

**T.C.**

**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**NESNEYE DAYALI PROGRAMLAMA ÖDEV RAPORU**

**ÖDEV 3**

**Zeynep ALDİNÇ B181210390**

**SAKARYA**

**Nisan, 2019**

Nesneye Dayalı Programlama Dersi

C# Dilinde Hesap Makinesi Programı

**Özet**

Bu projede C# Form uygulamasında kullanıcı tarafından girilen bir matematiksel işlemi hesaplayan program yapılmıştır. Fakat programı kısıtlı işlemler yapmaktadır. Mesela parantezli ifadeler, üs alma, karekök, trigonometrik ifadeleri içerisinde barındırmamaktadır. Program temel 4 işlemi yapmaktadır.

**Yapım**

Öncelikle yapılması gereken kullanıcının girdiği işlemdeki sayıları ve işaretleri tespit etmekti. Bir döngü içerisinde işlemin üzerinden tek tek geçerek sayıları bir listede, işaretleri başka bir listede toplamayı düşündüm. Ancak sayı birden fazla basamaklı olabileceği için öncelikle bir if kontrolü içerisinde IsDigit() fonksiyonunu kullanarak verinin sayı olup olmadığını kontrol ettim. Burada kullanıcının girdiği verinin içerisindeki değer bir rakam olduğu sürece veri bu if içerisine girecek ve içeride sayi isimli string değişkenine atanacak. Böylece sayıyı tüm basamaklarıyla tam olarak bir değer içerisine almış olacağım. Else kısmında ise ondalıklı sayıları da alabilmesi için bir if kontrolü daha açarak eğer veri içerisinde virgül varsa onun da sayi değişkenine atanmasını istedim. Bu if’in else’inde ise dört işlem işaretlerine ne olacağını belirledim. İşaretleri, Isaretler isimli bir listeye ekleyerek orada tuttum. Program bir işarete denk geldiği için aynı zamanda verideki ilk sayıyı da tamamen okuduğumu ve sayi değerine attığımı anlamış oldum. Bu değeri double’a çevirerek Sayilar isimli double bir liste atadım. Ve yeni sayıyı almaya hazır olması için sayi değerinin içini boşalttım. En son sayıyı da veriden aldıktan sonra hazırladığım while döngüsünden çıkılacağı için en son sayı, sayi değerinden alınıp Sayilar listesine eklenmemiş olacak. Bu yüzden döngüden sonra tekrardan sayi değerini Sayilar listesine ekledim. Isaretler listesindeki işaretlerin işlem önceliği vardır. Bu işlem önceliğinde önce çarpma veya bölme sonra da toplama veya çıkarma işlemleri gelir. Öncelikle listedeki çarpma ve bölme işaretleri bulunmalıydı. Bu yüzden listenin sonuna “.” işaretini ekledim ve while döngümü bu “.” işaretini görene kadar dönmesi için ayarladım. Bu döngünün üzerinde, listelerin indeksine göre işlem yapabilmek için, bir j int değeri tanımladım ve sıfıra eşitledim. İlk if kontrolümde çarpma işlemini aradım ve j. indeks çarpma işaretine eşit olduğunda o işaretin arkasında ve önünde bulunması gereken sayıları yani Sayilar listesinin j. indeksini ve j+1. indeksini birbirleriyle çarptım ve tekrardan j. indekse atadım. Böylece daha önce o indekste bulunan sayıyı (zaten işleme kattığım için ihtiyacım kalmamıştı) kaldırmış oldum. İşleme giren diğer elemanı da sildikten sonra Isaretler listesindeki \* ifadesini de kullandığım için onu da sildim. Aynı işlemi bölme işareti için de uyguladım. Başka bir işaret kullanılmadıysa işlemin tamamlanmış olması gerekir. Bunu kontrol edip eğer elimizde tek bir sayı kaldıysa bunu kullanıcıya göstermesini istedim. Hala yapılacak işlemler varsa tekrardan döngü hazırlayıp bu defa aynı işlemleri toplama ve çıkarma işlemleri için yaptım. Bu defa işlemin tamamlanmış olması gerekli. Kullanıcıya elde kalan sayıyı ilettim. Ancak bu program negatif sayıları ve parantez işaretlerini tanımayacak. Bu sebeple kullanıcının girdiği yanlış bir ifade varsa işlem yapılmayacak. Bu hatayı da catch yapısını kullanarak ekranda hatalı giriş yapıldığını ifade eden bir metinle gösterdim.

**Sonuç**

Bu ödev ile birlikte C# programlama dilinde List yapısını kullanarak Hesap Makinesi programını geliştirdim. Karşıma çıkan problemleri araştırarak kaynaklarından buldum ne çözümü için algoritmalar kullandım. Proje ideal olmasa da bizden istenen bütün işlemleri yapmaktadır.

**Referanslar**

1. <https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1?view=netframework-4.8>
2. <https://docs.microsoft.com/en-us/dotnet/api/system.char.isdigit?view=netframework-4.8>