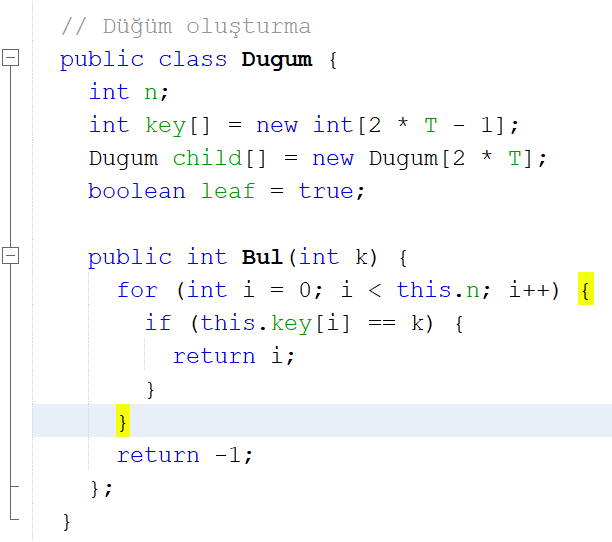
**B Ağaçları ve İfade Ağaçları**

**B Ağaçları**

B ağaçları bir düğümde birden fazla veri tutulmasına olanak sağlayan ikili ağaçlardır. B ağaçları dengelidir. Bir düğümde tuttukları veri sayısı B ağacının kapasite değeri olarak adlandırılır. B ağaçları sıralı ağaçlardır. B ağaçlarına eleman ekleme yapıldığında eğer düğümdeki kapasite değeri aşılmışsa düğüm bölünür. T, B ağacının kapasite değeri olarak kabul edilirse,

* Kök düğümde en az 1, en çok 2T kayıt tutulabilir.
* Kök düğümden en az 2, en çok (2T+1) düğüme işaret edilebilir.
* Kök düğüm dışındaki herhangi bir düğüm en az T, en çok (2T+1) düğüme işaret edebilir.

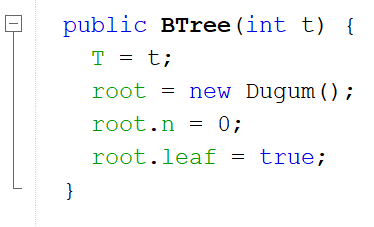
Bu özellikler dikkate alınarak aşağıdaki gibi bir B ağacı oluşturulmuştur. B ağacı oluştururken öncelikle aşağıdaki kod parçacığı kullanılarak bir düğüm sınıfı oluşturulur.



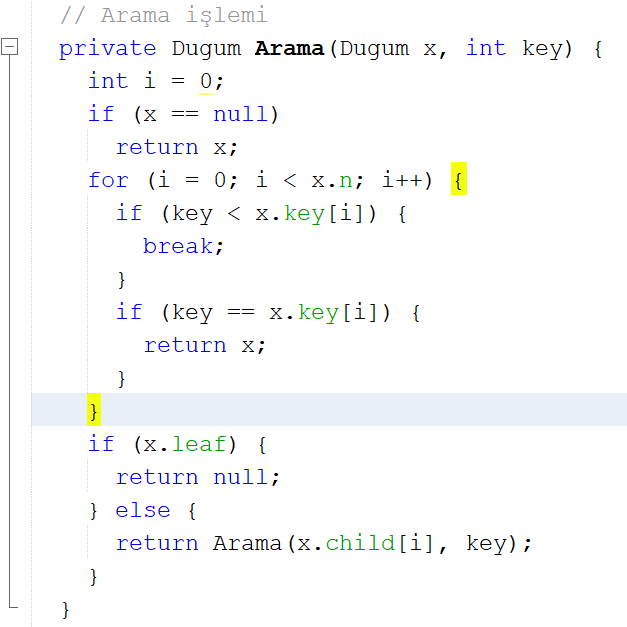
Buradaki T değişkeni kapasite değerini belirtir ve aşağıdaki gibi tanımlanır.



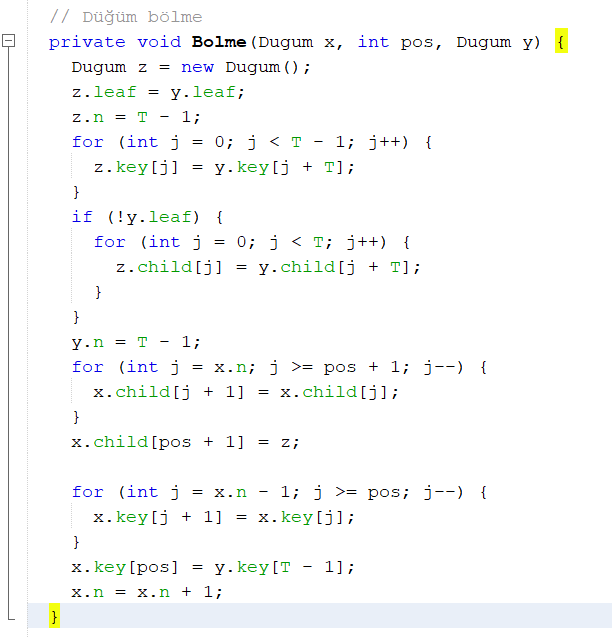
Düğüm oluşturulduktan sonra B ağaç yapısı oluşturmak için aşağıdaki kod parçacığı kullanılır.



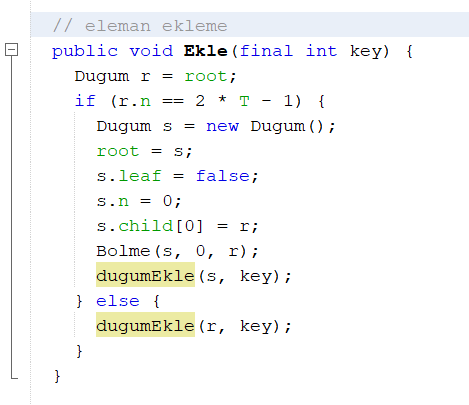
Ağaçta bir elemanı aramak için aşağıdaki kod parçacığı yazılabilir;



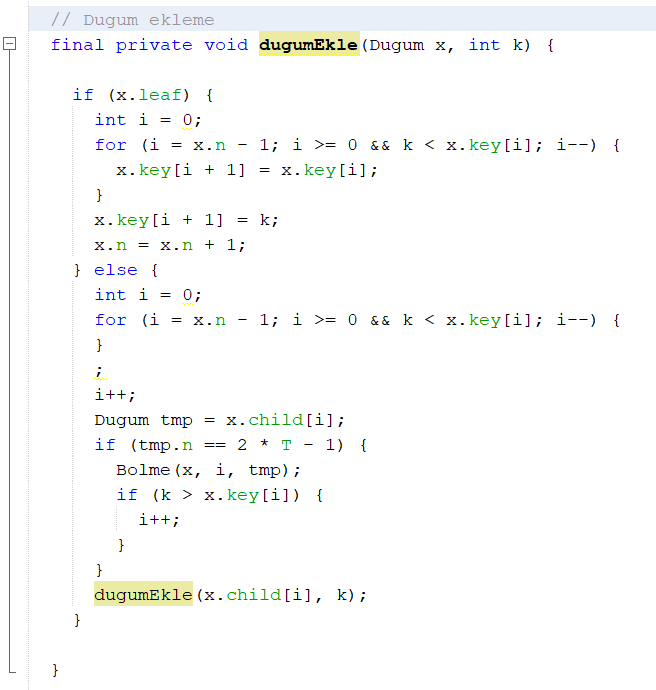
Ağaca eleman eklenince kapasite değeri aşıldığında aşağıdaki gibi düğüm bölme işlemi gerçekleştirilir.



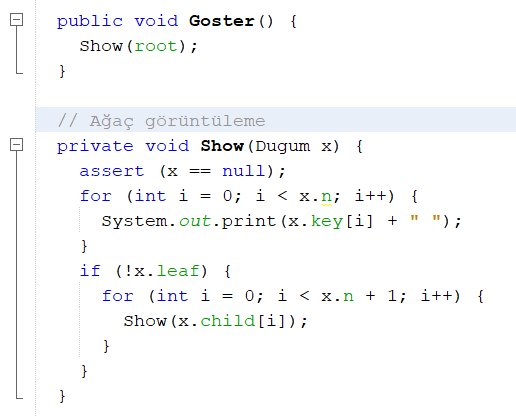
Ağaca eleman eklemek için aşağıdaki kod parçacığı yazılabilir.



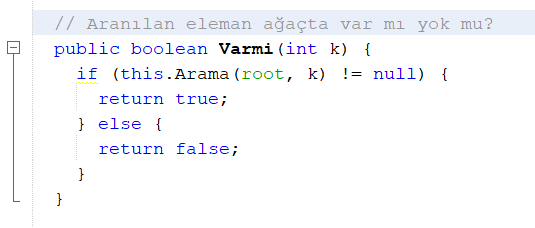
Eleman ekleme metodunda kullanılan düğüm ekleme metodu aşağıdaki gibidir.



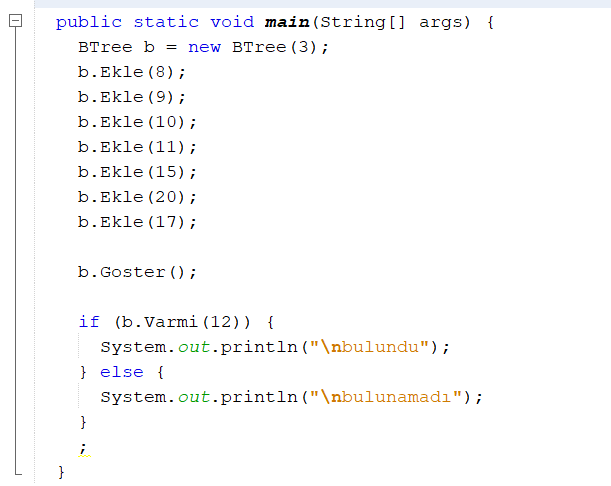
Ağacı görüntülemek için aşağıdaki kod parçacığı yazılabilir.



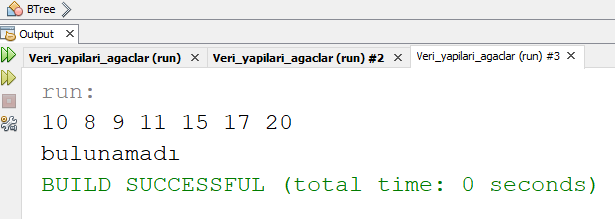
Aranılan elemanın ağaçta olup olmadığı aşağıdaki metot ile kontrol edilebilir.



B ağacı oluşturulduktan sonra aşağıdaki gibi main metodu ile aşağıdaki gibi eleman eklenebilir ve bir elemanın ağaçta olup olmadığı sorgulanabilir.



Bu kod çalıştırıldığında aşağıdaki gibi bir çıktı elde edilir.



**İfade Ağaçları**

İfade ağaçları bir matematiksel ifadenin ağaç şeklinde tutulması için tanımlanmış bir veri yapısı türüdür. A + (B \* (C / D) ) şeklinde bir aritmetik işlem verilmiş olsun. Bu işlemin ifade ağacındaki yerleşimi aşağıdaki şekildeki gibidir:

**Kök**

Ağacın genel yapısı:

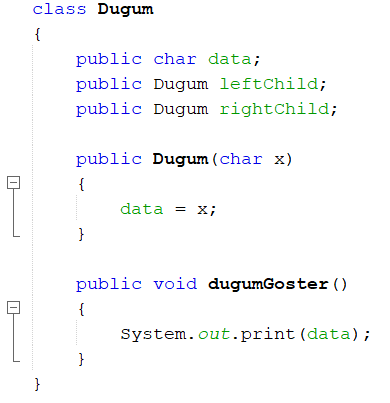
* + Yaprak düğüm = değişken/sabit değer
  + Kök veya ara düğümler = operatörler şeklindedir.

Birçok derleyicide kullanılmaktadır. Parantez gereksinimi yoktur.

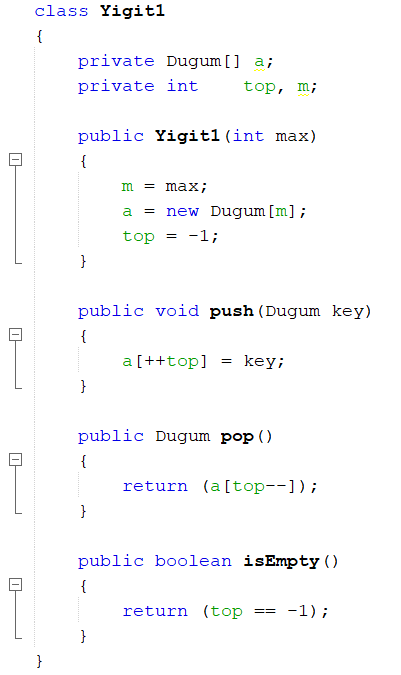
Verilen denklemden ifade ağacı kurulması veya verilen ağaçtan denklemin çıkartılması için üç değişik yöntem vardır:

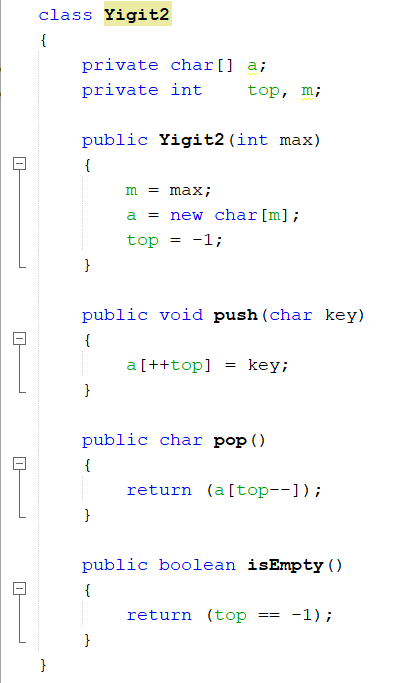
* + Inorder: matematikte alışılagelen şekilde, operatörlerin ortada, değişken veya sabit değerlerin operatörün kenarında bulunan yöntemdir.
  + Preorder: operatörler kendilerine parametrelerin(değişken/sabit) önündedir.
  + Postorder: operatörler parametrelerinin arkasından gelmektedir.

İfade ağacı oluşturmak için öncelikle aşağıdaki gibi bir düğüm yapısı oluşturulur.

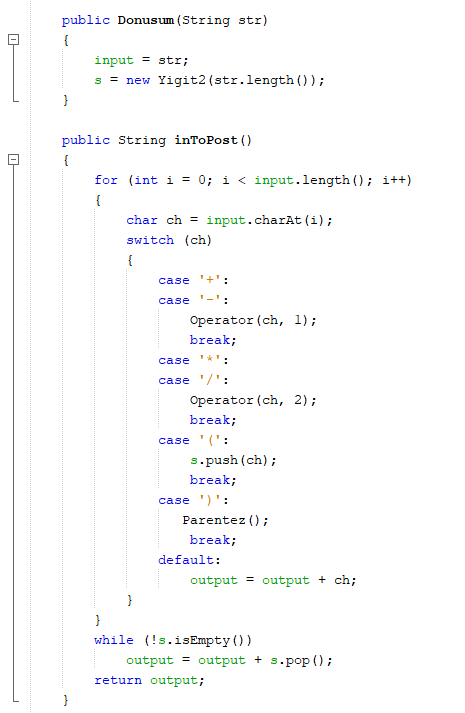


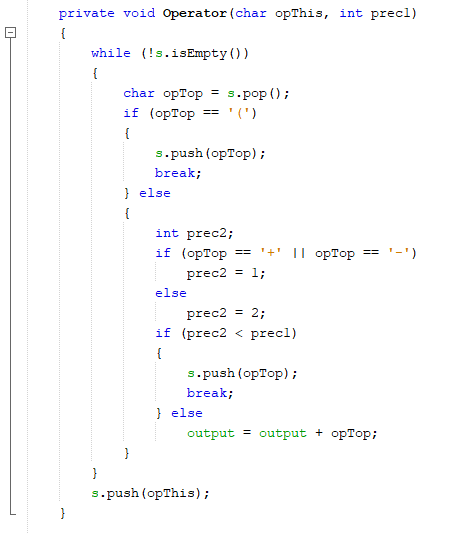
Değişkenlerin ve operatörlerin tutulması için iki ayrı yığıt yapısı oluşturulur.

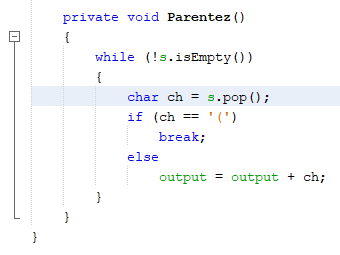




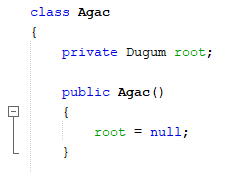
Girilen ifadedeki karakterlerin operatör mü, değişken mi veya parantez mi olduğunu belirleyebilmek için aşağıdaki gibi bir dönüşüm metodu yazılabilir.



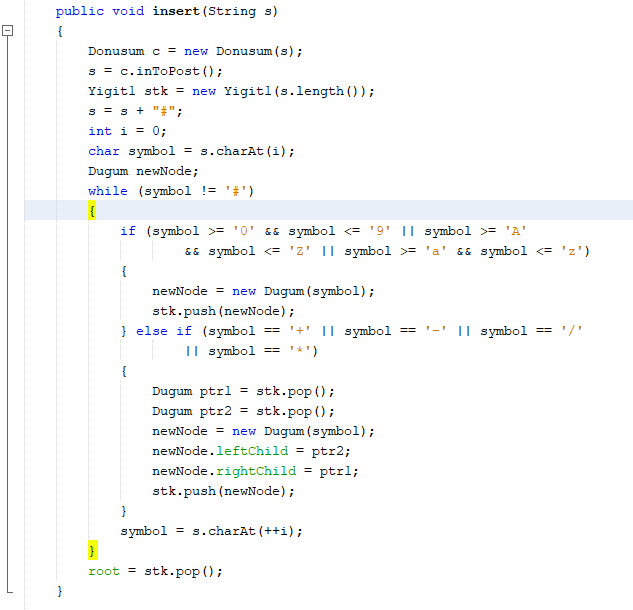




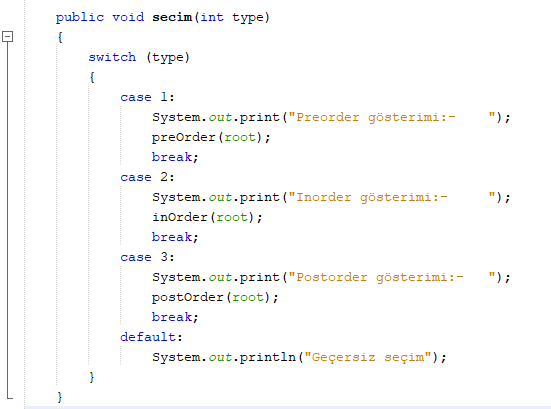
Daha sonra aşağıdaki gibi bir ağaç sınıfı oluşturulur.



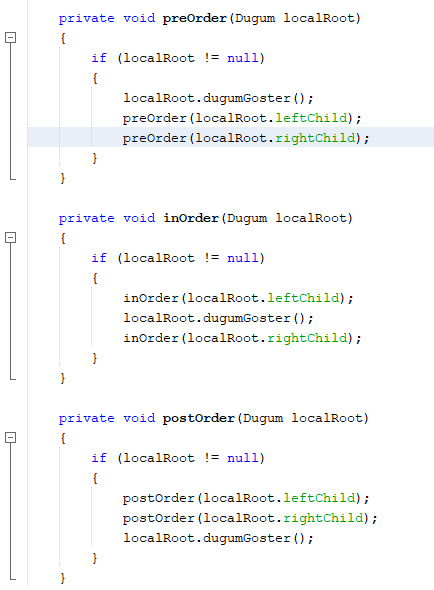
Ağaç sınıfında aşağıdaki gibi ekleme metodu yazılır.



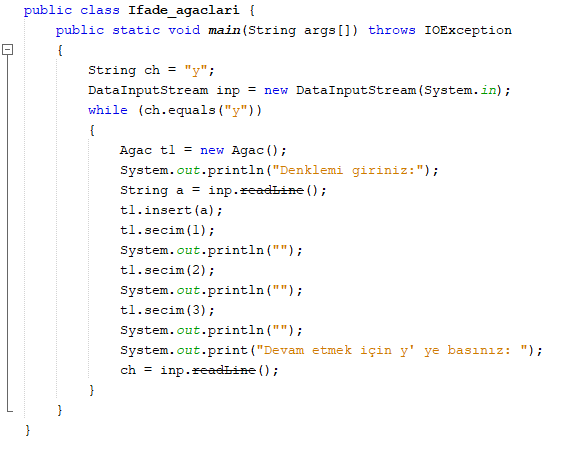
Denklemin gösterim şeklinin seçimi için de aşağıdaki metot yazılabilir.



Preorder, inorder ve postorder gösterimler için aşağıdaki metotlar yazılır.



Oluşturulan ifade ağacına eleman ekleme ve gösterim yapmak için aşağıdaki gibi bir main metot yazılabilir.



Bu kod çalıştırıldığında aşağıdaki gibi bir çıktı elde edilecektir.

