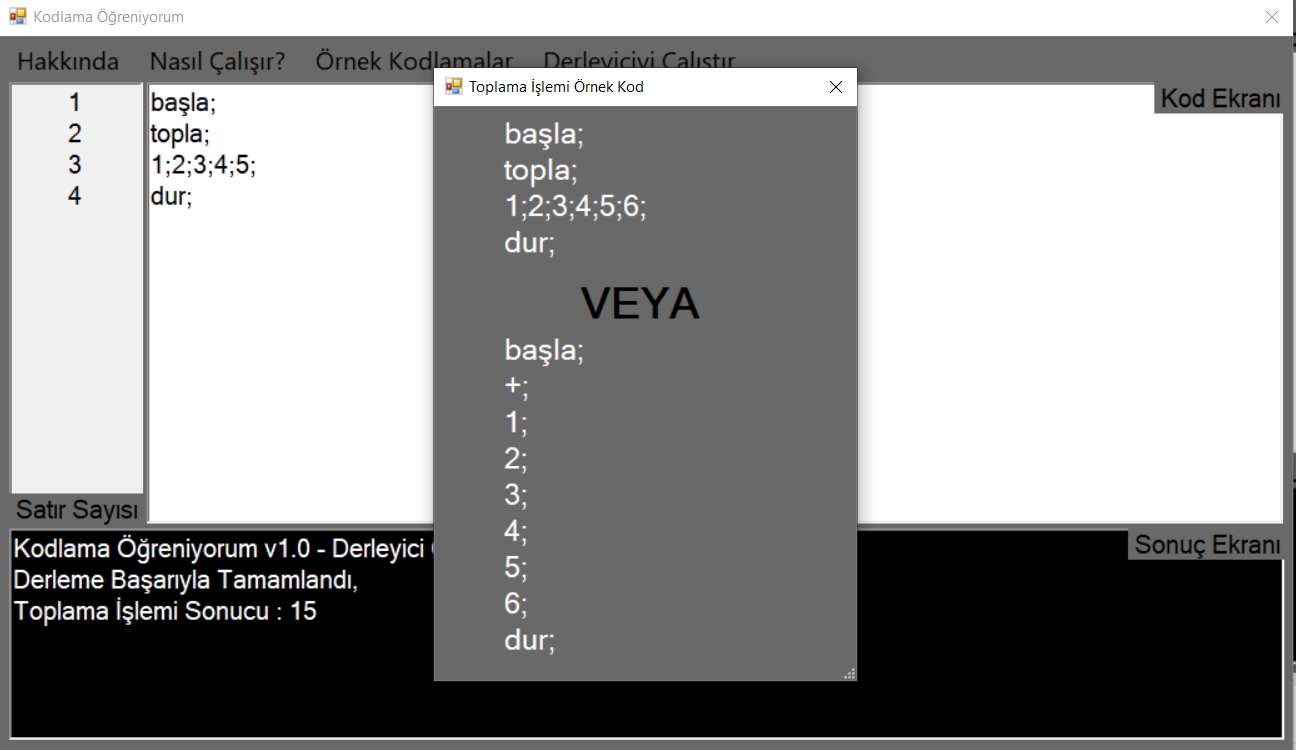
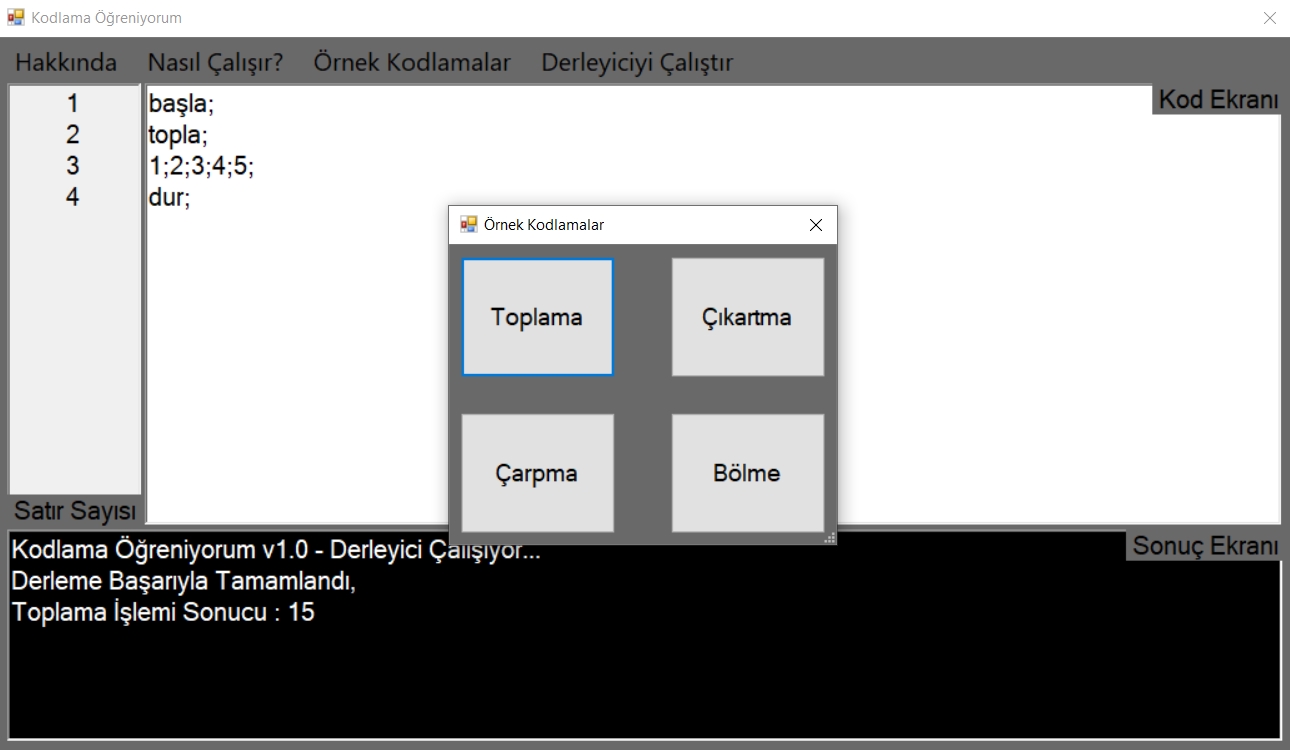
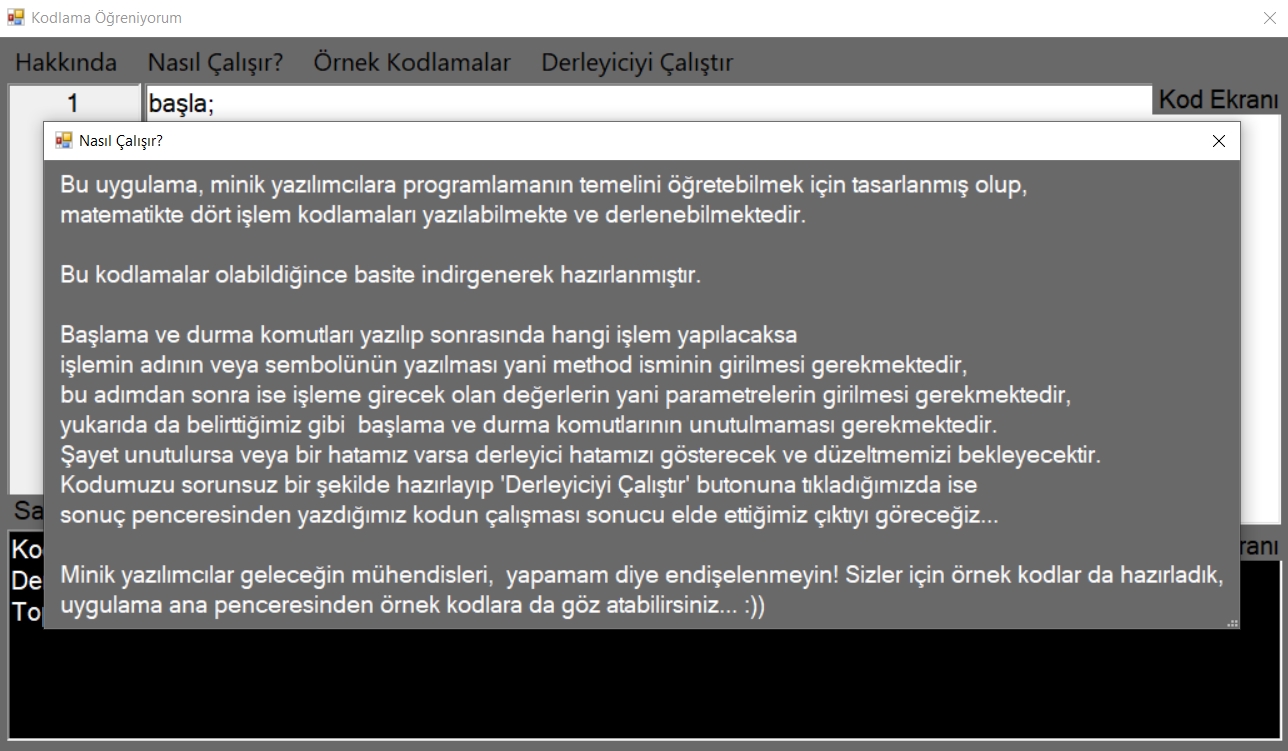
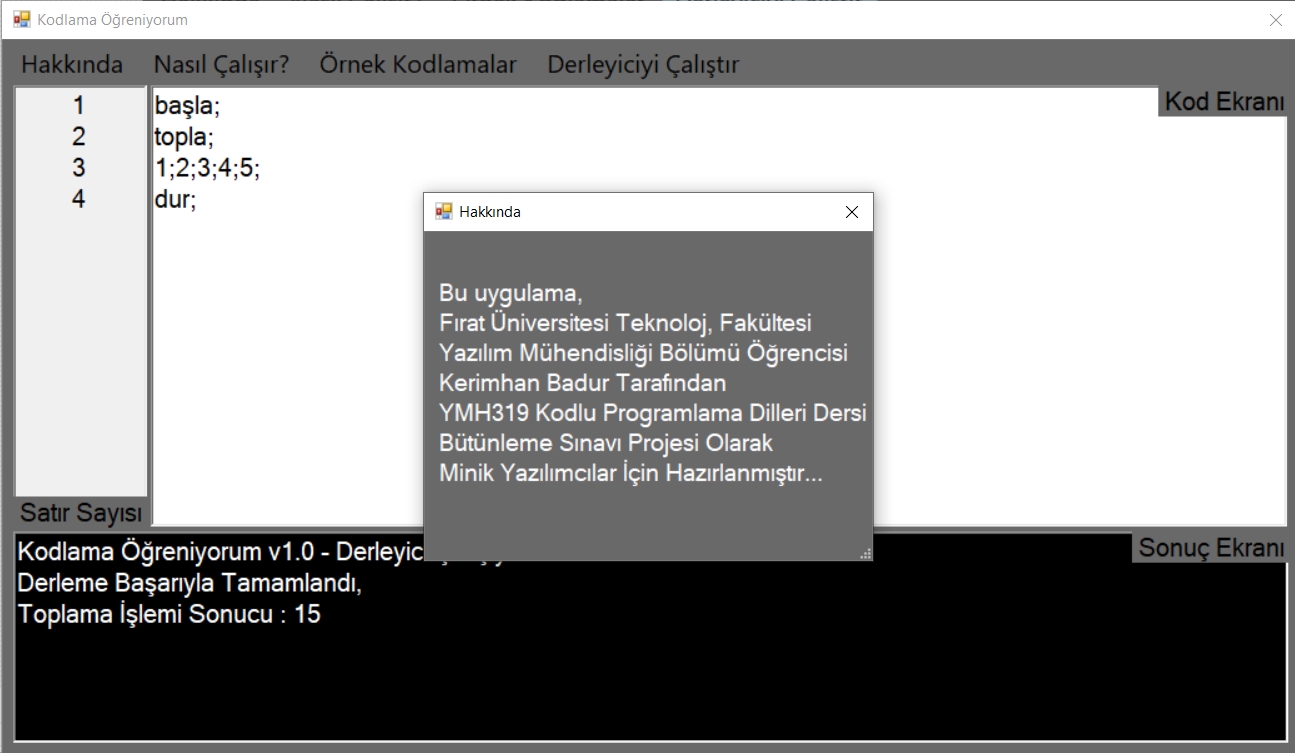
**TASARIM AŞAMALARI**

**Giriş :**

Bizden istenilen, matematiksel dört işlem yapabilen basit bir kod tasarlanması, gerçeklenmesi ve derleyicisi ile birlikte uyarlanabilmesi.

**Arayüz Tasarımı :**

Hedef kitlemiz çocuklar olduğu için basit bir arayüz tercih ettim, gerçekleştirmiş olduğum programın arayüzünü aşağıdaki görseller ile görmektesiniz…



**Programlama Dili Tasarım Süreci :**

* Dil tasarımında hedef kitlemiz çocuklar olduğu için, çocukları çok sıkmadan fakat

aynı zamanda da en azından temel bir şeyleri öğretebilmek adına bir yol izledim.

* Tasarlamış olduğum dilin “Başla” ve “Dur” komutları mevcut buradaki ana gayem

hedef kitlemiz olan çocuklara algoritma mantığındaki temel kuralı kavratmak ve aynı zamanda yazacakları kodların bu blok dışında çalışmayacağını benimsetmek.

* Tasarlamış olduğum dilde değişken atama bulunmamakta olup

kullanıcılar(çocuklar) yapacağı işlemi yazıp bu işleme tabi olacak değerleri yazıyorlar ve işlem böylelikle şayet bir hata yok ise gerçekleşiyor.

* Bir üst maddede belirttiğim şayet hata yok ise ibaresi ne anlama geliyor? Derleyici

kısmında kontrolü sağlayabilmek ve hedef kitlemiz olan çocuklara bir çok dilde yaygın kullanılan “;” noktalı virgül kullanımını kazandırmak için ben de tasarlamış olduğum dilde bir hayli kullandım, yani bir hata yok ise ibaresi buradan geliyor eğer kullanıcı “;” kullanmayı unutursa tabi ki derleyici kısmında hata ile karşılaşıyor ve yazmış olduğu program çalışmıyor…

* Dil tasarım sürecinde kullanıcıya destek olmak amaçlı örnek kodların olduğu

farklı bir form oluşturarak kullanıcının dile daha hızlı adapte olmasını amaçladım…

**Örnek Görsel**



**Örnek Kod Blokları**

başla; başla;

topla; +;

1;2;3;4;5;6;7;8;9; 1;2;3;4;5;6;7;8;9;

dur; dur;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

başla; başla;

çıkar; -;

15;10;2; 15;10;2;

dur; dur;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

başla; başla;

çarp; \*;

1;2;3;4;5; 1;2;3;4;5;

dur; dur;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

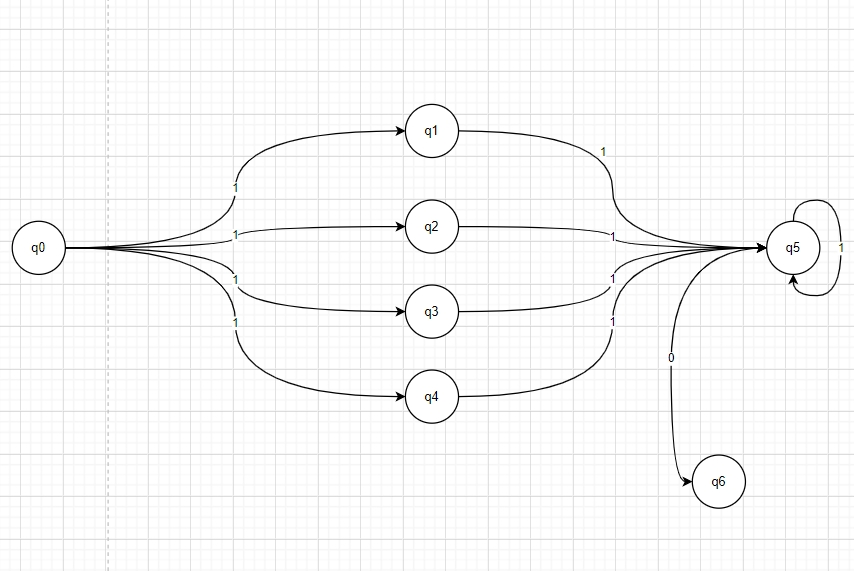
başla; başla;

böl; /;

15;5 15;5;

dur; dur;

**Deterministic Finite Automata (Dfa Diyagramı) Tasarımı :**



**q0 🡪 başla; komutudur.  
q1,q2,q3,q4 🡪 topla; çıkar; çarp; böl; komutlarıdır.**

**q5 🡪 işleme girecek olan değerlerdir.**

**q6 🡪 dur; komutudur.**

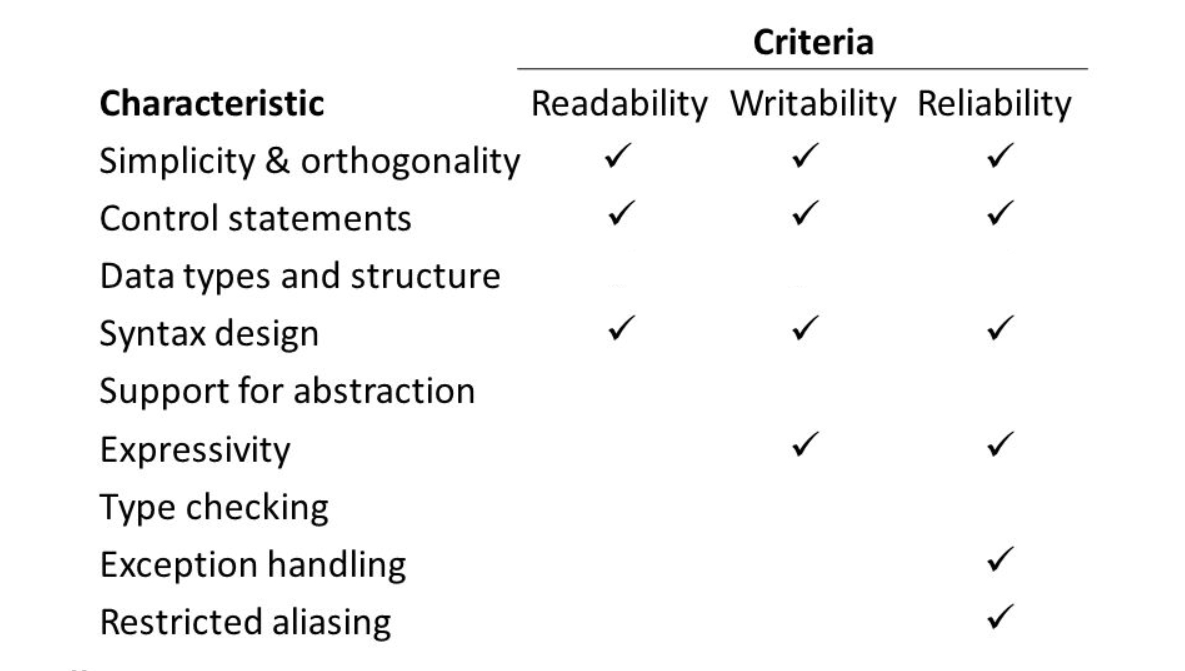
**Dil Değerlendirme Kriterleri :**

Okunabilirlik 🡪 Gerçekleştirmiş olduğum bu projenin hedef kitlesi çocuklar olduğu için tasarlamış olduğum dil son derece okunabilir ve anlaşılabilir kodlardan oluşmalılıdr, zaten bu bağlamı göz önüne alarak herhangi bir atama işlemi gerçekleştirme yapılmamıştır…

Yazılabilirlik 🡪 Gerçekleştirmiş olduğum bu projenin hedef kitlesi çocuklar olduğu için tasarlamış olduğum dilde yazılabilirlik de oldukça kolay olmalıdır. Bu bağlamda main bloğu, sınıf, vs... gibi kavramlar kullanılmamış olup oldukça basit yazılabilir bir dil olmasına özen gösterilmiştir…

Güvenilirlik 🡪 Gerçekleştirmiş olduğum projede yazılacak olan kodlar sadece matematiksel dört işlem içereceği için güvenilirlik üzerinde durulmamıştır…

Maliyet 🡪 Gerçekleştirmiş olduğum projede maliyet açısından önemli ve etkin olan tek kavram süre idi. 24 saat gibi bir sürede elimden gelenin en iyisini yapmaya çalıştım…



**Programlama Paradigmaları :**

Gerçekleştirmiş olduğum bu dil programlama paradigmaları kategorisinden;

**Zorunlu Programlama Paradigması (Imperative)**

**Fonksiyonel Programlama Paradigması (Functional)**

**Nesne Yönelimli Programlama Paradigması (Object Oriented)**

**Paralelizm (Parallelism)**

**Mantıksal Programlama Paradigması (Logical)**

işaretli paradigmaya uymaktadır…

**Derleyici Aşamaları :**

* **Lexical Analyzer**
* **Syntax Analyzer**
* **Semantic Analyzer**
* **Ara Kod Üretici**
* **Kod Optimizasyonu**
* **Kod Üretici**
* **Sembol Tablosu Yöneticisi**
* **Hata Yönteicisi**

**Syntax Ve Semantic :**

**Syntax :** başla; ve dur; komutları dışında syntaxı şu şekildedir,

kullanıcı yapacağı işlemi veya operasyon sembolünü yazar ve bu işleme tabi tutacağı değerleri atar..  
  
 Örnek gösterim 🡪 başla; çıkar; 15;10; dur; 🡪 kullanıcı bu kod bloğunu yazdığında sözdizim hatası yapmamışsa 15-10 : 5 sonucunu elde eder.

**Semantic :** Anlamı ise kullanıcı programı başlatır, yapacağı işlemi girer, işleme tabi olacak değerlerin atamasını gerçekleştirir ve son olarak sonlandırma adımı gerçekleştirip derleme işlemine tabi tutar…

**Soyut Sözdizimi :**

Programın arka tarafında ise işlem şöyle işlemektedir, program kullanıcının başla; ve dur; komutları arasında yazdığı fonksiyonu(Ör: topla; çıkar; çarp; böl;) belirler ve bu fonksiyona tabi olacak sayıları kullanıcıdan alır, söz dizimini denetledikten sonra eğer hata yok ise sonrasında bu işlemlerde yapılması mümkün olmayan bir durum var mı(Ör: başla; böl; 4; 0; dur; herhangi bir sayının 0’a bölünmesi mümkün değildir…) gibi çeşitli kontrolleri gerçekleştirdikten sonra hata yok ise kullanıcıya yazdığı kodun sonucunun derlenmiş haliyle sonuç ekranında gösterir…

**Metinsel Sözdizimi :**

Ör 🡪 başla; topla; **14;** 16; dur;

Çizili olan kısmı ele alacak olursak;

**Token** 🡪 Num **Lexeme** 🡪 14

**Değişkenler :**

Gerçekleştirmiş olduğum dilde değişken olarak sadece işlemler kullanılmaktadır, bunlar ise 🡪 topla; veya +; 🡪 çıkar; veya -; 🡪 çarp; veya \*; 🡪 böl; veya /;

**Değişkenler – Atama :**

Gerçekleştirmiş olduğum dilde değişken atama işlemi de ; ile gerçekleştirilmektedir(bu işleme tam bir değişken atama denilemez bir örnek ile görelim…)

Ör 🡪 başla; topla; **1;2;3;4;5;** dur; burada işaretlenmiş alanda toplama işlemine tabi olacak sayıların atanmasını görmekteyiz…

**Deyimler :**

Gerçekleştirmiş olduğum dilde deyimler olarak : başla; topla; çıkar; çarp ; böl; dur;

Sayılabilir…

**Durağan Bağlama (Statik Bağlama) :**

**+ : toplama;**

* **: çıkarma;**

**\* : çarpma;**

**/ : bölme;**

**; : sonlandırma;**

**Dinamik Bağlama :**

Bu işlem değişkenlerin değerler ile bağlanmasıdır, çalışma zamanında gerçekleştirilir tasarlamış olduğum dilden örnek verecek olursam;

Ör 🡪 başla; topla; **1;2;3;4;5;** dur;

işaretli sayılar toplama işlemi için yapılan atamaları göstermektedir ve çalışma zamanında uyarlanmaktadır…