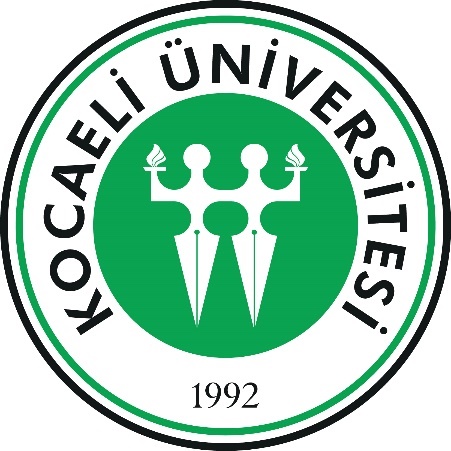
****

**KOCAELİ ÜNİVERSİTESİ**

**KOCAELİ MESLEK YÜKSEKOKULU**

**BİLGİSAYAR PROGRAMCILIĞI**

**PROJE BİTİRME TEZİ**

**C# İLE ANAOKULU TAKİP PROGRAMI**

**ŞAHİN BARAN EŞSİZ**

**KOCAELİ 2020**

**PROJE**

**ŞAHİN EŞSİZ**

**2**

**0**

**2**

**0**

8

cm

8

cm

1. ÖNSÖZ VE TEŞEKKÜR

Günümüz dünyasında artık her şey elektronik ortama taşınıyor. Bu yüzden artık okul sistemlerinin de elektronik ortama taşımak gerekiyor.

Yapım aşamasında programlama alanında beni geliştiren uygulamayı kullandığım dilin öğretiminde bana yardımcı olan UDEYM ACADEMYE teşekkür ederim. İlerlettiğim projede kontrol edip fikir ve yönler veren projede destek çıkan değerli bölüm hocam “Mustafa OF” hocama teşekkür ederim.

Ocak – 2020 Şahin Baran EŞSİZ

**C# İLE ANAOKULU TAKİP PROGRAMI**

2. ÖZET

Günümüz dünyasında artık her şey elektronik ortama taşınıyor. Bu yüzden artık okul sistemlerinin de elektronik ortama taşımak gerekiyor.

Benim bu yazılarımda size sunacağım bu dilin bilgisayar içinde yaptığım anaokulu takip uygulamasının nasıl yapıldığını ne işe yaradığını nasıl kullanılacağını anlatmak ve onu savunmaktır.

**İÇİNDEKİLER**

[1. ÖNSÖZ VE TEŞEKKÜR iii](#_Toc29956331)

[2. ÖZET vi](#_Toc29956332)

[3. GİRİŞ 6](#_Toc29956333)

[4. C # NE İŞE YARAR ? 7](#_Toc29956334)

[5. .NET FRAMEWORK NEDİR? 14](#_Toc29956335)

[6. C# İLE NELER YAPİABİLİR 14](#_Toc29956336)

[7. Webform uygulaması 1](#_Toc29956343)5

[8. MOBİL PROGRAMLAMA 12](#_Toc29956345)

[9. WEB SERVİSLERİ 15](#_Toc29956356)

[10. DLL YAZMA 16](#_Toc29956357)

[11. DLL 16](#_Toc29956358)

[12. C# IN C, C++ VE JAVA İLE BAĞLANTISI 31](#_Toc29956361)

[13. C# GENEL ÖZELLİKLERİ 41](#_Toc29956362)

[14. C# KODLARI DERLEME: 54](#_Toc29956369)

[15. ASSEMBLY KAVRAMI: 61](#_Toc29956376)

[16. VISUAL STUDIO.NET: 66](#_Toc29956381)

[17. SOLUTİON EXPLORER PENCERECİĞİ: 67](#_Toc29956389)

[18. CLASS VİEW PENCERECİĞİ 67](#_Toc29956390)

[19. KODLAMA: 68](#_Toc29956391)

[20. VİSUAL STUDİO.NET VE DOSYALAR: 69](#_Toc29956392)

[21. HATA AYIKLAMA: 69](#_Toc29956393)

[22. SOLUTİON KAVRAMI: 71](#_Toc29956394)

[23. DERLEME: 73](#_Toc29956395)

[24. .C# – Sınıflar: 74](#_Toc29956396)

[25..EKRAN ÇIKTISI: 75](#_Toc29956397)

[26. C# Metotlar ve Metot Örnekleri: 76](#_Toc29956398)

[27. C#' ta kullanılan operatörler ve işlevleri 78](#_Toc29956399)

[28. SQL NEDİR ? 81](#_Toc29956400)

[29. SQL NASIL ÇALIŞIR? 1](#_Toc29956333)

[30. SQL NASIL ÇALIŞIR? 1](#_Toc29956334)

[31. SQL SİSTEMLERİ NASILDIR? 9](#_Toc29956335)

[32. SQL EĞİTİMLERİ NEDİR 9](#_Toc29956336)

[33. NEDEN SQL KULLANILMALIDIR 11](#_Toc29956343)

[34. SQL YAZILIMLARI 12](#_Toc29956345)

[35. MYSQL 15](#_Toc29956356)

[36. MSSQL 16](#_Toc29956357)

[37. MİCROSOFT SQL Server 16](#_Toc29956358)

[38. Firebird 31](#_Toc29956361)

[39. Access 41](#_Toc29956362)

[40. MS SQL NEDİR? ÖZELLİKLERİ 54](#_Toc29956369)

[41. SQL SERVER MANAGEMENT STUDIO NEDİR 61](#_Toc29956376)

[42. DÜŞÜK SEVİYELİ PROGRAMLAMA DİLİ 66](#_Toc29956381)

[43. YÜKSEK SEVİYELİ PROGRAMLAMA DİLİ 67](#_Toc29956389)

[44. ÇOK YÜKSEK SEVİYELİ PROGRAMLAMA DİLLERİ 67](#_Toc29956390)

[45. C++ DİLİ NEDİR 68](#_Toc29956391)

[46. C++ NE İŞE YARAR 69](#_Toc29956392)

[47. C++ ÖĞRENME 69](#_Toc29956393)

[48. C PROGRAMLAMA DİLİ NEDİR 71](#_Toc29956394)

[49. ALGORİTMA NEDİR 73](#_Toc29956395)

[50. ALGORİTMA NE İÇİN KULLANILIR 74](#_Toc29956396)

[51. ALGORİTMA VE PROGRAMLAMA BAĞLANTISI 75](#_Toc29956397)

[52. KAYNAKLAR 41](#_Toc29956362)

[53. EKLER 54](#_Toc29956369)

[*Şekil1 Web Form Örnek* 10](#_Toc29961129)

[Şekil *2.1 64 K DİSK AYRIMA* 17](#_Toc29961130)

[Şekil 2.2 İnstallation tabında seçilecek yerin gösterimi 19](#_Toc29961131)

[*Şekil 2.3 Evaluation seçim yeri gösterimi* 20](#_Toc29961132)

[*Şekil 2.4 Lisans tabı gösterimi* 20](#_Toc29961133)

[Şekil 2.5 Kurulun öncesi gerekli güncellemeler yeri 21](#_Toc29961134)

[Şekil 2.6 Gerekli bileşenler kontrol sağlama yeri 21](#_Toc29961135)

[Şekil 2.7 SQL özellik seçim yeri 22](#_Toc29961136)

[Şekil 2.8 Server ismimizi belirlediğimiz yer 22](#_Toc29961137)

[Şekil 2.9 Server 2017 çalıştıracak hesapları belirler 23](#_Toc29961138)

[Şekil 2.10 Dil karakter ve sat ayarı kısmı 23](#_Toc29961139)

[Şekil 2.11 Kimlik doğrulama metodu seçme yeri 24](#_Toc29961140)

[Şekil 2.12 Log, db bileşenlerin kurulum yeri 24](#_Toc29961141)

[Şekil 2.13 Gerekli sayıda Temp yükleme ekranı 25](#_Toc29961142)

[Şekil 2.14 Yapılan seçimlerin özeti 25](#_Toc29961143)

[Şekil 2.15 Kuruluyor 26](#_Toc29961144)

[Şekil 2.16 Kurulum sonrası gelen ekran 26](#_Toc29961145)

[Şekil 2.17 Server Kurulmuştur 27](#_Toc29961146)

[Şekil 2.18 SSM de gelen ilk ekran 28](#_Toc29961147)

[Şekil 2.19 Kurulurken alınan görüntü 28](#_Toc29961148)

[Şekil 2.20 Kurulum başarıyla tamamlandı ekranı gösterimi 29](#_Toc29961149)

[Şekil 3 Server Adımız ve giriş ekranı 29](#_Toc29961150)

[Şekil 3.1 Server ve veri tabanı oluşumu 30](#_Toc29961151)

[Şekil 4 Tarayıcı arama yerine Microsft’un sitesine giriyoruz 31](#_Toc29961152)

[Şekil 4.1 Visual Studio Community seçilecek yer 31](#_Toc29961153)

[Şekil 4.2 Dosya tarama ekranı 32](#_Toc29961154)

[Şekil 4.3 İndirme ekranı 32](#_Toc29961155)

[Şekil 4.5 33](#_Toc29961156)

[Şekil 4.6 Kurulum sonu 34](#_Toc29961157)

[Şekil 5 Form Uygulaması Seçimi 35](#_Toc29961158)

[Şekil 5.1 Proje Örneği 35](#_Toc29961159)

[Şekil 5.2 Sözlükte eklenen araçlar 36](#_Toc29961160)

[Şekil 5.3 Hakkında Formu 38](#_Toc29961161)

[Şekil 5.5 SQL Management veri tabanı ve tabloları 39](#_Toc29961162)

[Şekil 6 Visual Studio üzerinde veri ekleme komutları gösterimi. [1] 48](#_Toc29961163)

[Şekil 6.1 Visual Studio üzerinde veri ekleme komutları gösterimi [2] 48](#_Toc29961164)

[Şekil 6.2 Visual Studio üzerinde veri ekleme komutları gösterimi [3] 49](#_Toc29961165)

[Şekil 6.3 Visual Studio üzerinde veri ekleme komutları gösterimi [4] 49](#_Toc29961166)

[Şekil 6.4 Visual Studio üzerinde veri ekleme komutları gösterimi [5] 50](#_Toc29961167)

[Şekil 7 Eşleşen kelime siyah anlamı mor 52](#_Toc29961168)

[Şekil 8 Tıp terimi kelime anlamlarının çıkması ve renklendirilmesi 59](#_Toc29961169)

[Şekil 8.1 İngilizce-Türkçe kelime anlamlarının çıkması ve renklendirilmesi 60](#_Toc29961170)

[Şekil 9 Hakkında butonuna tıklandığında hakkında formu geliyor 61](#_Toc29961171)

[Şekil 9.1 Türkçe-İngilizce Kelime eklenmiştir 65](#_Toc29961172)

[Şekil 10 Gri olan sözlük değiştir butonuna tıkladıktan sonra yeşil olacaktır [1] 71](#_Toc29961173)

[Şekil 10.1 Programın arka planı yeşil renk olmuştur [2] 71](#_Toc29961174)

[Şekil 11 Uygulama yazı tipi Arial Black 36 boyutundadır 72](#_Toc29961175)

3. GİRİŞ

Öncelikle C# dilini anlatmak isterim. C# yazılım sektörü içerisinde en sık kullanılan iki yazılım dili olan C ve C++ etkileşimi ile türetilmiştir. Ayrıca C#, ortak platformlarda taşınabilir bir programlama dili olan Java ile pek çok açıdan benzerlik taşımaktadır. En büyük özelliği ise .Net Framework platformu için hazırlanmış tamamen nesne yönelimli bir yazılım dilidir. Yani nesneler önceden sınıflar halinde yazılıdır. Programcıya sadece o nesneyi sürüklemek ve sonrasında nesneyi amaca uygun çalıştıracak kod satırlarını yazmak kalır.

Sizlere C# ı ve farkları anlattıktan sonra İngilizce-Türkçe Türkçe-İngilizce sözlük yapımını anlatacağım.

**Microsoft** tarafından geliştirilen C# ,C++ ve Visual Basic dillerinden yer alan tutarsızlıkları kaldırmak için geliştirilmiş bir dil olmasına rağmen kısa süre içerisinde nesne yönelimli dillerin içinde en gelişmiş programlama dillerinden biri olmayı başarmıştır.

Ayrıca gelişmiş derleyicisi ile hata olasılığını ortadan kaldırmaktadır. Yazılan program çalıştırıldıktan sonra derleyici tarafından algılanan sınıf ve söz dizimi hataları yazılımcıya ayrı bir ekranda ayrıntısı ile gösterili ve yazılımcı bu hata penceresinden hataları tespit ederek kolayca düzeltebilir.

Ayrıca pek çok programcı tarafından kullanılan bir dil olmasından ötürü takıldığımız noktalarda uzman programcılardan yardım almak oldukça kolaydır.

4. C # NE İŞE YARAR?

Yeni nesil programlama dili olan C# programlama konusunda yenilikleri de beraberinde getirmektedir. C# sunucu ve gömülü sistemleri için tasarlanmıştır. Microsoft tarafından geliştirilen C# .NET ortamında kullanılmak üzere geliştirilmiştir. Dilin tasarlanmasında **Anders Hejberg** öncülük etmiştir. Zamanla gelişen bu programlama dili artık gerek masaüstü olsun gerek web uygulamaları olsun çoğu yerde kullanılmaktadır.

4.1.C# Değişkenler ve Veri Tipleri

Değişkenler, girdiğimiz değerleri alan veya programın çalışmasıyla bazı değerlerin atandığı veri tutuculardır. Değişken tanımlama ise girilecek veya programın getireceği verinin ne olduğuna bağlı olarak değişken türünün belirlenmesidir. Örneğin girilecek verinin bir tamsayı olması halinde o değişkeni de tamsayı olarak tanımlamak gerekir. Değişik türler için bu işlem farklı farklı tekrarlanmalıdır.

**Tanımlamada Dikkat Edilecek Durumlar**

* Bir değerleri olmalıdır veya mutlak suretle null olarak tanımlanmalıdır.
* Tanımlama yaparken büyük-küçük harf ayrımı vardır.
* Değişken adları rakamla başlayamaz.
* Class, namespace ve kontrol isimleri gibi program tarafından kullanılan isimler verilemez.
* Aynı kod bloğu içerisinde aynı isimden birden fazla değişken tanımlanamaz.
* Özel karakter içermez örneğin /,\*,-,+.
* Boşluk kullanılamaz.
* Türkçe karakter kullanılmamalıdır. Ş, ğ,ü,ö,ç,ı gibi…
* Özel sözcükler (if, else, random,  vb.) kullanılmaz.

**Değişken Tanımlama**

Değişken tanımlamaları şu şekilde yapılmaktadır.

(Değişken Türü)   ( Değişen Adı)  =  (Değeri)

**Değişken Türleri**

**Sayısal Türler;**

1. Sbyte: işaretli tam sayı
2. Short: işaretli tam sayı
3. Int: işaretli tam sayı
4. Long: işaretli tam sayı
5. Byte: İşaretsiz tam sayı
6. Ushort: İşaretsiz tam sayı
7. Uint: İşaretsiz tam sayı
8. Ulong: İşaretsiz tam sayı
9. Float: Tek kayan sayı
10. Double: Çift kayan sayı
11. Decimal: Ondalıklı sayı

**Metinsel Türler;**

* Char: Tek bir karakteri tutar
* String: Metin tutar

**C#'ta hem metin hem de sayısal olmayan türler de vardır:**

**Bool:** Koşullu yapılarda kullanılır. Bool türünden değerlere true; false gibi ifadeler örnek verilir.

**Object:** Bu değişken türüne her türlü veri atanabilir.

**Var:** C# 3.0 da artık tür belirtmeksizin değişken tanımlamamıza olanak sağlayan yenilikler mevcut. Ancak bu değişkenlerin özelliği object değişkenler gibi referans tipli değişkenler değildir. Bunun yerine değeri atanırken tipinin belirkendiği değişkenlerdir.

**Datetime**

İçinde zaman barındıran değişken tipidir.

C# Döngüler

Programlama dilinde döngüler belirtilen komut satırlarının tekrar tekrar çalıştırılmasını sağlayan yapılardır. For döngüsü genellikle üç ifade ile kullanılır. Bu ifadelerden ilkinde döngü değişkeni için bir başlangıç değeri verilir. İkinci ifadede döngü değişkenin değerine bağlı bir koşul belirlenir bu koşul sağlandığı sürece döngü içerisindeki belirtilen işlemleri tekrar eder. Üçüncü ifade de çoğunlukla döngü değişkeninin artış yada azalış miktarı için kullanılır.

For Döngüsü;

for(ifade1;kosul;ifade2)

{

komut1;

komut2;

}

for (int i = 1; i <= 5; i++)

{

Console.WriteLine(i);

Yukarıdaki döngüde öncelikli olarak i değişkeninin başlangıç değeri olan 1 kurulur. Daha sonra koşul operatörü olan i<=5 ifadesi çalışır. Bu ifadenin sonucu eğer ***Tru***e ise döndü satırları çalıştırılır. Döngü içeriği bittikten sonra i++ arttırma operatörü çalışır ve i değişkenin değeri 2 olur. Bu değer tekrar i<=5 için çalıştırılır. Ve sonuç ***True*** olduğu sürece döngü devam eder. i değişken değeri bu örnek için 6 olduğu durum da i<=5 ifadesinin sonucu ***False*** olur ve döngü kodu işletilmeden döngü sonlandırılır.

HATIRLATMA!

***for*** döngüsünün parantezleri içindeki ” for (int i=0;i<5;i++) ” iki ifade ve bir koşulun istenirse bir tanesi, istenirse bazıları, istenirse de tamamı boş bırakılabilir; ancak noktalı virgüller mutlaka yazılmalıdır. For(;;)//Sonsuz döngü  
Tahmin edebileceğiniz gibi for döngüsünün içinde veya “for (int i=0;i<5;i++) ” kısmında tanımlanan herhangi bir değişken döngünün dışında kullanılamaz. Bu bütün döngüler ve koşul yapıları için geçerlidir

While Döngüsü

While döngüsü, belirlenen komut bloğunu koşul sağlandığı sürece çalıştıran temel döngü yapılarından bir tanesidir. Bu döngü, basit bir kullanıma sahip olması sebebiyle tercih edilir. While döngüsünün genel formu aşağıdaki gibidir.

While(koşul){

//Döngü gövdesi-Komutlar

}

Döngü gövdesi koşul deyimi true olduğu sürece çalıştırılır, koşul false olduğunda ise döngüden çıkılır. While döngüsünde döngü değişkeni bulunmaz, bunun yerine while bloğuna girmeden başlangıçta değeri bilinen bir değişken kullanılır. Koşul ifadesiyle ilişkili olan bu değişkenin değeri döngü bloğu içerisinde değiştirilir. Konunun daha iyi anlaşılabilmesi için sorular yazarak çözümlerini inceleyelim.

İnt i =1;

While (i<=10)

{

Console.WriteLline(“Merhaba Dünya”);

İ++;

}

Console.Readkey();

Adım1: “i” değişkenini döngü bloğu dışarısında tanımladık ve başlangıç değeri olarak 1 değerini atadık, bu değişken sadece döngü değişkeni olarak görev yapacak.

Adım2: while döngüsü içerisinde yukarıda tanımladığımız “i” değişkenine bağlı olan bir koşul belirledik. i<=10 yani “i” değeri 10’dan küçük olduğu için döngü bloğu çalıştırıldı.

Foreach Döngüsü:

**Foreach** listeler ya da diziler üzerinde işlem yapmak için kullanılan **döngü** yapısıdır. Özellikle eleman sayısının bilinmediği durumlarda büyük kolaylık sağlamaktadır.  
Aşağıdaki örnekte **Foreach**döngüsü kullanılarak **string** bir **dizi** oluşturulup bu dizinin elemanlarının yazdırılması işlemi yapılmıştır.

class Program

    {

        static void Main(string[] args)

        {

            string[] isimler = {"Ahmet CANSEVER","Emirhan SOYLU","Elif SÜTÇÜ",

                                "Hakan ALTIN","Sema ÇALIŞKAN"};

            foreach(string eleman in isimler)

            {

                Console.WriteLine(eleman);

            }

            Console.ReadKey();

        }

Programı çalıştırdığımızda isimler dizisindeki elemanların ekrana yazılır.

Do While Döngüsü:

While döngüsünde olduğu gibi döngünün kaç kez döneceğinin tahmin edilemediği durumlarda kullanılır.

While döngüsünden en önemli farkı şartın döngünün sonunda kontrol edilmesidir[.](http://www.btdersleri.com/ders/Do-While-D%C3%B6ng%C3%BCs%C3%BC-ve-%C3%96rnekleri) Bu nedenle de do while döngüsü kullandığımızda döngü içerisindeki işlemler en az bir kere yapılacak demektir.

Yapısı şu şekildedir:

**do**

**{**

**İşlemler...**

**}**

**while( döngünün çalışma koşulu );**

**int toplam =0, sayi;**

**do**

**{**

**Console.Write(“Sayı);**

**Sayi=int.Parse(Console.ReadLine());**

**Toplam+=sayi;**

**While (sayi!=sayi;**

**} while(sayi!=0);**

**Console.WriteLine(toplam);**

**Console.Readkey();**

**Bu kod dizini kullanıcı 0 girinceye kadar sayı istemeye devam eden, girilen sayıları toplama ekleyen , 0 girdiğinde döngüyü sonlandırıp toplamı ekrana yazan program**

**Koşullar**

**İf-Else:**

If-else yapısı diğer programlama dillerinde olduğu gibi C#’dada, koşullu dallanma ifadesi olarak kullanılmaktadır. Programda alternatifli durumlarla karşılaşıldığında, duruma özgü komutların işletilmesini sağlar. Dallanma işlemlerinde programcıların en çok tercih ettiği yapıdır. If yapısının kendi içerisinde birçok farklı kullanımı bulunmaktadır, bu kullanımları teker teker örneklerle incelemekte fayda var. If bloğu kendi başına kullanılabileceği gibi else bloğuyla kullanımı daha yaygındır.

If-Else ifadesinin genel formu aşağıdaki gibidir.

If (koşul)

{

Koşul doğruysa çalışacak komutlar

}

else

{

Koşul yanlışsa çalışacak komutlar

}

Console.Write(“Bir sayı giriniz”);

İnt sayi=Convert.ToInt32(Console.ReadLine());

İf (sayi %2==0)

Console.WriteLine(“Sayi Çifttir”);

Else

Console.WriteLine(“Sayi Tektir”);

Bir sayının ikiye kalansız bölünebilmesi o sayının çift olduğu anlamına gelir, bunu bilmek örneğin anlaşılabilmesi açısından önemlidir. Eğer sayi değişkeninin ikiye bölümünden kalan 0 ise; if bloğundaki komut çalıştırılır, ekrana “Sayı Çifttir” yazılır. Değilse else bloğundaki komut çalıştırılır ekrana, “Sayı Tektir” yazılır.

**NOT:**

Eğer if ve else bloklarından sonra tek bir satır kod yazılacaksa süslü parantez kullanmaya gerek yoktur.

Console.Write(“Öğrencinin notunu giriniz”);

İnt not=ConverToInt32(Console.ReadLine());

İf (not>=50)

Console.WriteLine(“Geçti”);

Else

Console.WriteLine(“Kaldı”);

Console.Readkey();

Bu örneğin sonucu öğrenci 50 not üzerinde ise “geçti” değilse “kaldı” yazdıracaktır.

Koşul ifadeleri sadece sayısal değerleri kontrol etmek için kullanılmazlar. String, char gibi metinsel değerlerin kontrolü içinde sıklıkla kullanılırlar. Yukarıda ki örnekler ve anlatılanlar okuyucu sözlük içinde ki tabirleri daha rahat anlayabilsin diyedir.

5. .NET FRAMEWORK NEDİR?

C# ve .Net Framework bazı kişiler tarafından tek bir kavram olarak algılanmaktadır. Fakat bu iki kavram birbirlerinden tamamen farklı amaçlar için geliştirilmiştir**. C# nesne yönelimli bir programlama diliyken, .Net Framework ise C# için geliştirilmiş bir çalışma ortamıdır.**

Aslında C# dili, Microsoft tarafından .Net platformu için kod geliştirmek amaçlı tasarlanmış ve C# içerisindeki tüm kütüphaneler .Net platformu içinde tanımlanmış kütüphanelerdir.

Java’dan önce, geliştirilen yazılımlar makine koduna çevrilerek çalıştırılırdı. Java ise program kodlarını önce sayı sistemine çevirir. Sonrasında Java bu kodları işletim sisteminin istediği koda çevirerek programın çalışmasını sağlar.

.Net Platformu da Java diline benzer bir çalışma mantığı izleyerek kodları çalışabilir hale getirmektedir.

.Net platformunda kod ilk önce Microsoft Ara Dili olarak isimlendirilmiş dosya haline dönüştürülür bu dosya içerisinde derlenen kodların Microsoft’un standart haline getirdiği bir makine dili dönüştürür. Bu ara dil de saklanan dosyalar çalıştırılmak istendiğinde ise CLR adı verilen sistem MSSIL kodlarını çalıştırır.

6. C# İLE NELER YAPILABİLİR

Windows Form Uygulamaları

Grafiksel kullanıcı ara yüzü olan GUI olarak tanımlanır. Windows form uygulamalarını, konsol uygulamalarından ayıran en önemli özelliği de budur. Windows form uygulaması oluşturmak için Toolbox adı verilen araç kutusundan nesnel araçlar sürükle bırak yöntemi ile kolayca oluşturulabilir. Şekil 1 de Windows Form Uygulamaların basit gösterimidir.

Windows form uygulamaları için veri tabanı bağlantısı kurabilmek için kullanılan .Net kütüphanesinin adı Ado.Net olarak tanımlanmıştır. Bu konun üzerinde çalışmak isteyenlerin internet üzerinde Ado.Net olarak tarama yapmaları gerekmektedir.

7. Web Form Uygulaması (Asp.Net)

Web form uygulamaları ise Ado.Net yerine ASP.Net sayfa yapısı ile çalışmaktadır. Web form uygulamaları temel olarak Windows Form Uygulaması gibi düşünülebilir. Fakat aralarında temel farklılıklar vardır. Web Form uygulamaları internet üzerinden erişilebilen form uygulamalarıdır. Bundan dolayı uygulamalar kişisel bilgisayar yerine web tarayıcısı ile çalışmaktadır.

ASP.Net kullanılarak oluşturulan bir web uygulaması içerisinde bir yada birden daha çok ASP.Net sayfası olabilir. Web formları , .Net kütüphanesinde özel etiketler içeren bir HTML sayfalarıdır. Web formları, uzantısı .aspx olarak işlenir. Ayrıca her web form uygulamasında web.config ile belirtilen bir yapılandırma dosyası vardır. Bu dosya XML biçimdedir ve web uygulamasının güvenlik, önbellek yönetimi gibi ilgili işlerin bilgilerini içerir.

8. MOBİL PROGRAMLAMA

Windows Phone işletim sistemleri için uygulama geliştirme kısmıdır. Eğer Windows Phone için uygulama geliştirmek istiyorsanız bu kısmı kullanmanız gerekmektedir. Xamarin ile birlikte mobil programlamaya giriş yapabilirsiniz.

9. WEB SERVİSLERİ

Kısaca veri iletimi için kullanılan sistemlerdir diyebiliriz. Biraz daha açacak olursam ise Evrensel veri transfer metodu sayılan XML ve http kuralları ile internet ağı ile

dünyanın her hangi bir yerine veri taşıyan sistemlerdir. Web servisi işlemlerinde veriler bir başka kullanıcıya gönderildiği için güvenlik ve adresleme gibi konular oldukça önemlidir. Microsoft, C# ta web servisleri kullanımını güvenlik kılan pek çok işlemler yapmaktadır.

10.DLL YAZMA

İlk olarak DLL nedir ? diye soranlara DLL hakkında bilgi vereceğim. Sonrasında ise DLL i açıklayacağım. Dynamic-link library, Microsoft'un paylaşılan kitaplık konseptini Microsoft Windows ve OS/2 işletim sistemlerinde uygulamasıdır. Bu kitaplıklar genellikle DLL, OCX veya DRV dosya uzantısına sahiptir

11. DLL

Dinamic Link Library ifadesinin kısaltılmasıdır. Yani dinamik kütüphaneler oluşturmak için kullandığımız yapıdır. Örnek verecek olursam C# ta bir yapıyı sürekli kullanıyorsunuz böyle bir durumda hem kod sayfasında bir karmaşa olacak hem de sürekli aynı kodları yazmak zorunda kalacaksınız. İşte bu iki sorunu ortadan kaldırmak için DLL yazarsınız. DLL yazdıktan sonra uygulama sayfasında sürekli aynı kodları yazmak yerine yazılan DLL i çağırarak ilgili kodları çalıştırırız.

12. C# IN C, C++ VE JAVA İLE BAĞLANTISI VAR MIDIR ?

Her ne kadar isim benzerliği olsa da C ile pek benzer yanlarının olduğu söylenemez. Aralarındaki en büyük fark nesne yönelimli programlamadır.

C# nesne yönelimli programlama desteklerken, C nesne yönelimli programlamayı desteklememektedir. Ortak yanlarından belki de en önemlisi benzer söz dizime sahip olmalarıdır. Ana yapıları itibariyle C# programlama ile C ile programlamaya göre daha farklıdır.

C++ dili ile ise nesne yönelim programlama özellikleri aynıdır. Ayrıca sözdizimleri de benzerlik göstermektedir.

C# daha çok Java ve C++ programlama dillerinden esinlenerek oluşturulmuştur. Birebir aynı olmasa da çoğu özellikleri benzerlik taşımaktadır.

13. C# GENEL ÖZELLİKLERİ

* Basit ve genel amaçlı bir dil
* Platformlardan bağımsız bir dil
* Kullanımı kolay modern bir dil
* %100 Nesneye yönelik programlama dili
* Gelecek tabanlı uygulamalar için uygun ve sağlam bir dil
* Çöp toplama ve kaynak yönetimi ile programların belleğini çok daha verimli kullanan bir dildir.

Başlamadan önce sözlüğün yapımı için C# dilinin yanında Veri tabanı kullanımını yani SQL komutlarını da bilmek gerekir. SQL tanımını ve SQL mantığını bilmeyenler için şimdi sizlere SQL i basit bir şekilde tanıtalım.

14. C# KODLARI DERLEME:

Son bir-iki bölüm dışındaki bütün kodları Not Defteri'nde (notepad) yazacağız. Aslında Visual Studio adında son derece gelişmiş ve işimizi son derece kolaylaştıran bir editör yazılımı var. Ancak bizim buradaki amacımız gelişmiş kurumsal yazılımlar geliştirmek yerine C#'ı tam anlamıyla öğrenmek olduğu için bu gelişmiş yazılımı kullanmayacağız. Not Deferi'nde yazdığımız kod dosyasına "cs" uzantısı verip .Net Framework programıyla birlikte gelen csc.exe derleyicisi ile derleyeceğiz. Bu derleyici komut satırında çalışıyor ve dolayısıyla da kodumuzun derlenmesi için komut satırında kod yazacağız. Yani .Net Framework programı hem kodumuzu derlemek için, hem de programımızın çalışması için gerekli. .Net Framework'u herhangi bir download sitesinden ya da Microsoft'un resmî sitesinden ücretsiz olarak indirip bilgisayarınıza kurabilirsiniz. Vista kullanıyorsanız çok büyük ihtimalle, XP kullanıyorsanız da bir ihtimalle .Net Framework sizde zaten kuruludur. Ancak işinizi garantiye almak isterseniz veya son sürümünü edinmek isterseniz tekrar kurabilirsiniz.

.Net Framework'u kurduktan sonra bilgisayarımıza bir ayar yapmamız gerekecek. Bu ayara "path ayarlama" diyoruz. Bu ayarı yapmamızın sebebi komut satırında hangi klasörde olursak olalım csc.exe'ye erişebilmemiz. Bunun için şunları yapın:

Bilgisayarım'a sağ tıklayın.

* "Özellikler"i seçin.
* "Gelişmiş" sekmesine gelin.
* "Ortam Değişkenleri" butonuna tıklayın.
* "Sistem değişkenleri" kısmındaki "Path"a çift tıklayın.

Bu pencere burada kalsın, şimdi C:\WINDOWS\Microsoft.NET\Framework klasörüne gidin. Oradaki klasörlerin herbirinin içine tek tek bakın. Hangisinin içinde csc.exe varsa o klasörün adres çubuğundaki yolu kopyalayın.

Şimdi önceki açtığımız "Sistem Değişkenini Düzenle" penceresinin "Değişken Değeri" kısmının sonuna ; işaretini koyup yolu yapıştırın.

Bütün pencerelerden "tamam" diyerek çıkın.

Artık Not Defteri'nde yazdığımız kodu csc.exe derleyicisi ile derleyebiliriz. Örneğin "cs" uzantısı verdiğimiz kaynak kodumuzun adı "deneme.cs" olsun. Komut satırını açıp, kaynak kod dosyamızın olduğu klasörü aktif hâle getirip, csc deneme.cs yazıp entera basarak kodumuzu derleriz. Oluşan exe dosyamız kaynak kodumuzla aynı klasörde ve deneme.exe adıyla oluşur. Eğer aktif klasörde zaten deneme.exe diye bir dosya varsa eski dosya silinip yeni dosyamız kaydedilir. Programımızın kaynak kodun adından farklı bir adla oluşmasını istiyorsak

“csc /out:YeniAd.exe deneme.cs”

komutunu veririz. Bu örnekte programımız YeniAd.exe adıyla oluşacaktır.

15.ASSEMBLY KAVRAMI:

Derlenmiş exe ve dll dosyalarına assembly denir. Çalışan programımız içinde bir assembly olabileceği gibi birden fazla assembly de olabilir. Programımızda birden fazla assembly olması demek programımızın ilişkide olduğu bir ya da daha fazla dll dosyası mevcut demektir. System isim alanındaki AppDomain sınıfı çoklu assembly işlemleriyle ilgilenirken, System.Reflection isim alanındaki Assembly sınıfı ise tek bir assembly ile ilgili işlemler yapılmasını sağlar. Şimdi bu iki sınıfın çeşitli özelliklerini inceleyelim:

* AppDomain sınıfının static CurrentDomain özelliği o an üzerinde çalışılan assembly grubunu bir AppDomain nesnesi olarak döndürür.
* AppDomain sınıfının static olmayan GetAssemblies() metodu, ilgili AppDomain nesnesindeki bütün assemblyleri bir Assembly dizisi olarak tutar.
* Assembly sınıfının static olmayan Location özelliği ilgili assemblynin sabit diskteki yolunu string olarak verir. Şimdi bu teorik bilgileri bir örnek üzerinde görelim:



Şekil 1 Assembly kodları

mscorlib.dll dosyası şimdiye kadar sıklıkla kullandığımız System isim alanı ve bu isim alanına bağlı alt isim alanlarının içinde bulunduğu dosyadır. C# derlediğimiz her dosyaya bu dosyayı otomatik olarak ilişkilendirir. Bu dosyanın yolu kullandığınız .Net Framework sürümüne bağlı olarak sizde değişebilir.

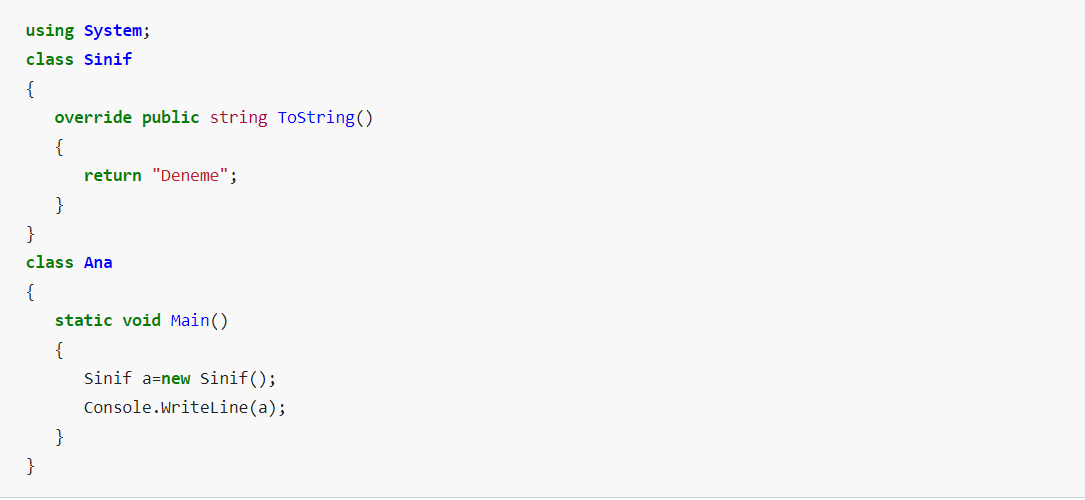
Bazen programımızın ilişkiye girdiği dll dosyasının da ilişkiye girdiği başka bir dll dosyası olabilir. Bu durumda AppDomain sınıfının CurrentDomain özelliğiyle oluşturulan AppDomain nesnesine söz konusu bütün dll dosyaları dâhildir. Ancak tabii ki farklı assemblyler aynı dll dosyasını kullanıyorsa yalnızca bir tane dll AppDomain nesnesine dâhil edilir. Ayrıca AppDomain sınıfının GetAssemblies() metodu ile oluşturulan Assembly dizisinde sıra esas programda karşılaşılma sırasına göredir. Eğer programda System isim alanı kullanılmışsa ilk öğe mscorlib.dll'dir. Ondan sonra gelen öğe esas programımız, bundan sonra gelen öğeler ise karşılaşılma sırasına göre dll dosyalarıdır. Eğer herhangi bir dll dosyasını programımıza ilişkilendirir ancak söz konusu dll'i programımızda kullanmazsak söz konusu assembly programımıza dâhil değilmiş gibi hesaba katılır.

* Assembly sınıfının static GetExecutingAssembly() metodu ile o an çalışan assemblynin kendisi bir Assembly nesnesi olarak döndürülür.
* Assembly sınıfının static olmayan EntryPoint özelliği ilgili assemblynin başlangıç metodunu MethodInfo sınıfı türünden bir nesne olarak döndürür. MethodInfo sınıfını ileride göreceğiz. Başlangıç metodu çalıştırılabilir exe dosyalarında genellikle Main() metodudur. Dll dosyalarında ise başlangıç metodu yoktur. Bu yüzden bu özellik dll dosyaları için null değeri döndürür. Şimdi bu iki bilgiyi örnekleyelim:



Şekil 1.1 Assembly kodları

Yeri gelmişken söylemek istiyorum. Bir sınıfın ya da yapının bir nesnesini Console.Write() ya da Console.WriteLine() metodu ile ekrana yazdırırsak aslında o nesnenin ToString() metodunun geri dönüş değerini ekrana yazdırmış oluruz. ToString() metodu normalde object sınıfına bağlıdır ve ilgili nesnenin türünü string olarak döndürür. Ancak tabii ki istersek kendi sınıf ya da yapılarımızda bu metodu override edip ilgili nesnemizin Console.Write() ya da Console.WriteLine() ile ekrana yazdırılırken nasıl davranması gerektiğini ayarlayabiliriz. Örnek:



Şekil 1.2 Assembly kodları

Bu program sonucunda ekrana Deneme yazılacaktır. İstersek Sinif sınıfına yeni üye elemanlar ekleyip ekrana yazdırılacak şeyin nesneye bağlı olmasını sağlayabiliriz. Bu örneğimizde ekrana yazılacak şey nesneye bağlı değildir. Bu ipucundan sonra asıl konumuza dönebililiriz.

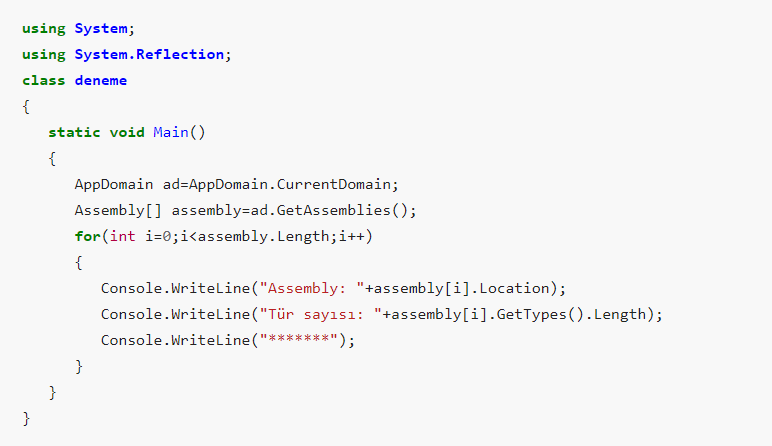
Assembly sınıfının static LoadFrom(string assembly\_yolu) metodu bir assembly nesnesi döndürür. Örnek:



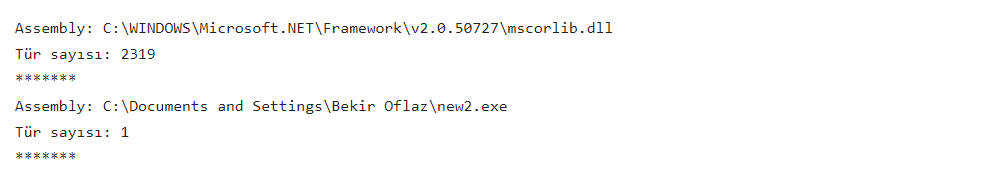
Şekil 1.3 Assembly kodları

Bu program sonucunda assemblynin başlangıç noktası olarak ekrana hiçbir şey yazılmayacaktır. Çünkü null demek yokluk demektir.

* Assembly sınıfının static olmayan GetTypes() metodu ilgili assemblydeki tüm türleri (sınıf, yapı, vb.) bir System.Type dizisi olarak döndürür. System.Type sınıfını ileride göreceğiz. Ancak şimdilik buna basit bir örnek verelim:



Şekil 1.4 Assembly kodları



Şekil 1.5 Assembly kodları çıktısı

Buradan anlıyoruz ki mscorlib.dll dosyasında tam 2319 tür varmış. Sizdeki .Net Framework sürümüne göre bu sayı değişebilir.

* Assembly sınıfının static olmayan GetType(string Tür) metodu ilgili assemblydeki belirli bir türü System.Type nesnesi olarak tutar. Örnek:



Şekil 1.6 Assembly kodları çıktısı

16.VISUAL STUDIO.NET:

Şimdiye kadar programlarımızı Not Defteri'nde yazdık ve csc.exe derleyicisi ile derledik. Böyle yapmamızın sebebi C# dilini tam anlamıyla öğrenmemizdi. Artık C# dilini bayağı öğrendik. Artık gelişmiş özelliklerle donatılmış, tümleşik bir derleyiciye sahip olan Visual C# Express Edition yazılımına geçebiliriz.

17.SOLUTİON EXPLORER PENCERECİĞİ:

Yeni bir Console Application projesi oluşturun. Sayfanın asıl kısmına kodlarımız otomatik olarak gelecektir. Sayfanın sağ kısmında ise Solution Explorer penceresi bulunur. Solution Explorer pencereciği yandaki gibidir. Bu pencerede kodumuzla ilgili dosyalar ve kodumuza refere ettiğimiz DLL dosyaları bulunur. Referans DLL'leri "References" klasörünün içindedir. Daha önce oluşturduğumuz bir DLL dosyasını programımıza refere etmek için "References" klasörünü sağ tıklayıp "Add Reference"i tıklarız. Açılan pencereden ".NET" sekmesinden .NET Framework'un kendi DLL'lerini, "COM" sekmesinden Windows'un COM bileşenlerini, "Browse" sekmesinden kendi DLL'lerimizi, "Projects" sekmesinden başka projelerin DLL'lerini ve son olarak "Recent" sekmesinden son kullandığımız DLL'leri programımıza refere edebiliriz. Aynı anda birden fazla DLL seçebiliriz.

18.CLASS VİEW PENCERECİĞİ

Menü çubuğundaki "View" menüsünden Class View pencereciğini görünür hâle getirin. Class View pencereciği yandaki gibidir. Class View pencereciğinde o anki projemizdeki ve projemizin referanslarındaki tüm isim alanları, türler ve üye elemanlar hiyerarşik bir yapıda listelenir. Bir türdeki üye elemanları görmek için türü tıklamamız yeterlidir. Türlerde türeme söz konusuysa türeyen üye elemanları da görebiliriz. Projemizde ilgili isim alanı, tür ya da üye elemana gitmek için sadece onun Class View penceresindeki karşılığına çift tıklamak yeterlidir. Türler ve üye elemanlar arasında dolaşmanın diğer bir yoluysa kaynak kodun hemen üstündeki drop down menüleri kullanmaktır.

19.KODLAMA:

Kaynak kodda herhangi bir yerde CTRL+boşluk tuş bileşimini kullanırsak orada kullanabileceğimiz her türlü kod (anahtar sözcük, tür, üye eleman, nesne vb.) bir menüde listelenir. Benzer şekilde kodun herhangi bir yerinde bir nesneyi kullanıp . tuşuna basarsanız o nesneyle erişilebilecek üye elemanlar listelenir. Visual Studio.NET'in bunun gibi birçok özelliği vardır. Bunları kendiniz program yazdıkça keşfedebilirsiniz.

20.VİSUAL STUDİO.NET VE DOSYALAR:

Visual Studio projelerimizi varsayılan olarak "C:\Documents and Settings\Kullanıcı\Belgelerim\Visual Studio 2008\Projects" klasörüne kaydeder. (Derleme ve kaydetmenin nasıl yapıldığını ileride ayrıntılarıyla anlatacağım. Şimdilik programın kapatma düğmesine basıp gelen uyarı penceresine "Yes" deyin.) Bu klasörde solution ismimiz, bu klasörün içinde proje klasörümüz/klasörlerimiz, proje klasörümüzün içinde ismi sınıf ismimizle aynı olan kaynak kod dosyamız, bu klasördeki bin klasörünün içindeki Debug klasöründe de exe dosyamız bulunur. Yani son tahlilde exe dosyamızın yolu C:\Documents and Settings\Kullanıcı\Belgelerim\Visual Studio 2008\Projects\ConsoleApplication1\ConsoleApplication1\bin\Debug şeklindedir. Projemize kaynak kod dosyası eklemek için Solution Explorer pencereciğinde projeyi sağ tıklayıp Add -> New Item yolunu izleyin. Açılan pencereden yeni oluşturulacak dosyadaki bildirimleri otomatik yapabilirsiniz. Örneğin bu açılan pencereden Code File'ı seçerseniz boş bir cs dosyası projemize eklenir, eğer Class'ı seçerseniz içinde bir sınıf bildirimi olan bir cs dosyası projemize eklenir. Yine Solution Explorer penceresindeki kaynak kodları sağ tıklayıp silebilir, projeden çıkarabilir, projeyi sağ tıklayıp Add -> Existing Item yolunu izleyerek var olan bir kaynak kodu projemize ekleyebiliriz. Ayrıca projeye sağ tıklayıp Add -> New Folder yolundan kaynak kodlarımızı koordine etmek için projemize bir klasör de ekleyebiliriz.

21.HATA AYIKLAMA:

Hata ayıklama kaynak kodumuzun çeşitli yerlerine yerleştirdiğimiz breakpointlerle yapılır. Debuglı çalıştırılan bir kod breakpoint olan satırlardan birine geldiğinde programın çalışması duraklatılır (pause). Kaynak koddaki bir satıra breakpoint koymak için ilgili satır sağ tıklanıp Breakpoint -> Insert Breakpoint yolu izlenir. Bir satıra breakpoint koymanın başka bir yolu ise imleç ilgili satırdayken F9 tuşuna basmaktır. Kod debuglı çalışırken breakpointli bir satıra gelindiğinde Debug menüsünün içeriği değişir. Bu menüdeki önemli seçenekleri inceleyecek olursak; "Step Over" programın işleyişini bir sonraki satıra götürür. "Continue" ile bir sonraki breakpointe kadar programın işleyişi devam ettirilir. "Stop Debugging" ile de hata ayıklama modundan çıkılır. Windows -> Watch yolundan değişken değerlerini adım adım izleyebileceğimiz Watch pencereciği görünür hâle getirilir. Watch pencereciğine bir değişken izlemesi eklemek için pencere sağ tıklanır ve "Add Watch" seçilir. Bu adımdan sonra boş satırın "Name" sütunu çift tıklanır ve buraya değişken ya da değişkenlerden oluşan bir ifade yazılır. Artık ilgili satırın "Value" sütununda ilgili satırdaki değişken ya da ifadenin değeri adım adım gösterilecektir. Yine Debug menüsündeki "Exceptions" seçeneği ile bir istisnai durum fırlatıldığında ya da fırlatılıp yakalanamadığında debugın devam etmesini ya da durmasını sağlayabilirsiniz.  
NOT: Bir solutionın tamamını derleyip release olarak kaydetmek için "Build" menüsünden "Build Solution" seçilir.

22.SOLUTİON KAVRAMI:

Solution, projeleri koordine etmeye yarayan bir yapıdır. Bir projede yalnızca bir tane çalıştırılabilir (exe) dosya bulunur. Bir solution'da ise birden fazla çalıştırılabilir dosya bulunabilir. Bir solution'a proje eklemek için Solution Explorer pencereciğinde solution sağ tıklanıp Add -> New Project yolu kullanılır. Aktif solution'a proje eklemenin başka bir yolu ise File menüsünden "New Project"i seçmektir. Bir projeyi başlangıç projesi yapmak için ilgili proje sağ tıklanıp "Set as StartUp Project" seçilir. Başlangıç projesinin ne demek olduğunu ileride göreceğiz.

23.DERLEME:

Visual Studio.Net başlangıç projesindeki bütün kaynak kod dosyalarının birleştirilip derlenmesi için iki yol sunar. Bunlar Debug ve Release seçenekleridir. Debug seçeneği programımızın çalışmasını test etmek ve hata ayıklamak için kullanılır. Release seçeneği ise programımızın testi bittiğinde son kullanıcının istifadesine sunmak için kullanılır. Release seçeneğinde kodumuzdan bütün gereksiz kısımlar atılır ve program daha hızlı çalışır. Debug seçeneğinde ise hata ayıklama için gerekli kodlar kaynak kodumuza gizlice eklenir.

Başlangıç projesini debuglı çalıştırmak için debug menüsünden "Start Debugging" seçilir. Projeyi debugsız çalıştırmak içinse yine aynı menüden "Start Without Debugging" seçilir. Bunların kısayolları sırasıyla F5 ve Ctrl+F5'tir.  
NOT: İlk aşamada solution, araç çubuğundaki disket simgesine ya da File menüsündeki Save komutuna tıklanmadan sabit diske kaydedilmez.

25.C# – Sınıflar:

Sınıflar nesne yönelimli programlamanın en önemli ögesidir. C# %100 nesne yönelimli bir dil olduğu için tüm metot ve özellikler sınıflar içerisinde yer alır. Sınıfları veri yapısı veya veri yapısı modeli olarak tanımlayabiliriz.

Sınıflara Niçin İhtiyaç Duyarız ?

Metotları anlatırken; “Bütün işlemleri tek bir metot içerisinde yapmak yerine alt programlara yani metotlara ayırmak; kod tekrarını azaltır, okunabilirliği arttırır ve programın daha kolay geliştirilebilmesini sağlar. ” demiştik. (Bkz: [C# – Metotlar](https://www.srdrylmz.com/c-metotlar/))

Aynı şekilde; program geliştirirken bütün metotları tek bir sınıfın içinde tanımlamak yerine benzer görevleri yerine getiren metotları bir sınıf içerisinde tanımlamak kod tekrarını azaltır, okunabilirliği arttırır ve programın daha kolay geliştirilebilmesini sağlar. Örneğin veritabanı işlemlerini (ekleme, silme, güncelleme, arama…) yapan metotları diğer metotlardan ayırıp farklı bir sınıf içerisinde tanımlamak daha doğru bir yaklaşım olacaktır.

Sınıf Bildirimi ve Üye Elemanlar

Sınıfların üye elemanları özellikler ve metotlardır. Bir sınıf içerisinde çok sayıda özellik ve metot tanımlayabiliriz.

|  |  |
| --- | --- |
| 4  5  6  7  8 | class SinifAdi  {      [Erişim Belirleyicisi] [Veri Tipi] OzellikAdi;      [Erişim Belirleyicisi] [Geri Dönüş Değerinin Tipi] MetotAdi([Parametreler])      {          // Metot Gövdesi      }  } |

Sınıf ve sınıf üyelerinin bildirimi yukarıdaki şablona uygun bir şekilde yapılmalıdır.

**Erişim Belirleyicisi:** Metot ve özelliklere olan erişimin sınırlarını belirtmektedir.

**Private:** Sadece tanımlandığı sınıf içerisinden erişilebilir.

**Public:** Her yerden erişilebilir.

**Internal:** Sadece bulunduğu projede erişilebilir.

**Protected:** Sadece tanımlandığı sınıfta ya da o sınıfı miras alan sınıflardan erişilebilir.

Sınıf Türünden Nesneler Oluşturma

Şimdi, edinmiş olduğumuz bilgileri kullanarak konuya ilişkin bir örnek yapalım ve bu örnek üzerinden sınıfları incelemeye devam edelim.

|  |  |
| --- | --- |
| 2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | class Ogrenci  {      // Özelliklere diğer sınıflardan erişebiliriz. (Public)      public string AdSoyad;      public int Numara;      public int Sinif;        // Bilgi() metoduna diğer sınıflardan erişebiliriz. (Public)      public void Bilgi()      {          Console.WriteLine("Öğrencinin Adı:{0}", AdSoyad);          Console.WriteLine("Öğrencinin Numarası:{0}", Numara);          Console.WriteLine("Öğrencinin Sınıfı:{0}", Sinif);      }  }    class Program  {      static void Main(string[] args)      {          // Ogrenci sınıfından bir nesne oluşturduk.          Ogrenci ogr1 = new Ogrenci();            // Nesne içerisindeki public özelliklere değer atadık.          ogr1.AdSoyad = "Serdar Yılmaz";          ogr1.Numara = 134;          ogr1.Sinif = 4;            // Nesne içerisindeki Bilgi() metodunu çağırarak atamış olduğumuz değerleri ekrana yazdırdık.          ogr1.Bilgi();            Console.WriteLine(new string('-', 30));            // Ogrenci sınıfından bir nesne daha oluşturup farklı değerler atadık.          Ogrenci ogr2 = new Ogrenci();          ogr2.AdSoyad = "Ümit Özkan";          ogr2.Numara = 118;          ogr2.Sinif = 3;          ogr2.Bilgi();      }  } |

Ogrenci isminde bir sınıf oluşturduk ve içerisindeki üye elemanları public (Her yerden erişilebilir) olarak bildirdik.

Program sınıfı içerisindeki Main metodunda; “new Ogrenci()” ifadesi ile Ogrenci sınıfından bir nesne oluşturduk. Belleğin heap bölgesinde oluşturulan bu nesneye erişebilmek için adresini, ogr1 adını verdiğimiz referans değişkenine aktardık.

Nesneler belleğin heap bölgesinde, nesnelerin adreslerini tutan referans değişkenleri de stack bölgesinde tutulur.

ogr1 referansı ile nesnenin public üyelerine erişip değerler atadık. Daha son Bilgi() metodunu çağırarak atamış olduğumuz değerleri görüntüledik.

Bir nesne oluşturduğumuz an nesne içindeki üye elemanlara varsayılan değerler atanır. Numarik ise “0”, bool türünde ise “false”, string gibi referans tipliyse “null” değeri atanır.

Programımızın sonraki satırlarında “new Ogrenci()” ifadesi ile yeni bir nesne daha oluşturduk ve adresini ogr2 referans değişkenine aktardık. Benzer şekilde; üye elemanlarına değerler atayıp, Bilgi() metodu ile atamış olduğumuz değerleri görüntüledik.

26.EKRAN ÇIKTISI:

Öğrencinin Adı:Serdar Yılmaz

Öğrencinin Numarası:134

Öğrencinin Sınıfı:4

------------------------------

Öğrencinin Adı:Ümit Özkan

Öğrencinin Numarası:118

Öğrencinin Sınıfı:3

Press any key to continue . . .

Son olarak Sınıflar için kilit nokta sayılabilecek bir kaç hususa değineceğiz.

Örnek programımızda iki adet nesne oluşturduk ve ilkinin adresini ogr1 değişkenine, ikincisinin adresini ogr2 değişkenine atadık. Her iki nesneyi de Ogrenci sınıfından oluşturmamıza rağmen nesneler birbirinden tamamen bağımsızdır. Çünkü içeriğin başında da belirttiğimiz gibi Sınıflar sadece birer veri modelidir. Bir sınıfı model alan çok sayıda nesne oluşturabiliriz ve bu nesneleri birbirinden bağımsız olarak kullanabiliriz.

Bir sınıftan oluşturulan nesnelerin heap bölgesinde, nesnelerin adreslerini tutan referans değişkenlerinin de stack bölgesinde tutulduğunu söylemiştik. Eğer bir sınıftan herhangi bir nesne oluşturulmazsa o sınıf bellekte her hangi bir yer kaplamaz. Özetle sınıf bildirimleri için bellekte yer tahsis edilmez.

27.C# Metotlar ve Metot Örnekleri:

Bu yazımızda Metotların tanımı, metot çeşitleri, metot kullanımlarının nasıl yapıldığını ve bununla ilgili örnekler oluşturacağız. Öncelikle Metot ne demektir? Metot ne işe yarar? sorularının cevaplamaya çalışacağım.

Metotlar belirli işlemleri yerine getiren kod bloklarıdır. Aynı kodların tekrar tekrar kullanılması gereken durumlarda büyük kolaylık sağlamaktadır. Oluşturulan metot, ismiyle çağırılarak içerisinde bulunan kod bloklarının çalıştırılması sağlanır. Metotların yapısı temel olarak şu şekildedir.

Geri dönüş tipinin metot tanımlanırken mutlaka belirtilmesi gerekir. Burada metottan geriye herhangi bir değer dönmeyecekse void anahtar sözcüğü kullanılmalıdır. Eğer geriye bir değer döndürülecek ise geri dönecek değerin türü  (string, int,double,List<>,vs..) belirtilmelidir. Geriye değer döndüren metotlarda geriye döndürülecek değer return anahtar kelimesiye bildirilmelidir.

Metot ismi ise metodumuza bizim belirleyeceğimiz metodun yapacağı işi kısaca özetleyen bir isim olabilir. Örnek olarak “Temizle”, “AlanHesapla”, “Topla”  gibi isimler verilebilir.

Son olarak parametre kısmına gelelim. Bir metot parametreli veya parametresiz olabilmektedir. Eğer metot parametre almıyorsa yani dışarıdan değer verilmeyecekse “( )” parantez açılıp kapatılarak  metot oluşturulur.

Görüleceği gibi 4 farklı şekilde metot oluşturup kullanabiliriz.

1.Geriye Değer Döndürmeyen ve Parametre Almayan metotlar.

2.Geriye Değer Döndürmeyen ve Parametre Alan metotlar.

3.Geriye Değer Döndüren ve Parametre Almayan metotlar.

4.Geriye Değer Döndüren ve Parametre Alan metotlar.

Şimdi bu yazdıklarımızı nasıl uygulayacağımıza bakalım. Dikdörtgenin alanını hesaplayan bir metodu 4 farklı şekilde de tanımlayalım.

İlk olarak Geriye Değer Döndürmeyen ve Parametre Almayan Metot Kullanarak Alan Hesabı yapalım.  
Metodumuz aşağıdaki şekilde olacaktır. Görüleceği gibi void anahtar sözcüğü kullanılarak metodun bir değer döndürmediğini belirtiyoruz. Bütün işlemler metot içerisinde yapılmıştır.

28.C#' ta kullanılan operatörler ve işlevleri

Programlama dillerinde tek başlarına herhangi bir anlamı olmayan ancak programın işleyişine katkıda bulunan karakter ya da karakter topluluklarına operatör denir. Operatörlerin etki ettikleri sabit ya da değişkenlere "operand" denir.

Operatörler işlevlerine göre 6 kısımda incelenebilir. Bunlar;

* Aritmetik operatörler.
* Karşılaştırma operatörleri.
* Bitsel operatörler.
* Mantıksal operatörler.
* Atama ve işlemli atama operatörleri.
* Özel amaçlı operatörler.
* **Aritmetik operatörler**
* + , ‐ , \* , /, ++ , -- ve % aritmetik  operatörlerdir.
* /, \*, + ve – operatörleri matematiksel işlemler yapar.
* % operatörü ise bölümden sonra kalanı bulmak yani "mod" almak için kullanılır.
* ++ ve -- işlemin uygulandığı sayıyı 1 arttırır veya 1 azaltır.
* \*, /, % operatörleri aynı satırdaysa eşit önceliğe sahiptir. + ve – operatörleri de aynı satırdaysa eşit önceliklidir fakat  \*, /, % operatörlerinin + ve – operatörlerine göre önceliği vardır.
* Karşılaştırma operatörleri

C#’ ta 6 adet karşılaştırma operatörü vardır. Bunlar; <(küçüktür), >(büyüktür), <=(küçük veya eşittir), >=(büyük veya eşittir), ==(eşittir), !=(eşit değildir).

Karşılaştırma operatörleri aritmetik operatörlerden düşük önceliklidir.

<, >, <=, >= operatörleri  == ve != operatörlerine göre önceliklidir fakat aynı satırda olduklarında eşit önceliğe sahiptir.

Bitsel operatörler

Bitsel operatörler &(bitsel ve), ~(bitsel değil), |(bitsel veya), ^(bitsel özel veya) operatörleridir. Sayıların kendileri yerine bitlerini kullanan operatörlerdir.

* &(ve) işlemi karşılıklı basamaklardan her ikisi de 1 ise ilgili basamağında 1 tutar, diğer durumlarda 0 tutar.
* ~(değil) operatörü operandının her basamağını tersleştirip tutar.
* |(veya) işlemi karşılıklı basamaklarından herhangi birisi 1 ise ilgili basamağında 1 tutar, diğer durumlarda 0 tutar.
* ^(özel veya) işlemi karşılıklı basamakları farklıysa ilgili basamağında 1 tutar, diğer durumlarda 0 tutar.
* Bitsel operatörler aritmetik operatörler ve karşılaştırma operatörlerinden düşük önceliklidir fakat bu operatörler aynı satırda olduklarında aralarındaki öncelik sıralaması &(bitsel ve), ^ (bitsel özel veya), |(bitsel veya) şeklindedir.

Mantıksal operatörler

C#’ ta 3 adet mantıksal operatör bulunmaktadır. Bunlar; &&(ve), !(değil), ||(veya) operatörleridir.

* **||**(veya) operatörü iki değerden birisi doğru ise doğru, ikisi de yanlış ise yanlış değerini döndürür ayrıca önceliği en az olan mantıksal operatördür.
* **&&**(ve) operatörü iki değerin ikisi de doğru ise doğru, en az birisi yanlış ise yanlış değerini döndürür.
* **&&**(ve), **||**(veya) operatörleri aritmetik, karşılaştırma ve bitsel operatörlere göre düşük önceliklidir fakat kendi aralarındaki öncelik sıralaması &&(ve), ||(veya) şeklindedir.
* **!**(değil) operatörü aritmetik operatörlerden (++)arttırma ve (--)eksiltme operatörleri ile eşit önceliklidir.

Atama ve işlemli atama operatörleri

=(atama) operatörü bir değişkene herhangi bir değer atamak için kullanılır.

\*=, /=, +=, -=, &=, ^=, |= gibi operatörlere işlemli atama operatörü denir. Kullanımı şu şekildedir;

a=a+b yerine a+=b

a=a/b yerine a/=b

a=a^b yerine a^=b

Atama ve işlemli atama operatörleri aynı satırda yer alıyorsa öncelik sıralamaları soldan sağa eşittir fakat genel olarak bakıldığında bu operatörler öncelik sıralaması bakımından en son sırada yer alır.

**Özel amaçlı operatörler**

* **?: operatörü**: C#’ ta üç operand alan tek operatördür. Kullanımı şu şekildedir;  
  koşul ? doğru\_değer : yanlış değer
* **() tür dönüştürme operatörü**: Tür değiştirmek için kullanılan bir operatördür.  
  (değiştirilecek tür) değişken\_yada\_sabit
* **[] operatörü**: Dizilerde elemanın dizinini belirtmek için kullanılır.
* **+ ve – operatörü**: Bir değişkenin pozitif veya negatif olmasını sağlar. Ayrıca + operatörü iki string(katar)’i birleştirmekte de kullanılır. Kullanımı şu şekildedir;  
    
  string a = “Test“;  
  string b = “Deneme”;  
  string c = a + b;

olarak ifade edildiğinde c değişkenine "TestDeneme" atanmış olur.

* **typeof operatörü**: Herhangi bir değişken türünün CTS (Common Type System - Ortak Tip Sistemi) karşılığını type(tip) olarak tutar.
* **sizeof operatörü**: Temel türlerin ve yapıların bellekte ne kadar alan kapladıklarını verir. "sizeof" operatörü sınıflar için kullanılamaz ancak tanımlanacak yapılar için kullanılabilir.
* **new operatörü**: Yeni bir nesne oluşturmak için kullanılan operatörlerdir.
* Özel amaçlı operatörler aynı satırda yer alıyorsa öncelik sıralamaları eşittir fakat bu operatörler diğer operatörler ile karşılaştırıldığında öncelik bakımından ilk sırada yer alırlar.

29. SQL NEDİR ?

SQL, bilgisayar başında sıklıkla olanlar için vazgeçilmez bir sistemdir. Bilgisayar başında bulunup çokça veri ile ilgilenmeniz gerekmektedir. Bu yazı da bir veri içerisinden sizlere sunulmuştur. Bu yazının içeriğindeki veriler, veri tabanı içerisinden sizlere sunulmuştur. SQL, belli kriterler içerisinde verileri size sunma ve sorgulama demektir. Bir nevi sorgulama dili denilmektedir. Sorgulama sırasında veriler üzerinde işlemler yapmamızı sağlamaktadır. Açılımı “Structured Query Language” olan ve Türkçesi “Yapılandırılmış Sorgu Dili” olan SQL, İngilizce isminin baş harflerinden oluşmaktadır.

30. SQL NASIL ÇALIŞIR?

Veriler üzerinden hızlı bir işlem yapılmasını sağlayan SQL sayesinde büyük veriler üzerinde rahatlıkla sorgulama e sunum işlemleri yapılabilmektedir.

**Büyük verilere sahip uygulamalar ve bilgisayar programları SQL sistemini kullanarak rahatlıkla veri tabanı yönetimi yapabilmektedir.**

Veri tabanı yönetiminde dikkat edilecek kısım ise dağınık verilerin düzenlenmesidir. Büyük bir veri üzerinde çalışıyorlarsa kategorize etmek ve dağınıklıkları düzenlenmek, sorgulamalar ve veri arama sırasında rahatlıkla işinize yaramaktadır. Hafızada dağınık olarak depolanan bilgiler, kategorize edilip düzenlendiği zaman, depolama alanı açısından ve düzen açısında rahatlık sağlanmış olur. Depolama alanı işgal edilmez ve alandan tasarruf edilir. Örneğin öğrenci bilgi sistemleri, pek çok öğrencinin notlarını, derslerini , kişisel bilgilerini ve eski pek çok dersi ile birlikte geçmiş senelerinin notlarını görülebilmektedir. Bu şekilde bir sorgulama yapıldığında ve öğrenci bilgi sistemine girmek isteyen bir öğrenci, SQL sistemi tarafından verilerin sorgulanmasını ekranında görmüş olur.

31. SQL SİSTEMLERİ NASILDIR?

Büyük bir veriye sahip şirketler, bu verilerini veri merkezlere denen devasa bölümlerde tutmakta ve buralardan yönetmektedir. Pek çok kamu kuruluşu da bu tarz veri merkezleri kurarak devlet denetimi altında işlemektedir. Sonra yılların popüler uygulaması olan e-devlet hizmeti de bir SQL veri tabanıdır. Bu veri tabanı sayesinde sisteme TC kimlik numaranız ile kayıt olduğunuzla sizinle ilgili devlet hizmetlerinin kayıtları veri tabanından çekilerek size sunulur. Veri tabanı birbiriyle ilişkisi olan pek çok farklı dosyayı birbiriyle tıpkı bir kütüphane mantığı gibi senkronize etmektedir.

32. SQL EĞİTİMLERİ NEDİR?

SQL kursu, ülkemizde özellikle son yıllarda ciddi oranda artış göstermiştir. Veri tabanı yazılımlarının da yaygınlaşması nedeniyle bazı eğitimciler ve kurslar SQL eğitimi açarak hizmet vermeye başlamıştır. Bu kurslarının yaygınlaşmasının temel nedeni artık IT sektöründe ve bilişim sektöründe doğan ihtiyaçtır. Bu ihtiyaçların temel sebebi kalifiye elemanın az yetişmesi ve üniversite bulunan bölümlerin yetersizliğidir. Artık büyük şirketler ve kamu kuruluşları kendi verilerini düzenlemek ve veri tabanı oluşturmak için veri tabanı yani SQL uzmanları işe almaktadır. Bu sektörde çalışmanın da ücretleri dolgun olduğundan ve rahat çalışma şartları nedeniyle tercih edilmektedir. Verilen eğitimlerin en iyi yanı da oldukça kapsamlı ve geniş olmasıdır. Bu kurslarda öğretilen bilgilerin büyük bir kısmı genel olarak SQL in temel mantığından başlayarak daha zor kısımlarına girmesiyle oluşmaktadır.

33. NEDEN SQL KULLANILMALIDIR?

SQL kullanımını bu denli popüler kılan oldukça ikna edici gerekçeler söz konusu. Örneğin SQLQ kullanıcıların veri erişimine izin vermektedir. Verilerin açıklamalarına olanak sağlar. Yine aynı şekilde kullanıcılara verileri değiştirme şansı verir. Veri tabloları oluşturur ve öte yandan modüllerin farklı dillere gömülmesine olanak tanır. Hem tescilli olması hem de açık kaynak olması onu özel kılan bir başka öğedir. Lisanslar her zaman ücretli değildir, dolayısıyla ucuz maliyetler sunar. Kurtarma, güvenlik ve performans gibi kriterlere ön plana çıkmasıyla da dikkat çeken SQL in yazılımları ile ilgili ayrıntılı bilgi için yazının okuyabilirsiniz.

34. SQL YAZILIMLARI

Yukarıda popüler bazı SQL yazılımlarından söz etmiştik. Şimdi bu yazılımlara daha yakından bakalım

35. MYSQL:

MySQL, Oracle isimli firma tarafından ortaya çıkarılmıştır ve bugün en popüler SQL yazılımlarından biri olarak ön plana çıkmaktadır. Hem ücretsiz ve hem de açık kaynak kodları kullanır. Çifte lisanslı bir yazılım olan MYSQL, GPL Lisansına sahiptir.

Veri tabanlarına ulaşmak adına en popüler ve yayın olan dili tercih ediyor. Genelde Linux çatısı altında daha verimli bir performans sunmaktadır. Oldukça esnek bir yapıya sahip olması onun en cazip özelliklerinden biri olarak ön plana çıkmıştır. Son sürümüyle beraber alt sorgu özelliği de dahil edilmiştir. Bugün veri tabanının kullanıldığı bütün online mecralarda rahatlıkla kullanılabilir. Tablo, görüntüleme ve yordam gibi veri tabanı nesnelerini desteklemektedir.

36. MSSQL

MSSQL Microsoft çatısı altında üretilmiştir. Lisans edinmek ücretlidir. Fakat hosting firmaları söz konusu maliyetleri üstlenmektedir. Çok büyük verilen söz konusu olduğunda bu size ekstra maliyet yansıyabilir. NET ya da ASP dilleriyle ortaya çıkartılmış internet sitelerinde sıklıkla tercih edilir. Genelde kullanım kolaylığıyla ön plana çıkmaktadır. İndirmek adına Microsoft’un kendi resmi sayfası üzerindeki yönergeleri takip edebilirsiniz

37. MİCROSOFT SQL Server:

Yine aynı şekilde Microsoft tarafından geliştirilen bir nevi sunucu yazılımı olarak bilinir. Şu anda piyasada pek çok versiyonuna ulaşmak mümkündür. Son versiyonu 2008 yılının Ağustos ayında kullanıcıların beğenisine sunuldu. Bu versiyonda veri yönetimini kendi kendine ayarlaması ve kapalı kalma konusunda sorun yaşamaması, ön plana çıkan bazı yeniliklerdir. Ayrıca bu versiyonla beraber “Master Data Services” adı verilen bir veri yönetim sistemi dahil edildi.

38. Firebird:

Borland tarafından Interbase 6.0 isimli veri tabanının artık açık kaynak kodu biçimine sokulmasıyla ortaya çıkmış oldu. Alert özelliğiyle beraber tablolarda herhangi bir farklılık söz konusu olduğu zaman, bu farklılıkları bizzat kendi yazılımınız üzerinden izleyebilirsiniz. Şu ana kadar çıkartılan tüm versiyonları ücretsiz olarak sunuldu. Oldukça büyük veri tabanları için bile çok defa tercih edildi. Şu anda sisteme Türkçe dil desteği de dahil edilmiştir.

39. Access:

Aynı şekilde Microsoft tarafından geliştirilmiştir. Bu genelde mikro uygulamalarda tercih edilmektedir. Büyük veri tabanlarında sizi yarı yolda bırakabilir ve muhtemelen pek bir verim alma şansınız olmaz. En ön plana çıkan özelliği ise mobil biçimde, yani taşınabilir nitelikte olmasıdır. Yakın zaman öncesinde dek yalnızca kendi dosyaları için kullanılıyordu ancak şu anda SQL uyumlu bütün veri tabanlarına teknik olarak erişmek mümkün hale geldi. Oluşan raporları, sisteminde Access kurulu olmayan kişilere hiçbir sorun yaşamadan ulaştırmanız mümkün.

SQL Genel Özellikleri:

• SQL bir veri tabanına erişmenizi sağlar.

• SQL bir ANSI standart bilgisayar lisanıdır.

• SQL bir veri tabanı ile ilgili sorguları çalıştırmanızı sağlar.

• SQL bir veri tabanından veri almanızı sağlar.

• SQL bir veri tabanına yeni kayıt eklemenizi sağlar.

• SQL bir veri tabanından kayıt silmenizi sağlar.

• SQL bir veri tabanında kayıtları güncellemenizi sağlar.

• SQL lisanını öğrenmek kolaydır.

Şimdi sözlüğümüzü MSSQL üzerinden yapacağız. Sözlükteki kelimelerimizi MSSQL üzerine yapacağız böylece veri tabanımızı kontrol edebileceğiz.

40. MS SQL NEDİR? ÖZELLİKLERİ

MS SQL (Microsoft – Structured Query Language), Microsoft tarafından geliştirilen bir veri platformudur. Bu platform sayesinde, veri tabanı oluşturmamıza ve geliştirmemize olanak sağlar.

MS SQL veri tabanı altyapısı ile çevrimiçi veya çevrimdışı işlemlerimiz için ilişkisel veri tabanları oluşturulabilir. Oluşturduğumuz bu veri tabanları üzerinde tablolara oluşturarak datalarımızı kolay bir şekilde yöneterek(Ekleme, Güncelleme, Silme, Listeleme gibi.) güvenli bir şekilde saklayabiliriz.

41. SQL SERVER MANAGEMENT STUDIO NEDİR?

MS SQL Server kullanıcı odaklı ürün destekleri bulunmaktadır. Bunlardan SQL Server Express sürümü ise masaüstünde, web üzerinde ve küçük sunucularda çalışan uygulamaları geliştirmek ve üretmek için ideal olan ücretsiz bir sürümdür.

Sözlüğün yapımına sıralı bir şekilde başlayabiliriz.

Başta MSSQL, SERVER 2017 ve VISUAL STUDIO 2019’u kuracağız.

# 4.1. SQL Server 2017 Kurulumu

Öncelikle SQL işletim sistemi ile aynı ortamda kurulmaz ise daha iyi olacaktır. Ayrıca DB , LOG, TEMP DB, Backup için ayrı disk birimleri kullanmamızda olumlu yarar olacaktır. Tabi buna göre depolama seçimi , disk seçimi ve baskın seçimi yapmamız gerekmektedir. Ben kurulum için ayrı ayrı disklerimi hazır hale getirdim.



Şekil *2 64 K DİSK AYRIMA*

SQL Server 2017 indirme dosyasını sağ tık yapıp yönetici olarak çalıştıralım

Gelen güvenlik uyarısını evet olarak geçelim.

•Şekil 2.1 de gelen ekranımızda iyi bir kurulum optimize edebilmek için bu ekranımızdaki linklerden yararlanarak gerekli noktalarda bilgiler alabiliriz. Bu kurulum adımına geçeceğimiz için “Installation” seçimine tıklayalım. Biz yeni temiz bir kurulum yapacağımız için “Installation” tabında yer alan “New SQL Server stand-alone installation or add features to an existing installation” seçeneğine tıklayalım.

* Kurulum için gerekli olan ürün anahtarımızı girelim. Biz test işlemleri yapacağımız ve lisans anahtarımız olmadığı için “Evaluation” seçimini yaparak “ Next” ile ilerleyelim.
* Lisans anlaşmasını kabul edelim ve “Next” ile ilerleyelim.
* Use Microsoft Update to check for updates(recommended) seçeneğini seçerek kurulum öncesi gerekli güncellemeleri çekelim.
* Kurulum öncesi sistem gereksinimlerimiz kontrol ediliyor. Gerekli bileşenler var olduğundan sadece Windows Firewall uyarı veriyor. Bu hata değil uyarı olduğundan “ Next “ ile ilerliyoruz.
* Bu ekranımızda istersek tüm SQL özelliklerini kurabilir, istersek sadece gerekli olan bileşenleri kurabiliriz. Bize gerekli olan kurulum için sadece “Database Engine Services” seçeneğini işaretleyerek buradaki seçimi tamamlıyor. SQL kurulum yollarını olduğunu gibi bırakıyoruz. Tabi bunu özelleştirme imkanımız var. Buradaki seçim performans optimizasyonuna göre değişir.
* Bu ekranımızda kurulacak olan SQL için bir “instance” ismi belirleyebilir veya varsayılan instance ismi ile kurulum adımını geçebiliriz. Biz burada ayar değiştirmeden varsayılan olarak “MSSQLSERVER” instance ismini değiştirmeden “Next” ile sonraki adıma gidiyoruz.
* Karşımıza “Server Configuration” ekranı geldi. “Service Accounts” tabında SQL server 2017 servislerini çalıştıracak kullanıcı hesapları belirlenir. Bu servis seçimlerini aşağıdaki gibi bırakabilirsiniz. “Collation” seçimini yapmak için bu tabımıza tıklayalım.
* •Collation tabından SQL Server dil ve karakter set ayarları yapılır. “Collation” ayarında “SSQL\_Latin1\_General\_CP1\_CI\_AS”seçimini yaptık. “Next” butonuna tıklayalım.
* Database Engine Configuration kısmında kimlik doğrulama metodunu seçmemiz ve bunun için kullanıcı belirlememiz istenmektedir. Sql Windows kimlik doğrulaması ve Sql kimlik doğrulaması ile kurulabilmektedir. Biz ikisinin aynı anda aktif olma durumunu isteyebiliriz. Bu nedenle “Mixed Mode” çimini işaretliyoruz. Add Current User butonuna tıklayarak mevcut kullanıcımızı Windows Authentication için ekleme işlemini yapalım.
* Data Direcktories” tabına tıklayalım. Bu tabta LOG, DB ve bileşenlerin kurulacağı yerler belirlenebilmektedir. Biz bu ekranda bu alanlar için gerekli disk yapılandırmamızı kuruluma başlamadan yapılandırmıştık. Burada gerekli yollarımızı gösterelim ve bu ekrandaki diğer yapılandırma için “TEMPDB” tabına tıklayalım.
* TEMPDB için yine hazırlamış olduğumuz disk yollarını gösterelim. Bu ekranımızda eski ekranlara göre farklı bir yapı gelmiş durumda. Performans için gerekli sayıda gerekli boyutta TEMPDB oluşturma imkanına sahibiz. Ben ayarlarımı aşağıdaki gibi yapıp “Next” ile bir sonraki ekrana ilerliyorum.
* Bu kısma kadar olan kısımda yapılan seçimlerin bir özeti karşımıza geldi. Bu ekranda yer alan “Configuration file path” yolundaki dosyasına erişerek bu kurumlar sırasında hangi komutları işletileceğini görebiliriz. Bunlar otomasyonel kurumlarda işimize yarayabilmektedir. “Install” butonuna tıklayarak kuruluma başlayalım.

42.DÜŞÜK SEVİYELİ PROGRAMLAMA DİLİ:

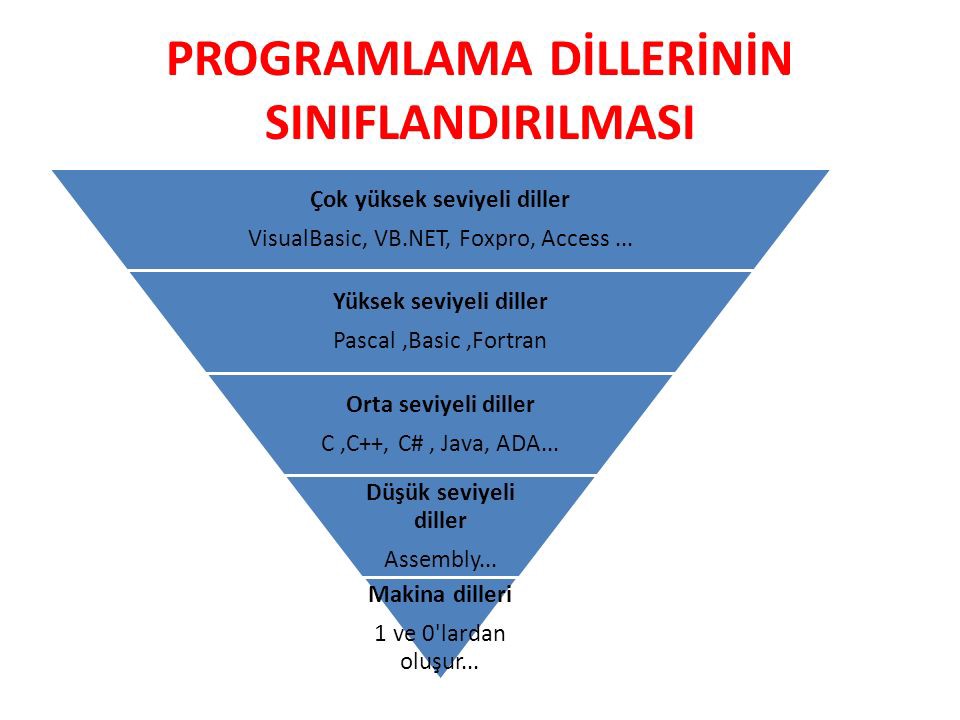
Bilgisayarın ilk dönemlerinde kullanımı çok zor olan Makine ve Assembly dilleri kullanılırdı. Makine dili , geliştirilen ilk programlama dilidir. Makine dilinde yazılan tüm komutlar 0 ve 1'lerden oluşur. Bütün talimatlar en detaylı bir şekilde tanımlanır ve iki tabanlı sayı sistemi kullanılarak kodlama yapılır. Bu dillerin kullanımı çok zordur. Bu diller ve makine diline yakın diller düşük seviyeli programlama dili diye nitelendirilir.

43.YÜKSEK SEVİYELİ PROGRAMLAMA DİLİ:

Bu dillerin öğrenilmesi , yazılması daha kolaydır. Bu diller aynı zamanda daha iyi sonuçlar sağlar. Bu tür yüksek seviyeli dillerde yazılan programlar makine diline çevrilmek zorundadır.

44.ÇOK YÜKSEK SEVİYELİ PROGRAMLAMA DİLLERİ:

Bu dillerdeki temel özellik kullanıcıların bilgisayara bir şeyin nasıl yapılacağını değil, ne yapılacağını ifade edebilmelerine imkan vermesidir.



Şekil *3 Visual Studio indirme sitesi*

C programlama dili düşük seviyeli dillere oranla daha anlaşılır , çok yüksek seviyeli dillere oranla daha fazla esnek bir dil olması sebebiyle arada bir noktada diyebiliriz. C dili makine diline yakın olması sebebiyle hem donanıma hitap eder , hem de uygulama geliştirmede kullanılır. C ile programlamada bilgisayara daha fazla hakimiyet vardır , daha az hazır fonksiyon ve kütüphane vardır. Bundan dolayı donanım hesaba katılarak programlama yapılması gerekir.

Düşük seviyeli veya yüksek seviyeli dil demek o dilin eksik yönleri olduğunu çağrıştırmasın. Düşük seviyeli ve yüksek seviyeli dil denilmesindeki asıl sebep dilin , makine diline yakın olması veya uzak olmasıdır.

45.C++ DİLİ NEDİR?

[C](https://wmaraci.com/nedir/cplusplus)**++**, 1979 yılında Bjarne Stroustrup tarafından Bell Labs’[da](https://wmaraci.com/nedir/da) geliştirilen nesne yönelimli ve yüksek seviyeli, genel maksatlı programlama dilidir. C++’ın ilk ismi “C with classes” dir ve C programlama dilinin bir eklentisi olarak işlev görmektedir.

C++’ın yaratıcısı Bjarne Stroustrup bu programlama dilini öğrenciyken geliştirmiştir. Kullandığı programlama dilllerini yeteri kadar hem işlevli hem de yüksek performanslı görmeyen Stroustrup, kendi programlama dilini oluşturarak bilgisayar tarihinin en önemli [yazılım](https://wmaraci.com/nedir/yazilim) dillerinden birinin ortaya çıkmasına yardımcı olmuştur.

C++ programlama dilinde C tarzında veya nesne yönelimli tarzda kesin senaryolarla kodlamalar yapılabilmektedir. Bu açıdan C++ en önemli ve işlevsel hibrit programlama dillerinden biri olma niteliğine de sahiptir. C++ programlama dilinin Türkçe’de telaffuzu “Si-Pılas-Pılas” şeklindedir.

46.C++ NE İŞE YARAR:

C++ geliştirilmesinden sonra tüm dünyada en yaygın kullanı[lan](https://wmaraci.com/nedir/lan) programlama dillerinden biri olmuştur. Özellikle söz konusu performans olduğunda C++ daima ilk seçim olmaktadır zira diğer programlama dilleriyle kıyaslandığında C++ çok daha fazla güncellenen bir yazılım dilidir. Son olarak C++ 11 sürümü 2011 yılında yayınlanmıştır.

C++ yazılım dili kullanılarak geliştiriciler tarafından sistem yazılımları, özel yazılımlar, uygulamalar, sürücü yazılımları, kullanıcı taraflı yazılımlar ve gömülü [firmware](https://wmaraci.com/nedir/firmware) yazılımlar üretilmektedir.  Bugün bilgisayarınızda kullanmakta olduğunuz birçok donanımın dahi sürücüsü C++ [programlama dili](https://wmaraci.com/nedir/programlama-dili) sayesinde hazırlanmış ve son kullanıcıların kullanımına tahsis edilmiştir.

47.C++ ÖĞRENME:

Basit programlama dillerinin aksine C++ gibi gelişmiş bir programlama dilini sıfırdan öğrenmek elbette daha zorlayıcı bir süreç olacaktır. İnternetteki çeşitli kaynaklardan faydalanarak öğrenmeye başlayabileceğiniz C++’ı yabancı kaynaklardan görsel eğitim setleriyle de ücretsiz olarak öğrenebilirsiniz. C++ ile ilgili WM Aracı kullanıcılarının yaptığı paylaşımlara erişmek için forumumuzdaki [C++](https://wmaraci.com/forum/c-c-c.html) kategorisini ziyaret edebilirsiniz.

C++’da ilk mini yazılımınızı yapmak ve ekrana “Hello World” çıktısını almak içinse aşağıdaki [kod](https://wmaraci.com/nedir/kod) betiğ[ini](https://wmaraci.com/nedir/ini) kullanabilirsiniz;

#include <iostream>

int main()

{

  std::cout << "Hello, world!n";

}

48.C PROGRAMLAMA DİLİ NEDİR?

C programlama dili 1972 yılında Bell Laboratuvarı’nda Dennis Ritchie tarafından geliştirilmeye başlandı. Bu dilin temelini “B” (BCPL: Basic Combined Programming Language) olarak adlandırılan dil oluşturur. C programlama dili yapısal bir dildir. Geliştirilme amaçlarından birisi UNIX işletim sisteminin uygulanmasıdır. C programlama dilinde belli bir standart oluşturulması için 1983 yılında Amerika’da ANSI (Amerikan Ulusal Enstitüsü) komitesi kuruldu. C dilinin standartları 1989 yılında tamamlandı ve bu standartlara “ANSI C” ismi verildi. C programlama dili ile ilgili ilk basılan yayın “The C Programming Language” (Dennis Ritchie ve Brian Kernighan)’dir. Bu yayın K&R C ismiyle tanımlanmıştır.

C programlama dili orta seviye bir dildir. Yani düşük seviye diller makine diline yakın olan çok kod yazma gerektiren diller, yüksek seviye diller ise az kod yazma işlemi gerektiren ama arka planda çok kod çalıştıran dillerdir. C programlama dili bu iki seviye arasındadır.

Windows, Linux ve Unix için C programlama dili kullanılarak program yazılabilir. Ayrıca C programlama dili Assembly dili ile geliştirilmiştir.

**C Programlama Dili Standartları**

C89/C90 standardı -> ANSI tarafından geliştirilmiştir. 1989 yılı C89 ve C90 standarlarını referans almıştır.

C99 standardı -> 1999 yılında düzenlenerek tekrar yayınlanmıştır. Bu düzenleme ile birçok veri tipi eklenmiş ve yeni özellik eklenmiştir.

**C Programlama Dilinin Özellikleri**

* C programlama dili esnek bir yapıya sahiptir.
* C programlama dili birçok dile göre oldukça güvenilir
* Birçok platformda çalışabildiği için taşınabilir bir dildir.
* İnteraktif bir dildir.
* Verimli ve modüler

**C Programlama Dilinin Kullanıldığı Yerler**

* Kelime işlemciler,
* Veri tabanı sistemleri,
* Grafik paketleri,
* İşletim sistemleri
* Ağ sürücüleri
* Tablolama yazılımları

49.ALGORİTMA NEDİR?

Bir sorunu çözmek veya belirlenmiş bir amaca ulaşmak için tasarlanan yola, takip edilen işlem basamaklarına algoritma denir. Algoritmalar açıkça belirtilmiş bir başlangıcı ve  sonu olan işlemler kümesidir. Amaca ulaşmak için işlenecek çözüm yolları ve sıralamaları belirlenir ve algoritma bu sırayı takip ederek en mantıklı çözüme ulaşır. İlk algoritma, El-Harezmi’nin ‘Hisab-el Cebir ve El Mukabala’ kitabında sunulmuştur ve algoritma kelimesi de El-Harezmi’nin isminden gelmiştir. Algoritma terimi genellikle matematikte ve bilgisayar bilimlerinde karşımıza çıkmaktadır.

50.ALGORİTMA NE İÇİN KULLANILIR?

Algoritma matematikte bir sonuca ulaşmak için takip edilmesi gereken işlem basamaklarını belirlemede kullanılır. Örneğin, 3 ve 5 sayılarının ortalamasını bulmak istediğinizde önce (3+5) işlemini yaptıktan sonra sırada “çıkan sonuç/2” işlemini yapmanız gerekmektedir.

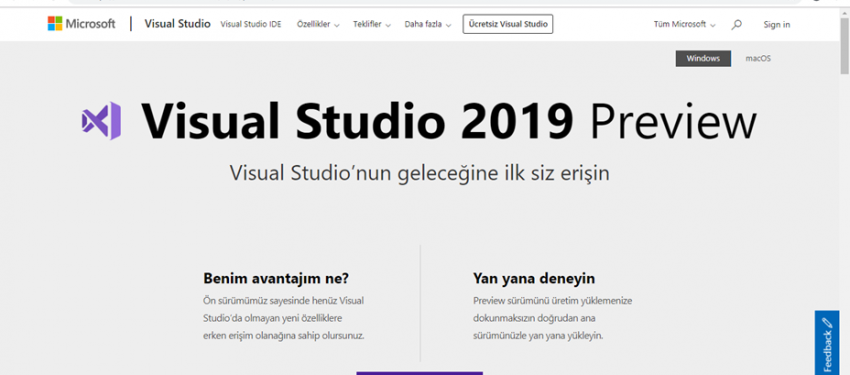
Bilgisayar biliminde algoritmalar yazılımdaki her yerdeler. Örneklendirecek olursak, bilgisayarda offline olarak yapılan tüm işlemler, oynanan oyunlar, çalıştırılan programlar, web siteleri, mobil uygulamalar, robotlar… Kısacası sanal ortamdaki her işlem algoritmalar ile gerçekleşiyor. Gelin beraber algoritma ile programlama arasındaki bağlantıyı inceleyelim.

51.ALGORİTMA VE PROGRAMLAMA BAĞLANTISI:

Tüm programlama dillerinin temelinde algoritma vardır. Algoritmalar, programlama dillerinin vasıtasıyla uygulanabilirler. Programda kullanılan dil ve kullanım alanı ne olursa olsun algoritması olmayan bir program yoktur. Programda bir algoritmanın işlemesi için dışarıdan gelen tüm girdiler “değişken” olarak tanımlanır. Algoritmadaki döngüler ve işlemler, bu değişkenler üzerinden gerçekleşir.

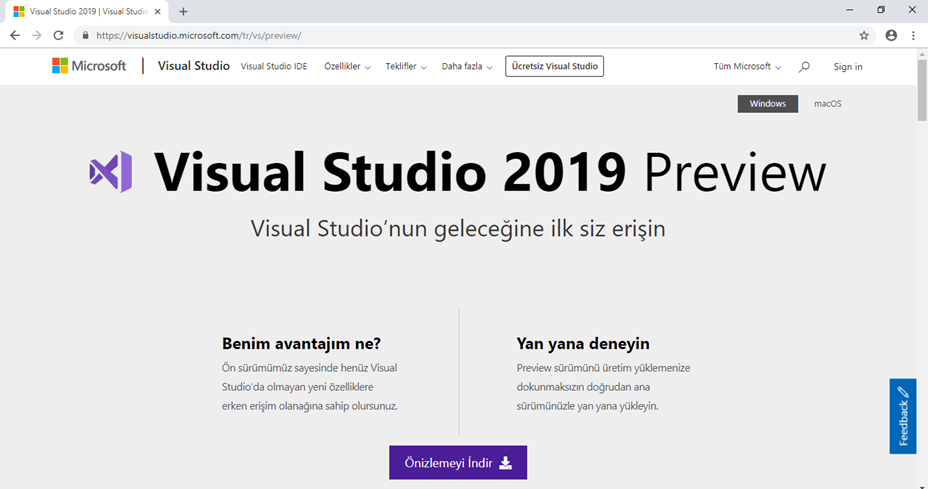
Algoritmadaki tüm ihtimaller belirtilmiş ve net olmalıdır. Hiçbir ihtimal şansa bırakılmamalıdır.

52. [MİCROSOFT VİSUAL STUDİO](https://www.mshowto.org/tag/microsoft-visual-studio) 2019 KURULUMUNU GERÇEKLEŞTİRECEĞİZ:



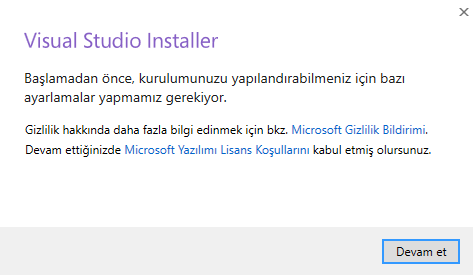
Şekil *4 Visual Studio indirme sitesi*

https://visualstudio.microsoft.com/tr/vs/preview/ adresine giriş yapalım;



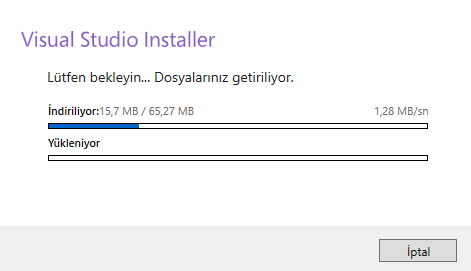
Şekil *4.1 Visual Studio indirme sitesi*

[**Visual Studio**](https://www.mshowto.org/tag/visual-studio-2) 2019 Preview sürümünü indirmek için Önizlemeyi İndir seçeneğine tıklayalım. İndireceğimiz dosyayı çalıştıralım ve yükleme işlemine başlayalım.



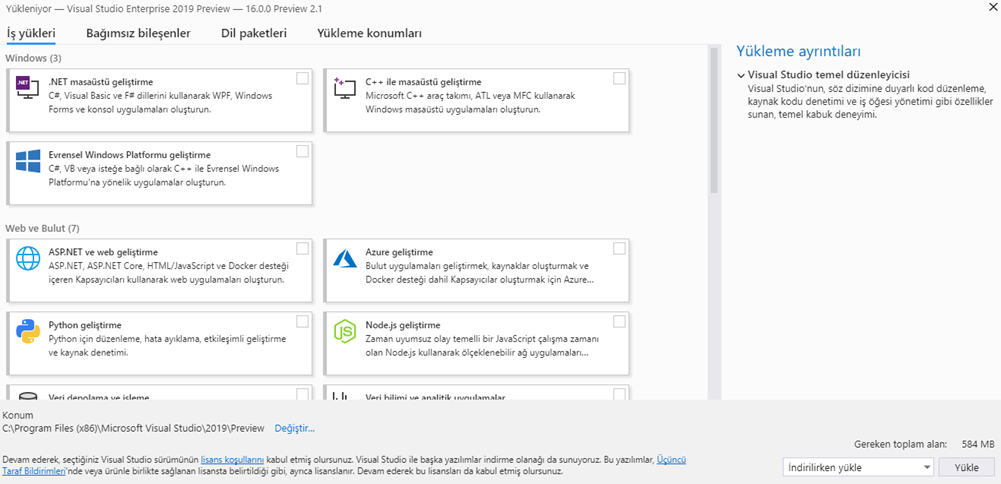
Şekil *4.2 Visual Studio kurulum ilk aşama*

Visual Studio Installer ortamında “Devam Et” seçeneği ile ilerleyelim.



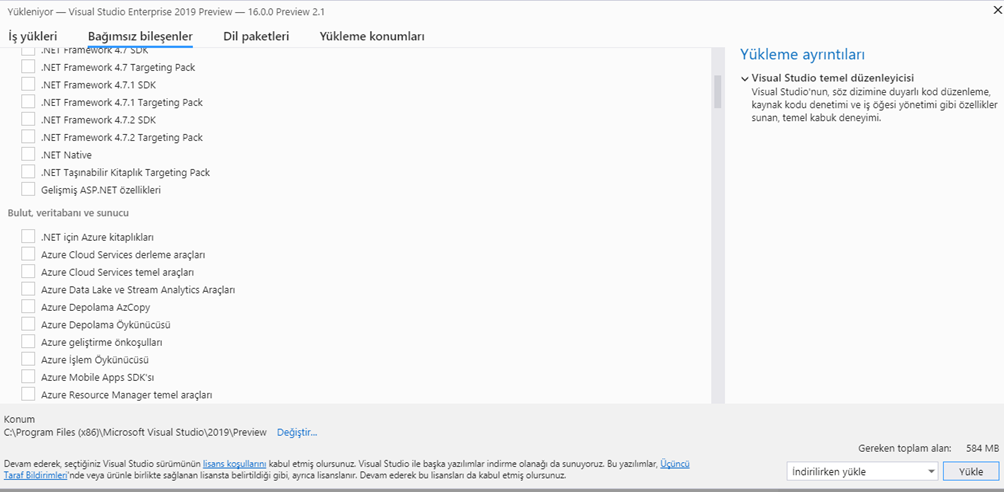
Şekil *4.3 Visual Studio indirme bölümü*

[Visual Studio 2019](https://www.mshowto.org/tag/visual-studio-2019) sürümüne ait dosyaların indirilme işlemini başlatmış olduk.



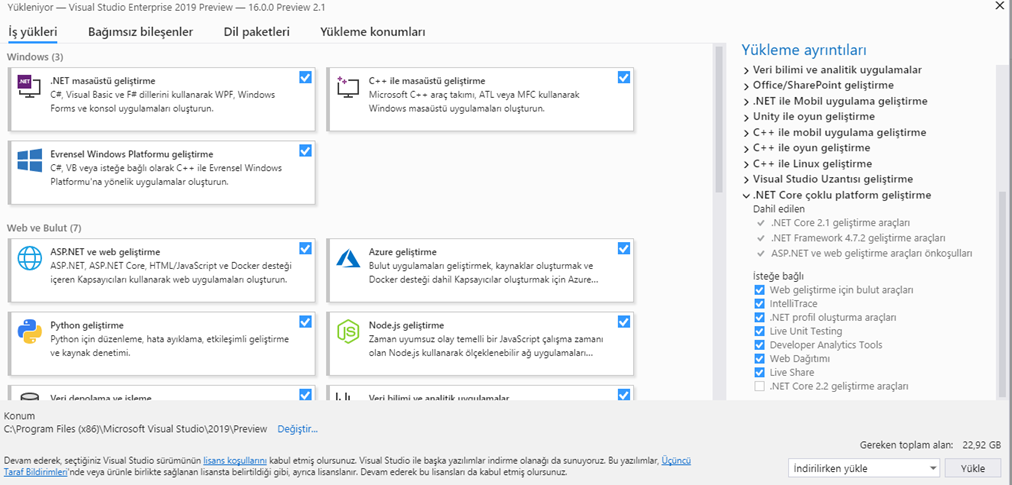
Şekil *4.4 Visual Studio iş yükleri sırası*

“İş Yükleri” menüsü altında geliştirme yapmak istediğimiz ortamları seçebiliriz.



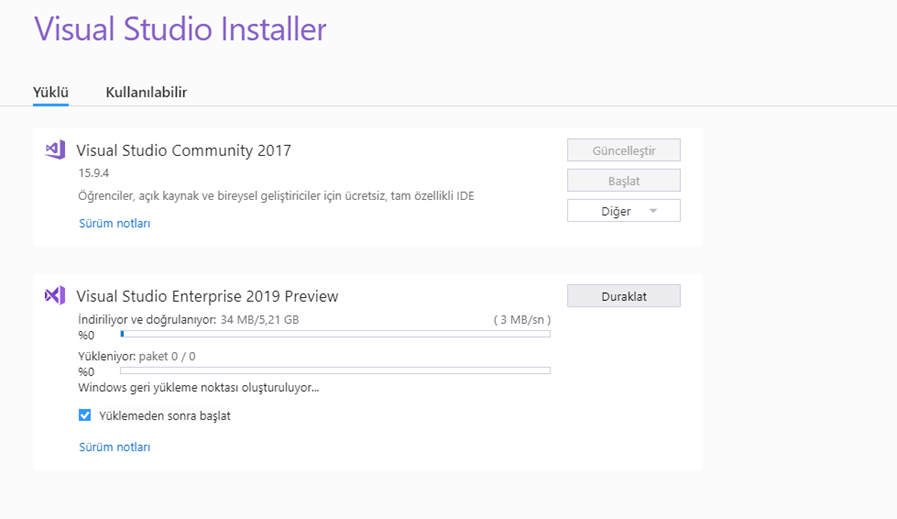
Şekil *4.5 Visual Studio bağımsız bileşenler*

“Bağımsız Bileşenler” menüsünde geliştirme yaparken kullanacağımız Framework veya Bulut paketleri ve geliştirme SDK paketleri bulunmaktadır.



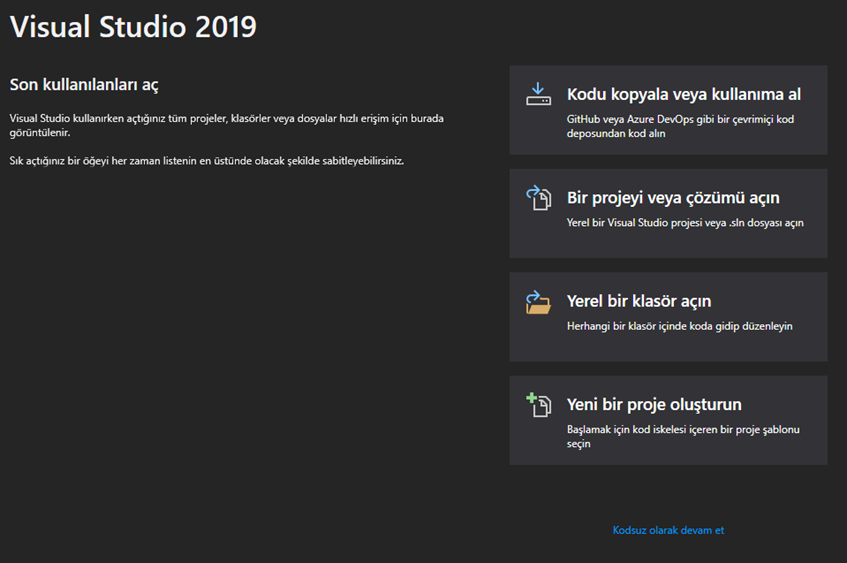
Şekil *4.6 Visual Studio bağımsız bileşenler*

“İş Yükleri” menüsünde bulunan tüm seçenekleri seçelim bu seçenekleri seçtiğimiz zaman “Bağımsız Bileşenler” menüsü altında bulunan seçeneklerden gerekli olanlar da otomatik olarak seçilmiş olacaktır. Yükle seçeneğini seçelim de işleme devam edelim.



Şekil *4.7 Visual Studio yükleme işlemi*

Yükleme işlemini başlatmış olduk. Bilgisayarımızda başka bir Visual Studio sürümü varsa yükleme yaparken görüntülenecektir.



Şekil *4.8 Visual Studio giriş sayfası*

Visual Studio yüklendikten sonra otomatik olarak açılacaktır. Visual Studio 2019 geliştirme ortamını yüklemiş olduk.

53.DATA GRİD VİEW’DE SQL VERİLERİ GÖSTERME:

Data Grid View de sql verilerini göstermek için kullandığım kodlar:

1. Öncelikle olaylar sekmesinden cell enter bölümüne 2 kez tıklıyoruz.
2. textbox.Text = dataGridView1.CurrentRow.Cells[0].Value.ToString(); yazıyoruz

Bu sayede Data Grid View de Sql verilerini gösttere biliyoruz.

54.SUNUCU ADINI PROGRAMA KAYDETME:

Programa ilk girildiğinde ayarlar butonuna tıklayıp açılan pencereye sql sunucu adını yazmanız gerekiyor.

1. if (Baglanti\_Test(txtVeriAdi.Text))

{

Ayarlar.Default.sunucu = txtVeriAdi.Text;

Ayarlar.Default.Save();

MessageBox.Show("Veritabanı ayarları kaydedildi\nProgram yeniden başlatılacak", "BİLGİ", MessageBoxButtons.OK, MessageBoxIcon.Information);

Application.Restart();

}

else

{

MessageBox.Show("Bağlantı testi başarısız olduğu için ayarlar kaydedilmedi !", "UYARI", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

} butona tıklandığında yapması gerekenleri yazıyoruz.

1. Bu kod sayesinde program içersindeki settings dosyasına kaydeder ve program kapatılıp açılsa bile kayıtlı kalır.
2. baglanti = new SqlConnection("Data Source=" + sunucu + ";Initial Catalog=Anaokulu;Integrated Security=True"); bu kod sayesinde veri tabanı bağlantısı sağlanıyor.

55.SQL’E VERİLERİ YÜKLEMEK İÇİN GERKEN KODLAR:

1. string sorgu = "INSERT INTO Rndv(AdSoyad,Numara,GrsSebebi,Tarih,GrsNotlari,Eposta) VALUES(@adi,@num,@sebeb,@tarihi,@notlar,@eposta)"; burda ki kodlar sayeside sql de olan veri isimlerini c# a çeviriyotuz.
2. komut.Parameters.AddWithValue("@Padi", Adtxt.Text);

komut.Parameters.AddWithValue("@Psoyad", SoyadTxt.Text);

komut.Parameters.AddWithValue("@Ptc", TcNotxt.Text);

komut.Parameters.AddWithValue("@Ptcserino", TcSeriNoTxt.Text);

komut.Parameters.AddWithValue("@Ptelefon", GsmTxt.Text);

komut.Parameters.AddWithValue("@Pevtel", EvTelTxt.Text);

komut.Parameters.AddWithValue("@Padresi", Adrestxt.Text);

Bu kodlarda sql sistemine kaydettiriyor.

56.SQL VERİLERİNİ SİLMEK İÇİN GEREKEN KODLAR:

1. string sorgu = "DELETE FROM Personel WHERE ID=@ID";

komut = new SqlCommand(sorgu, baglanti);

komut.Parameters.AddWithValue("@ID", Convert.ToInt32(ID.Text));

baglanti.Open();

komut.ExecuteNonQuery();

baglanti.Close();

Prsbilgi();

ID ile sql’de bulunan verinin bütün satır alınır ve butona basıldığı zaman veriyi siler.

57.SQL VERİLERİNİ GÜNCELLEMEK İÇİN GEREKEN KODLAR:

1. string sorgu = "UPDATE Rndv SET AdSoyad=@adi,Numara=@num,GrsSebebi=@sebeb,Tarih=@tarih,GrsNotlari=@notlar,Eposta=@eposta WHERE ID=@ID"; ID ile sql verilerini güncelliyoruz verileri Data Grid View üzerinden tıkladığımızda veriler textboxlar içersine gelir verileri değiştirip güncelle butonuna tıkladığımızda veriler güncellenir.

58.SQL İÇERSİNE RESİM KAYDETME:

1. Öncelikle picture box ekliyoruz programımıza.
2. openFileDialog1.ShowDialog();

pictureBox1.ImageLocation = openFileDialog1.FileName;

resim.Text = openFileDialog1.FileName; 1 adet buton yaratıp içersine bu kodları yazıyoruz ve program içersinde bu butona tıklayıp açılan pencereden resmi seçiyoruz ve picture box içersine resim geliyor ve kaydete tıklayınca resim sql servera kaydediliyor.

PROGRAMIN RENGİNİ DEĞİŞTİRMEYE YARAYAN KODLAR:

1. DialogResult tus;

tus = colorDialog1.ShowDialog();

if (tus == DialogResult.OK)

{ this.BackColor = colorDialog1.Color; } Buton yaratıp içerisine bu kodları yazıyoruz ve bu sayede sayfa renklerini değiştiriyoruz.

59.KULLANICI GİRİŞİ YAPMAYA YARIYAN KODLAR:

1. try {

baglanti.Open();

string sql = "Select \* From parola where Klcadi=@Klcadi AND Sifre=@sifresi";

SqlParameter prm1 = new SqlParameter("Klcadi", usertxt.Text.Trim());

SqlParameter prm2 = new SqlParameter("sifresi", passtxt.Text.Trim());

SqlCommand komut = new SqlCommand(sql, baglanti);

komut.Parameters.Add(prm1);

komut.Parameters.Add(prm2);

DataTable dt = new DataTable();

SqlDataAdapter da = new SqlDataAdapter(komut);

da.Fill(dt);

Prsbilgi();

if (dt.Rows.Count > 0)

{

AdminPage fr = new AdminPage();

fr.Show();

this.Hide();

Hide();

}

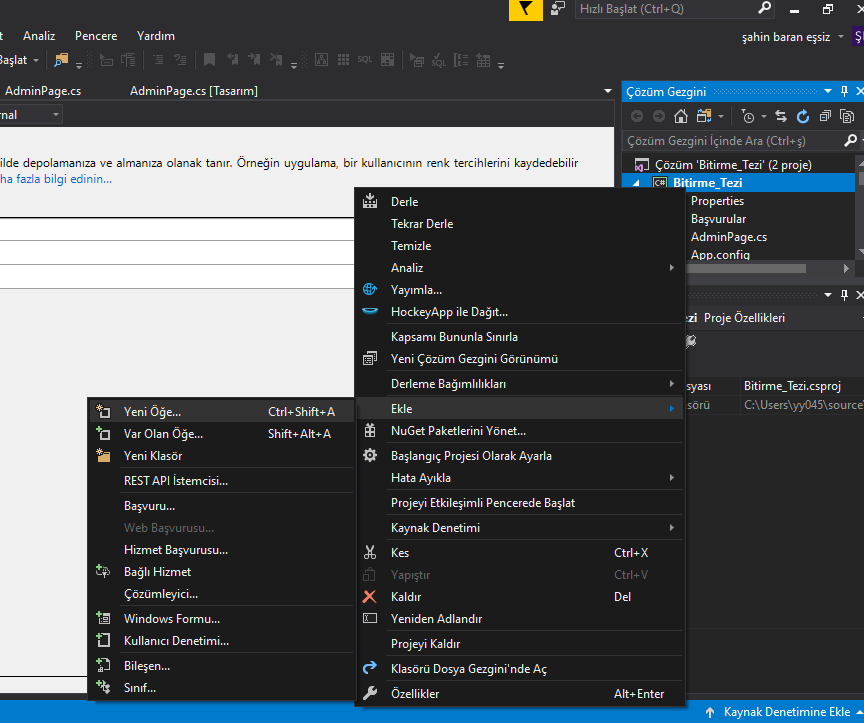
} Bu kodları kullanarak giriş yaptırıyoruz.

1. catch (Exception)

{ MessageBox.Show("hatalı giriş yapıldı tekrar deneyiniz."); }

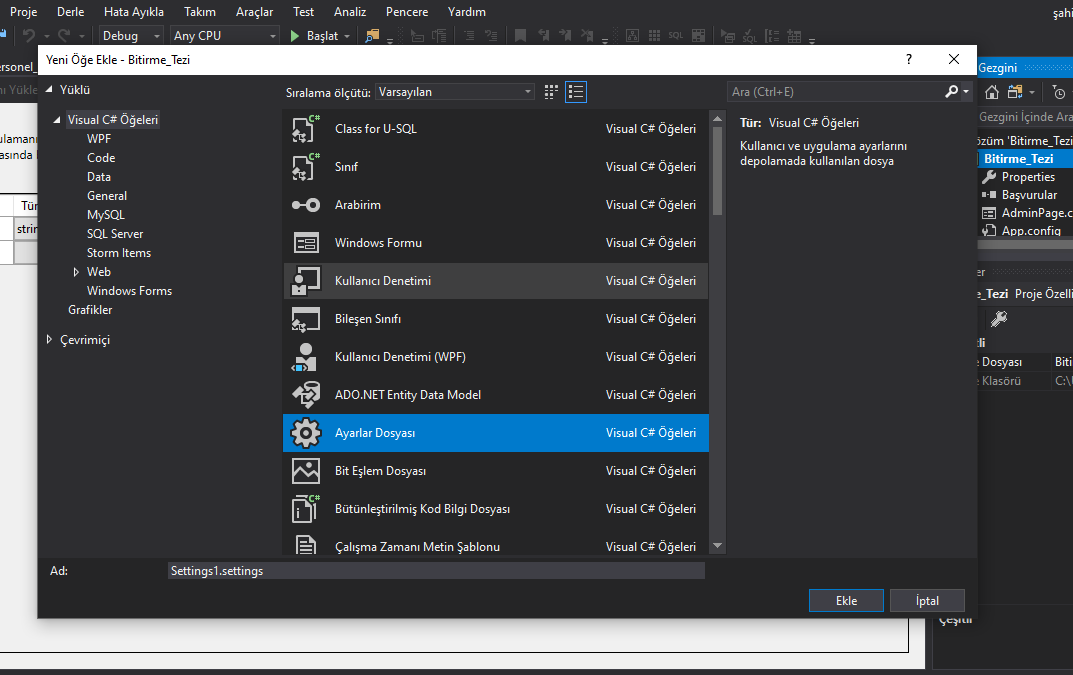
Eğer hatalı giriş yapılırsa messagebox açılıp hatalı giriş yapıldığını söyler.

60.PROGRAMA HERHANGİ BİLGİ KAYIT ETME:



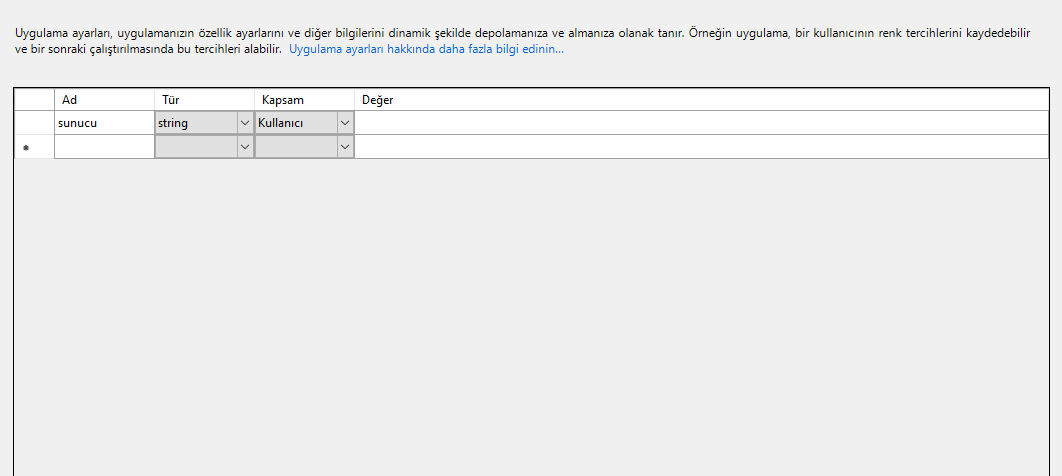
Şekil *5.1 settings dosyası ekleme*

Proje dosyasına sağ tıklayıp ekle yeni öğe diyoruz.



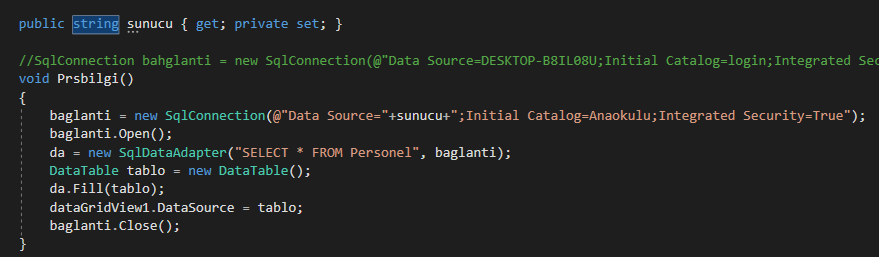
Şekil *5.2 settings dosyası ekleme*

Ayarlar dosyasını adını ayarlayıp ekliyoruz.



Şekil *5.3 settings dosyası*

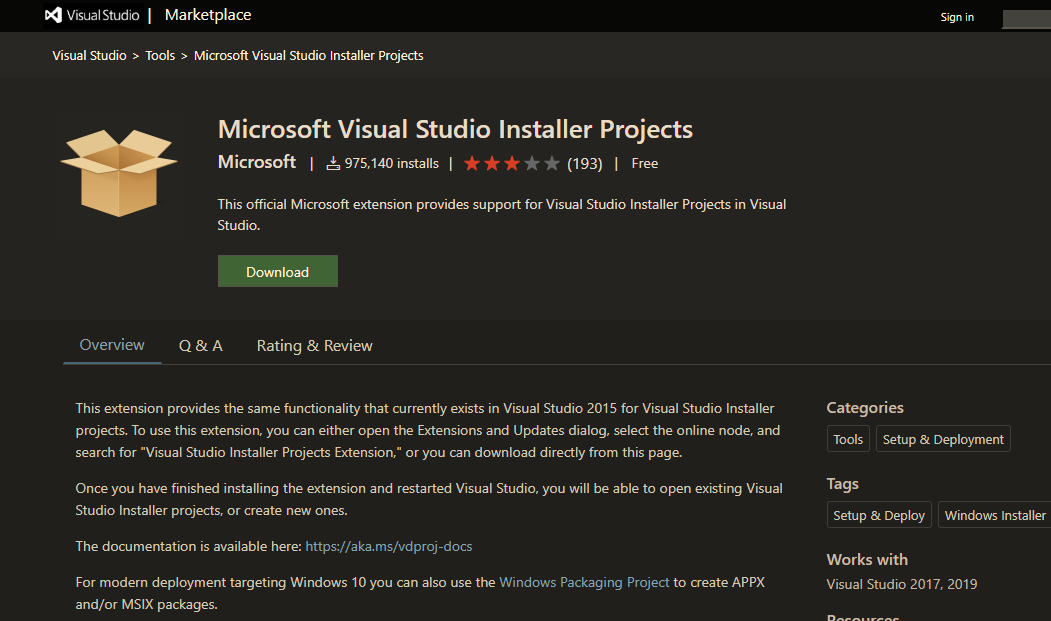
Programda kaydetmek istediğimiz değişkeni giriyoruz.



Şekil *5.4 kullanım örneği*

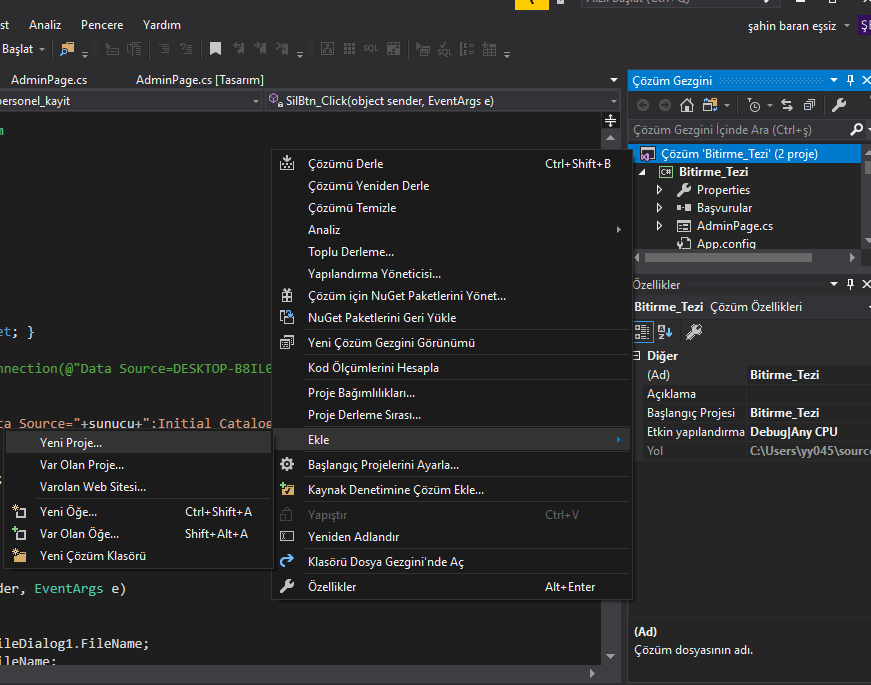
61.SETUP DOSYASI YAPIMI:

Anaokulu uygulamasının setup dosyasının ilk adımında Microsoft Visual Studio İnstaller yapılan bilgilsayar içinde kurulmuştur. Bu uygulama bizim setup dosyasını hazırlamamız için VisualStudio nun içine kurma işlemi gerçekleştirilecektir. Bu uygulama Visual Studio nun kendi setup oluşturma sihirbazıdır.

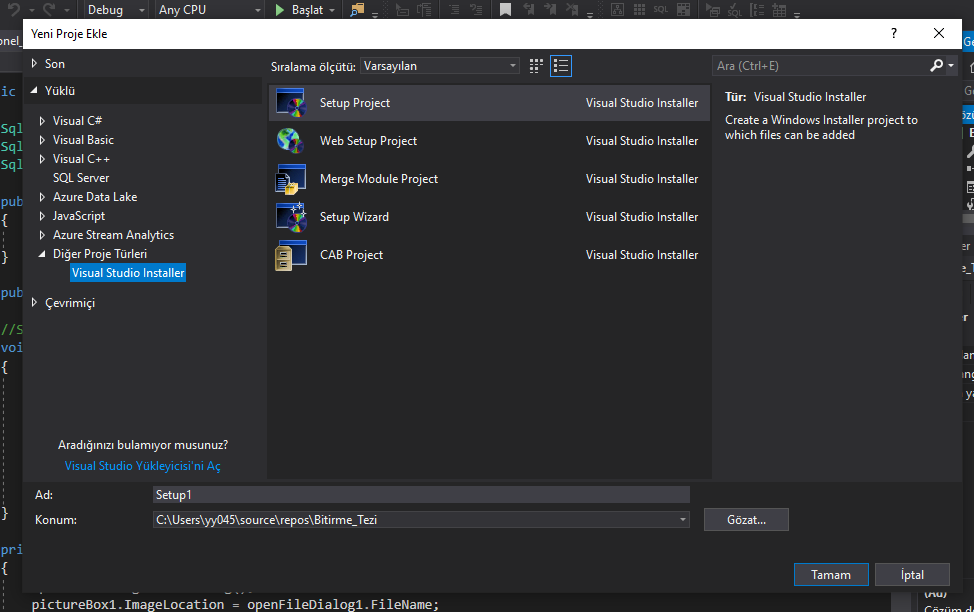


Şekil *6.1 setup eklentisi indirme sitesi*

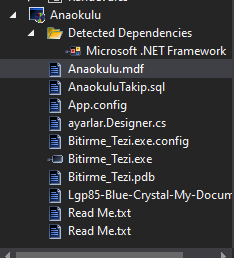
Kurulduktan sonra Visual Studio üzerinden file sekmesine tıklandıktan sonra new project ibaresine tıklıyoruz karşımıza çıkan ekranda **“Other Projects Type”** yazısına tıklanmıştır. Burada karşıya 3 seçenek çıkacaktır. **“Visual Studio”** installer seçilmiştir. Daha sonrasında kurulum dosyasına isim verilebilir. Name kısmından ismi değiştirilebilir. Sonrasında isimlendirmenin 3. Satırında **“Add the solution”** ifadesini seçilmiştir.



Şekil *6.2 setup eklemek için gereken adımlar*

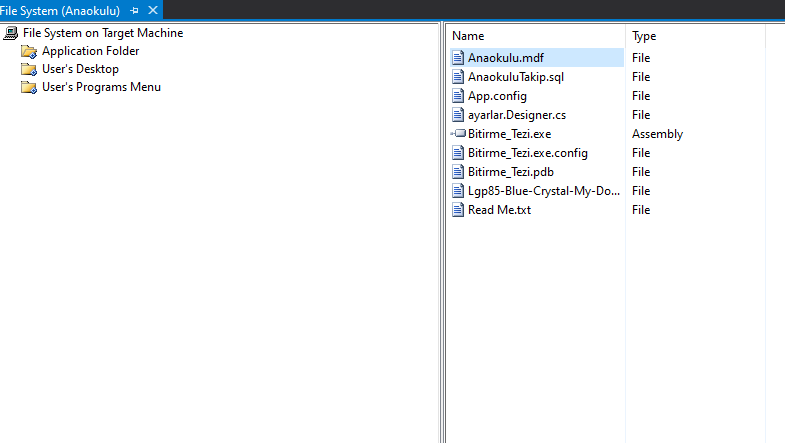


Şekil *6.3 setup dosyasının bulunduğu yer*

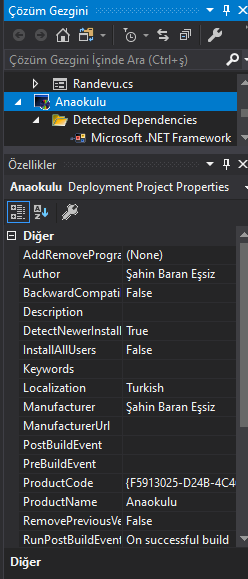


Şekil *6.4 setup çözüm gezgininde bu şekilde gözükür*

Karşıya 2 ye bölünmüş bir pencere açılmıştır. 1. Pencereden **“Application Folder”** uygulama klasör yazısına tıklanmıştır. Daha sonra sağ tık ile add ve file ifadelerine tıklanıp uygulama içine yerleştirilecek klasörler seçilecektir. Sonrasında Visual Studio dosyasında yaptığımız projeyi debug klasöründen hepsini seçiyruz. Orada farkedilecektir ki setup dosyasıda orada çıkmış olacaktır. Tekrar add folder diyoruz gereken dosyaları isimlendirerek oluşturulacaktır. Gereken klasörler oluşturulduktan sonra kullanıcı kurduğunda masaüstünde sadece .exe uygulaması gelmesi için **“User’s Desktop”** klasörüne tıklanmıştır. **2. Alanda “Create new shotcut”** sağ tıklayıp belirtilen yazıya tıklanmıştır. Sonra projemizin var olduğu dosyayı bulup, oluşturulan projenin .exesi alınmıştır. Daha sonra sağ tık özellikler bölümünü windows yapılması gerekiyor. Daha sonra özellikler kısmında exenin yani kurulduktan sonra kısayolu resimini seçiyoruz. Sözlük için kitap kısayolu resimi aktarılmıştır.



Şekil *6.5 setup dosyalarının eklendiği yer*



Şekil *6.6 setup dosyasının ince ayarları*

Bu kısımda programın icon, programlayan kişiyi, şirket adı vb. ayarlar yapılabilir.

Son işlemlere girmeden önce kullanıcı programı yüklerken klasör içinde oluşturması içinde **“Users Program Menu”** daha sonra klasörü isimlendirilmesi gerçekleştirilip klasörün

içine programın exesi atılmıştır. İlk dosyanın özellikler kısmına gelip uygulama kurulurken isim değişikliği yapılmıştır. Özellikler kısmında 2 ibare **“true**” olarak değiştirildikten sonra kurulum dosyamız ve kurulum sihirbazımız hazır duruma gelmiştir. Madde madde kısa cümlelerle bunun anlatımı gerçekleştirilecektir.

* Microsoft Visual Studio İndirin
* Yeni proje diyerek add the solution kısmına tıklayın
* Application folder isimli dosyaya sağ tıklayın
* Add- file diyerek uygulamanın dosyalarını seçin
* Seçildikten sonra User’s Desktop dosyasını seçin
* 2. alana sağ tıklayın ve **“Create New Shotcut”** tıklayın
* Masaüstünde kısayol oluşturulması için projenin .exe dosyasını seçin
* Kısayol resimleri vb. şeyler eklemek için özellikler kısmına tıklayın daha sonra istenilen resimi seçin ve kurulum dosyası başarılı bir şekilde oluşturulmuştur.

62. KAYNAKLAR

Murat Yücedağ, C# Form araçlarının özellikleri , https://www.udemy.com / (Ziyaret tarihi: 10 Aralık 2019)

Sen de kod yaz eğitim platformu, C# veri aktarma özellikleri , [https://www.youtube.com/watch?v=Z-9cV6-](https://www.youtube.com/watch?v=Z-9cV6-m0Mc&list=PLSuhOGv534vTFRq9dFB9Yn7su-4hIc0xg&index=33) / (Ziyaret tarihi:10 Aralık 2019)

Sen de kod yaz eğitim platformu, SQL komutları , <https://www.youtube.com/watch?v=U-i1iIixoqU&list=PLSuhOGv534vTFRq9dFB9Yn7su-4hIc0xg&index=35> / (Ziyaret tarihi: 10 Aralık 2019)

Sen de kod yaz eğitim platformu , SQL Sorguları , [ttps://www.youtube.com/watch?v=cauKndnQUUE&list=PLSuhOGv534vTFRq9dFB9Yn7su-4hIc0xg&index=37](https://www.youtube.com/watch?v=cauKndnQUUE&list=PLSuhOGv534vTFRq9dFB9Yn7su-4hIc0xg&index=37) / (Ziyaret tarihi: 10 Aralık 2019)

Sen de kod yaz eğitim platformu C# değişkenler , <https://www.youtube.com/watch?v=wKMfW8U75nY&list=PLSuhOGv534vTFRq9dFB9Yn7su-4hIc0xg&index=47> / (Ziyaret tarihi 17 Aralık 2019)

Sen de kod yaz eğitim platformu , C# veri bağlantısıaktarma , <https://www.youtube.com/watch?v=Yb9bqIcuKuM&list=PLSuhOGv534vTFRq9dFB9Yn7su-4hIc0xg&index=48> / (Ziyaret tarihi 18 Aralık 2019)

URL-1: <https://www.yazilimkodlama.com/programlama/c-sql-server-veritabani-baglantisi-select-insert-update-delete/> / (Ziyaret tarihi 18 Aralık 2019)

URL -2: <https://webdunya.com/c-sharp-sql-server-veri-tabani-baglantisi/> / (Ziyaret tarihi 22 Aralık 2019 )

URL 3- :<https://marketplace.visualstudio.com/items?itemName=VisualStudioClient.MicrosoftVisualStudio2017InstallerProjects> / (Ziyaret tarihi 5 Ocak 2020)

URL 4-: <http://forum.csharpnedir.com/forum_posts.asp?TID=38106&title=c-projeye-font-eklemek> / (Ziyaret tarihi 27 Aralık 2019)

URL-5: <https://www.algoritmaornekleri.com/c-sharp/windows-form/c-form-arka-plan-background-rengini-degistirme/> / (Ziyaret tarihi 27 Aralık 2019)

URL-6: <https://www.algoritmaornekleri.com/c-sharp/windows-form/combobox-elemanlari-icin-tooltip-ayarlama/> (Ziyaret tarihi 27 Aralık 2019)

URL-7: <https://www.algoritmaornekleri.com/c-sharp/windows-form/c-datagridview-icinde-textbox-ile-filtreleme-arama-yapma/> /(Ziyaret tarihi 27 Aralık 2019)

URL -8: <https://www.algoritmaornekleri.com/c-sharp/windows-form/c-datagridview1-secili-satiri-datagridview2-ye-aktarma/> (Ziyaret tarihi 28 Aralık 2019)

URL-9: <https://www.algoritmaornekleri.com/c-sharp/windows-form/c-listbox-secili-elemani-silme/> (Ziyaret tarihi 31 Aralık 2019)

URL-11 <http://edestek.kocaeli.edu.tr/course/view.php?id=375> (Ziyaret tarihi 19 Aralık 2020)

URL-12: <https://www.youtube.com/watch?v=Yb9bqIcuKuM&list=PLSuhOGv534vTFRq9dFB9Yn7su-4hIc0xg&index=48> (Ziyaret tarihi 19 Aralık 2020)

URL-13 <https://www.youtube.com/watch?v=_8Xt_eJt6NA&list=PLSuhOGv534vTFRq9dFB9Yn7su-4hIc0xg&index=61> (Ziyaret tarihi 23 Aralık 2020)

URL-14 <https://www.youtube.com/watch?v=U9h1lWlt3C4&list=PLSuhOGv534vTFRq9dFB9Yn7su-4hIc0xg&index=60> (Ziyaret tarihi 23 Aralık 2020)

URL-15 <https://www.yazilimkodlama.com/programlama/c-ile-excel-dosyasina-baglanma/> (Ziyaret tarihi 23 Aralık 2020)

URL-15: <http://www.gorselprogramlama.com/formlar-arasi-gecis-cnet/> (Ziyaret tarihi 27 Aralık 2020)

URL-16: <https://www.kodlamamerkezi.com/c-net/csharp-formlar-arasi-veri-gonderme-islemi/> (Ziyaret tarihi 27 Aralık 2020)

URL-17 <https://www.yazilimkodlama.com/programlama/arraylist-kullanimi-ornek-uygulama/> (Ziyaret tarihi 27 Aralık 2020)

URL-18 <https://www.yazilimkodlama.com/programlama/c-checkbox-kullanimi/> (Ziyaret tarihi 30 Aralık 2020)

URL-19: <https://www.yazilimkodlama.com/programlama/diziler-join-metodu-kullanimi/> (Ziyaret tarihi 30 Aralık 2020)

URL-20: <https://www.cozumpark.com/sql-server-2017-kurulumu/> (Ziyaret tarihi 30 Aralık 2020)

URL-21: <https://smartpro.com.tr/c-nedir/> (Ziyaret tarihi 13 Ocak 2020)

URL-22:  [<http://alkanfatih.com/metot-nedir-nasil-kullanilir-c-ile-metot-olusturma/>/](http://alkanfatih.com/ms-sql-nedir-kurulumu-ozellikleri/) (Ziyaret tarihi 11 Ocak 2020)

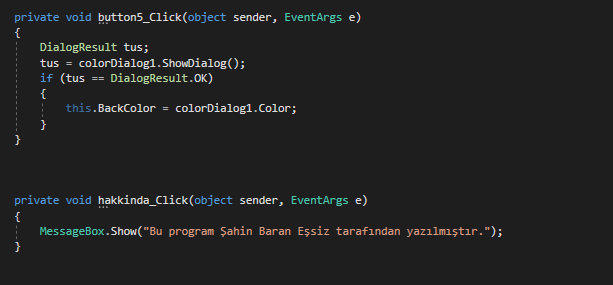
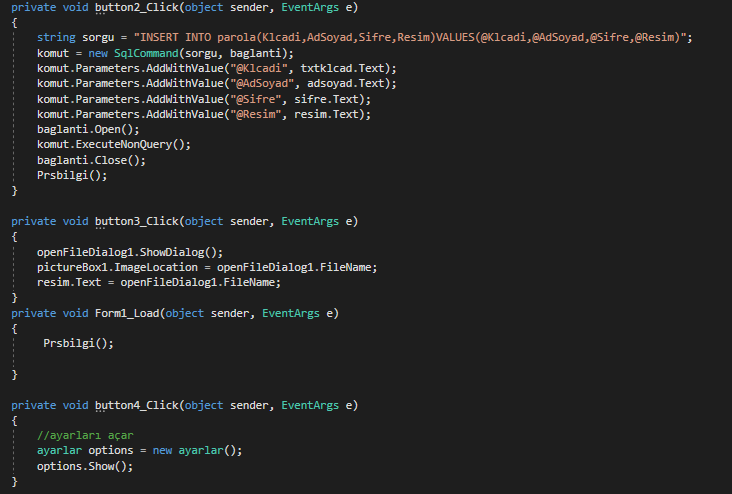
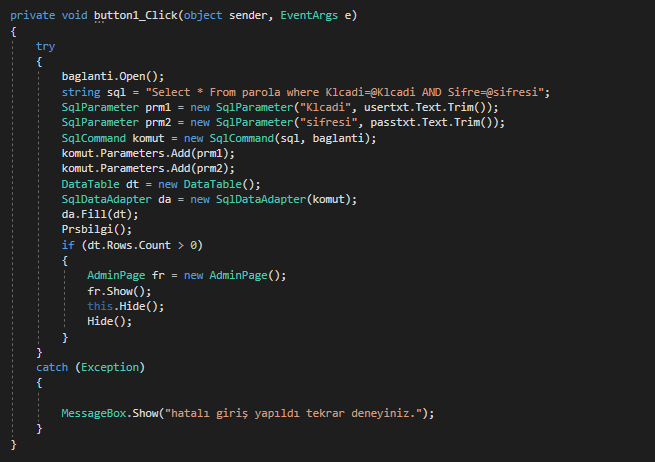
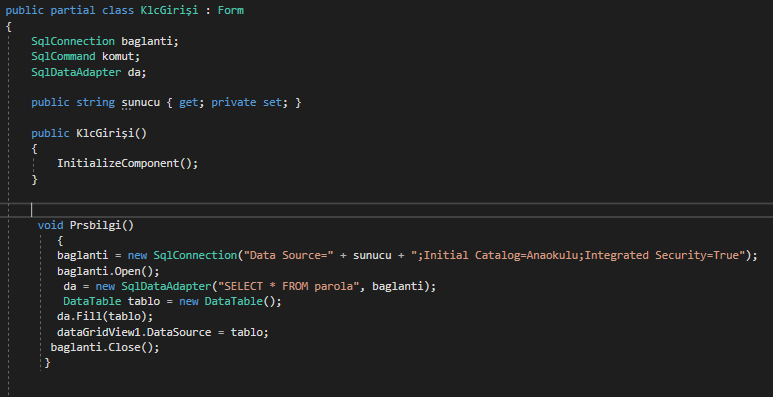
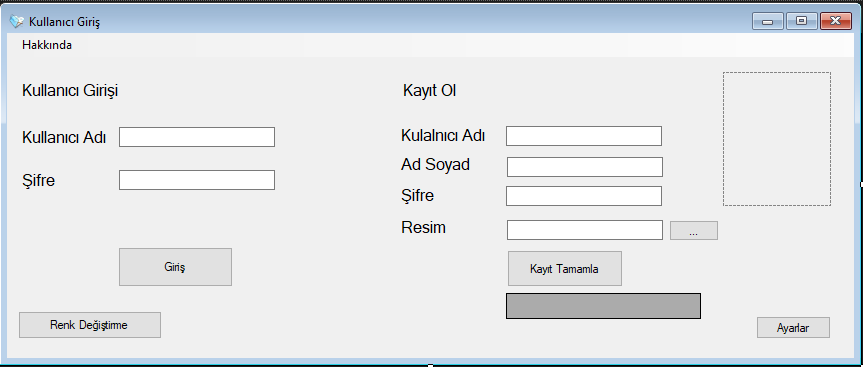
URL-23: <https://www.udemy.com/course/c-from-beginner-to-pro-build-a-windowsui-app-in-winforms/learn/lecture/13076214?start=0#overview> (Ziyaret tarihi 11 Aralık 2020)

URL-24:<https://www.udemy.com/course/c-sharp-kursu/learn/lecture/8921146?start=0#-overview> (Ziyaret tarihi 14 Ekim 2020)

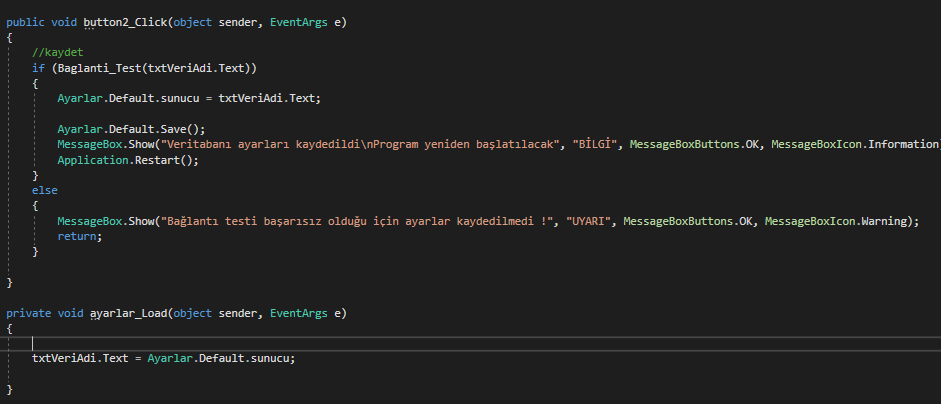
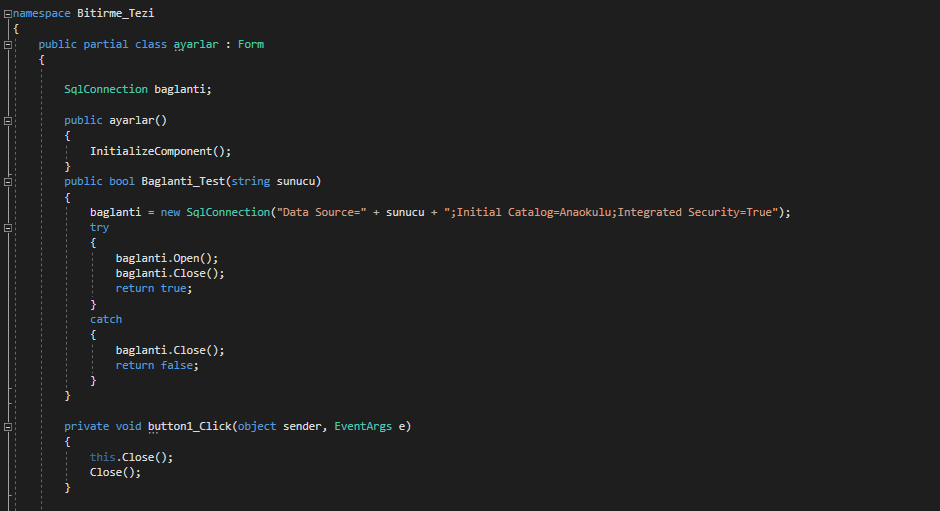
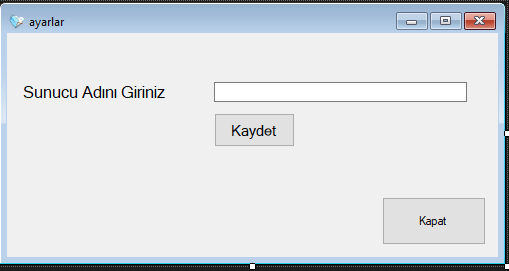
URL-25: <https://www.udemy.com/course/sql-for-beginners-s/learn/lecture/7897652?-start=0> (Ziyaret tarihi 12 Aralık 2020)

63.EKLER

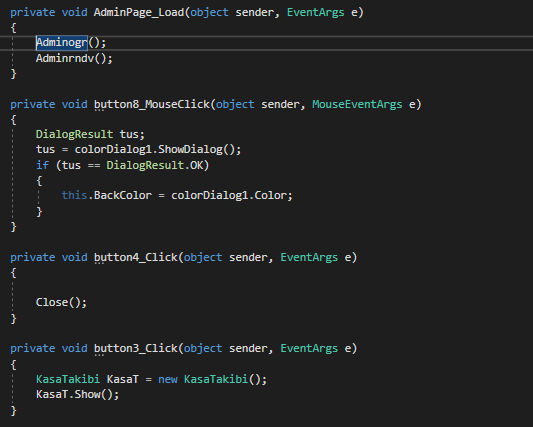
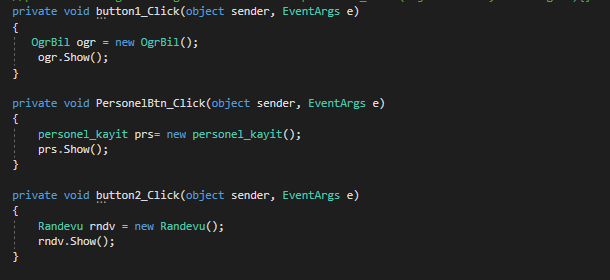
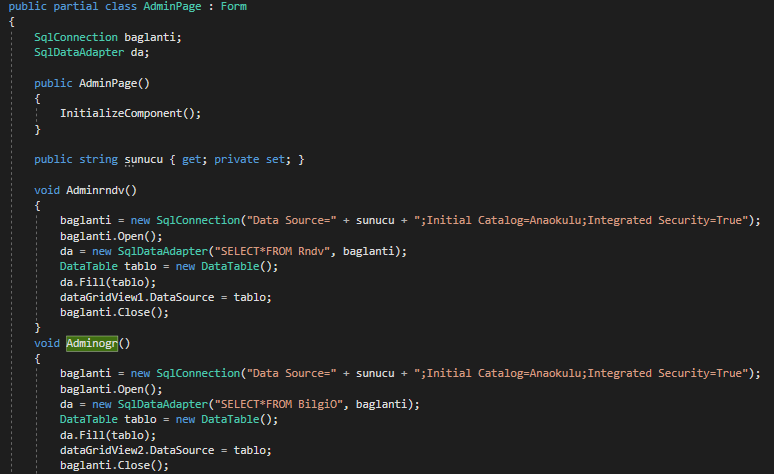
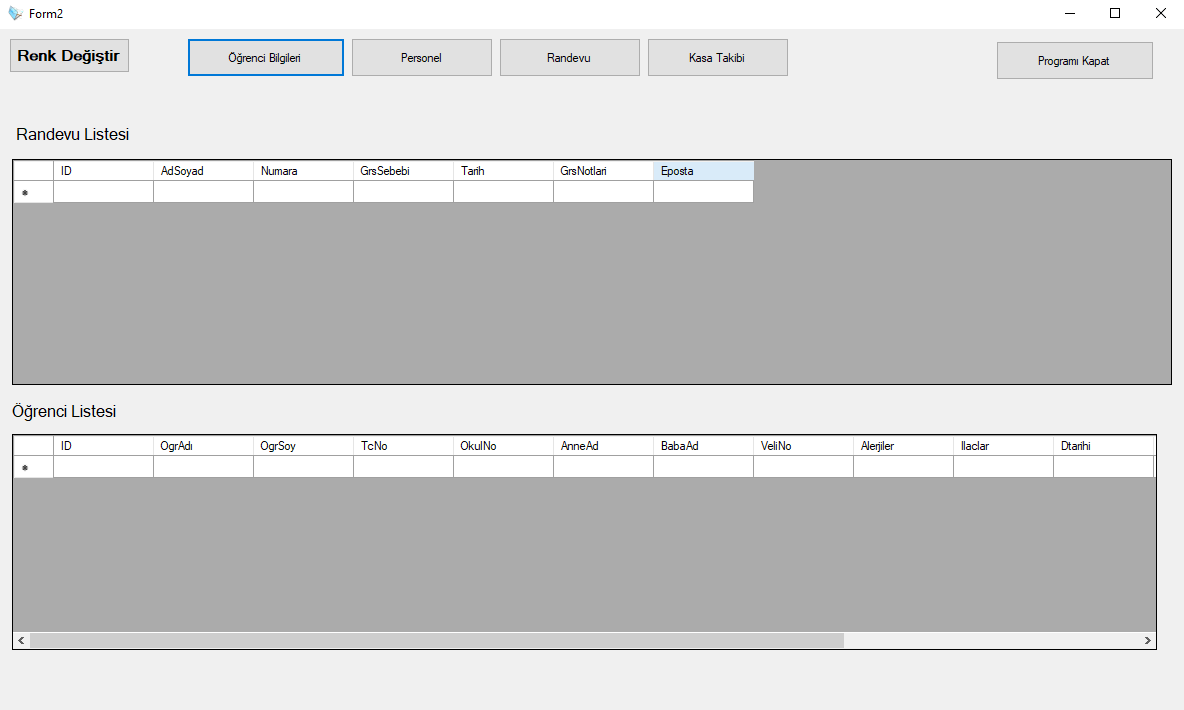
**EK1-**



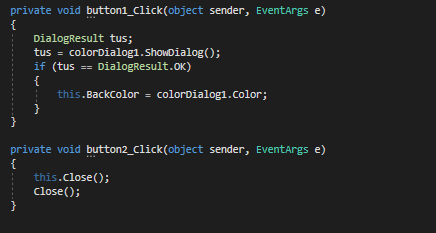
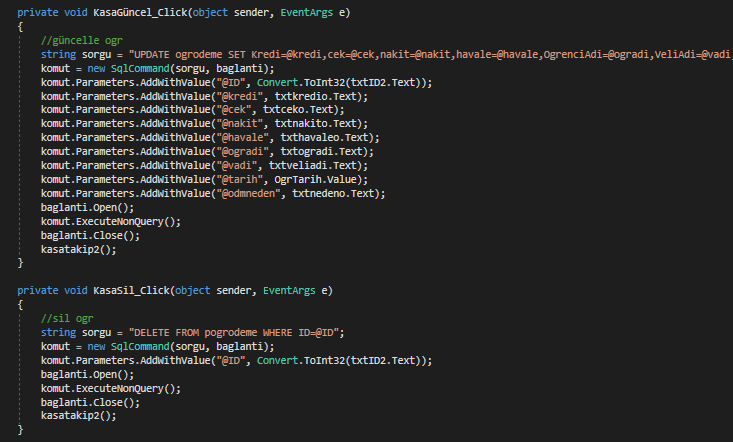
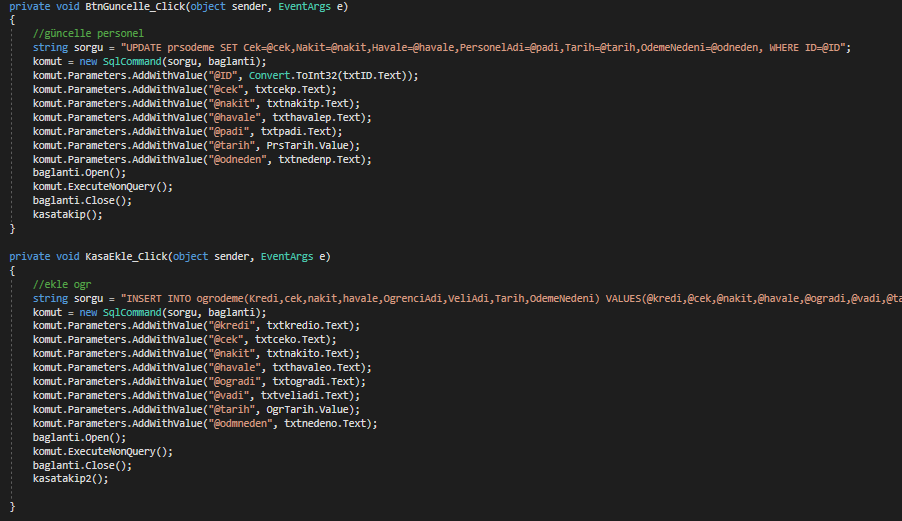
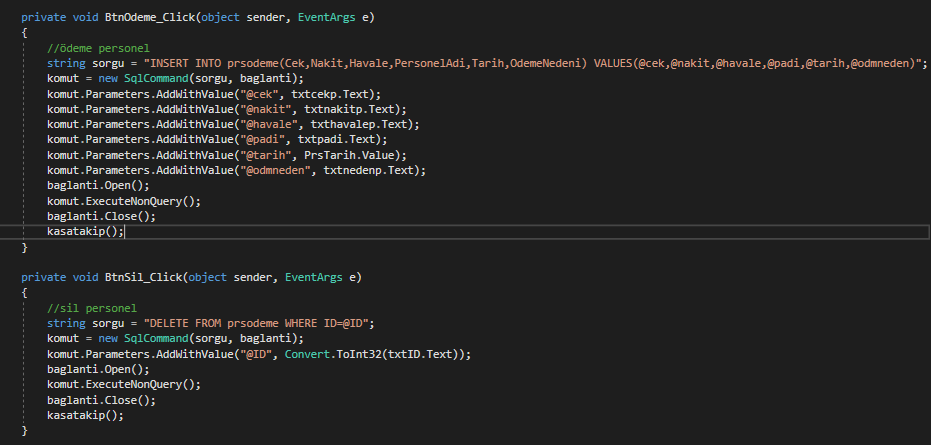
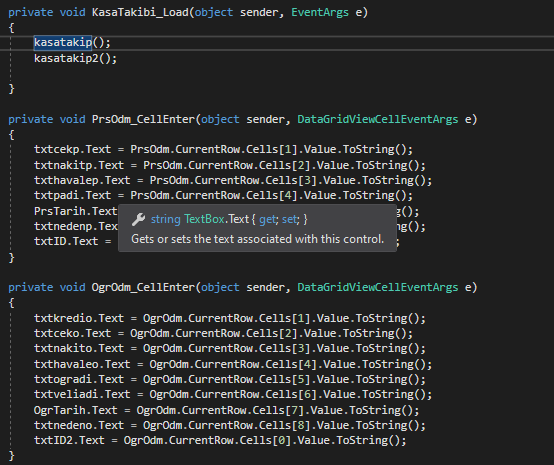
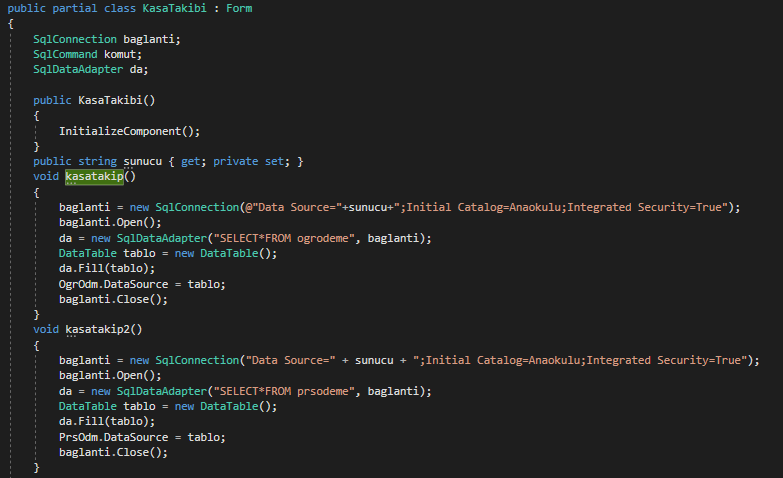
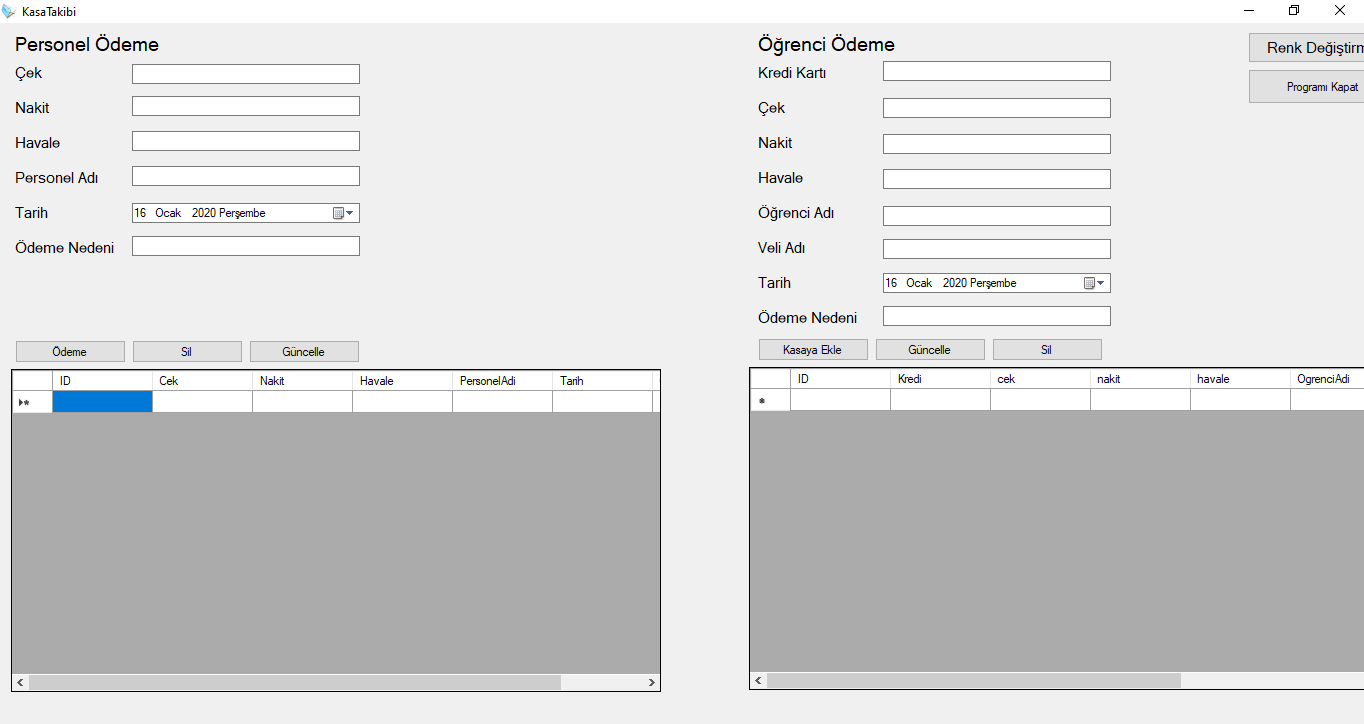
**EK2-**



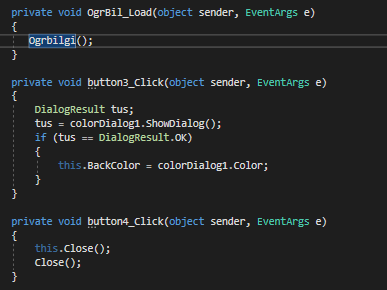
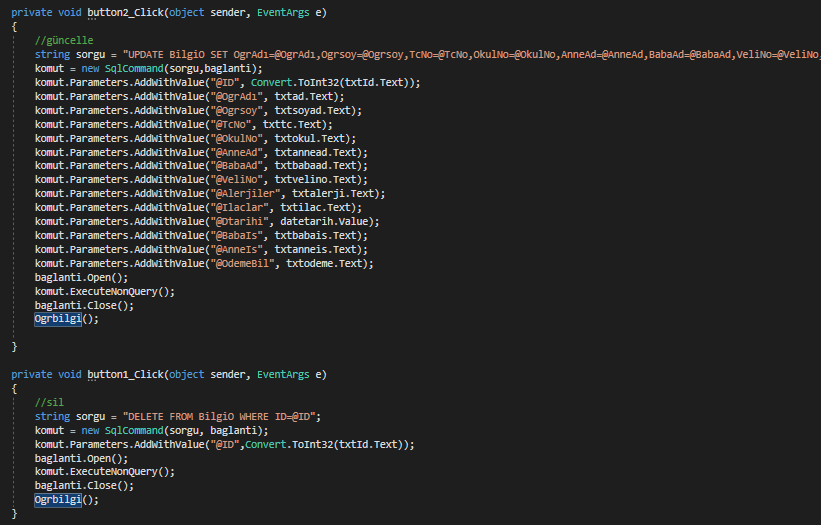
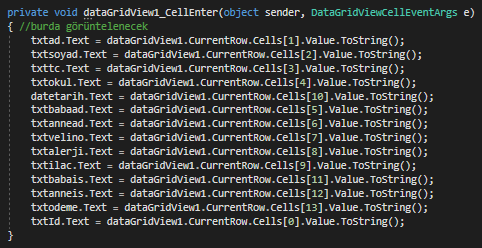
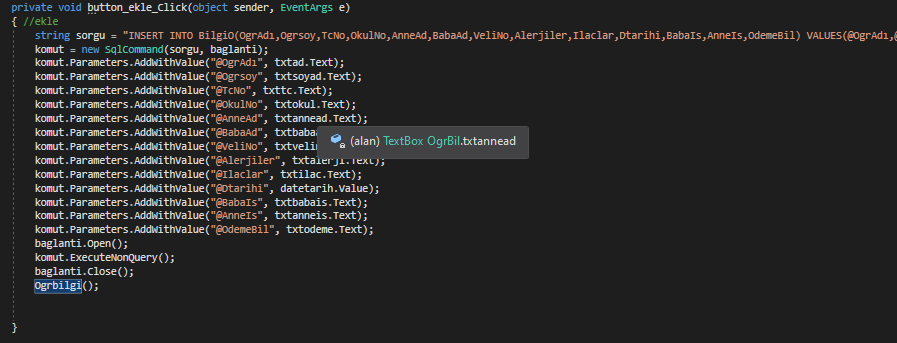
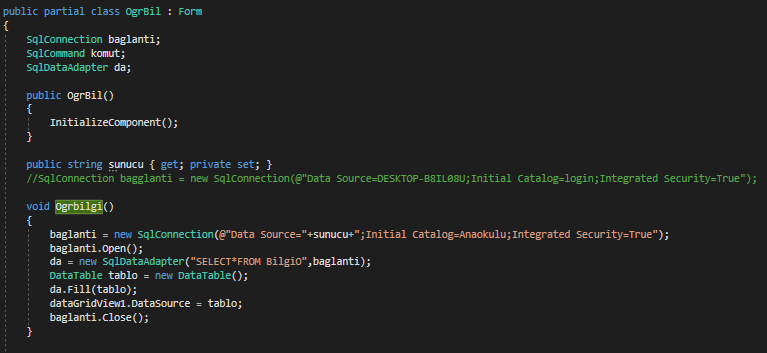
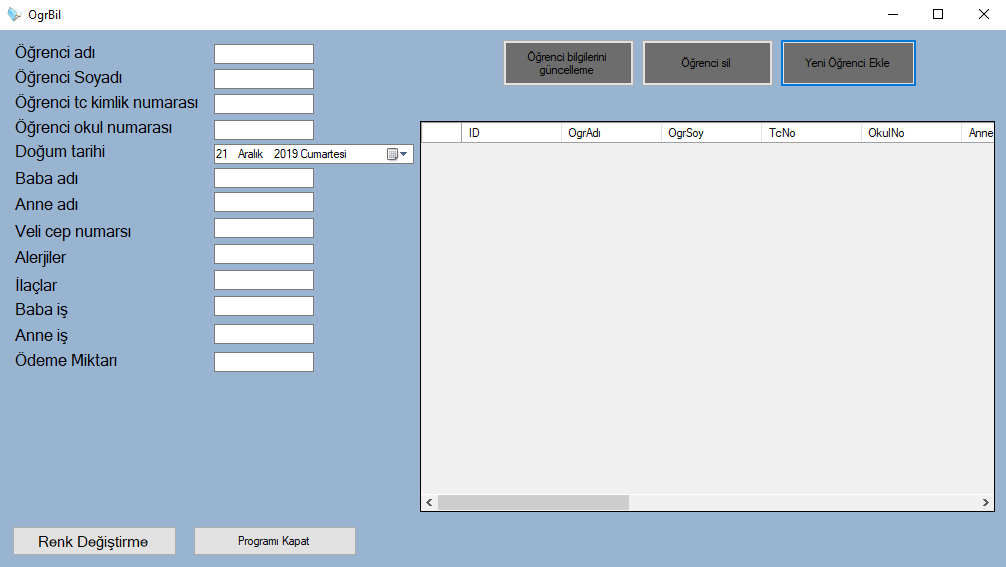
**EK3-**



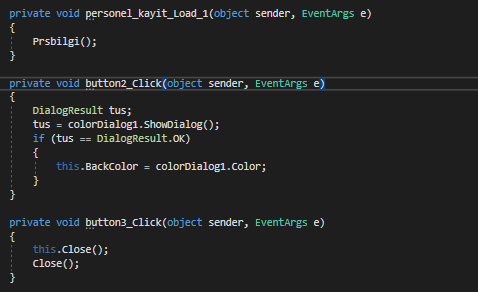
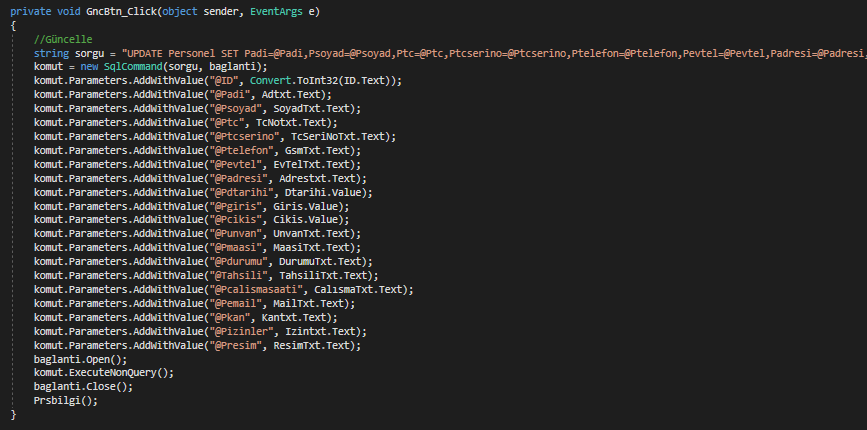
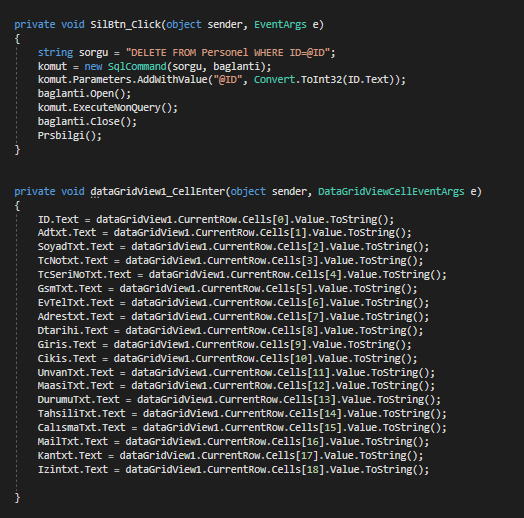
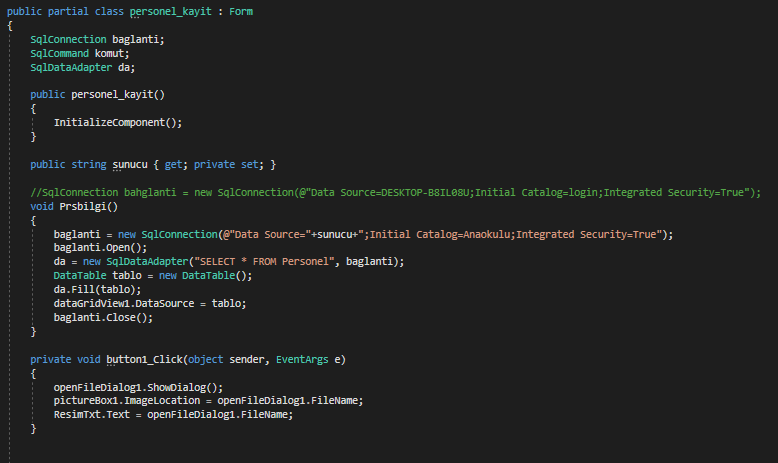
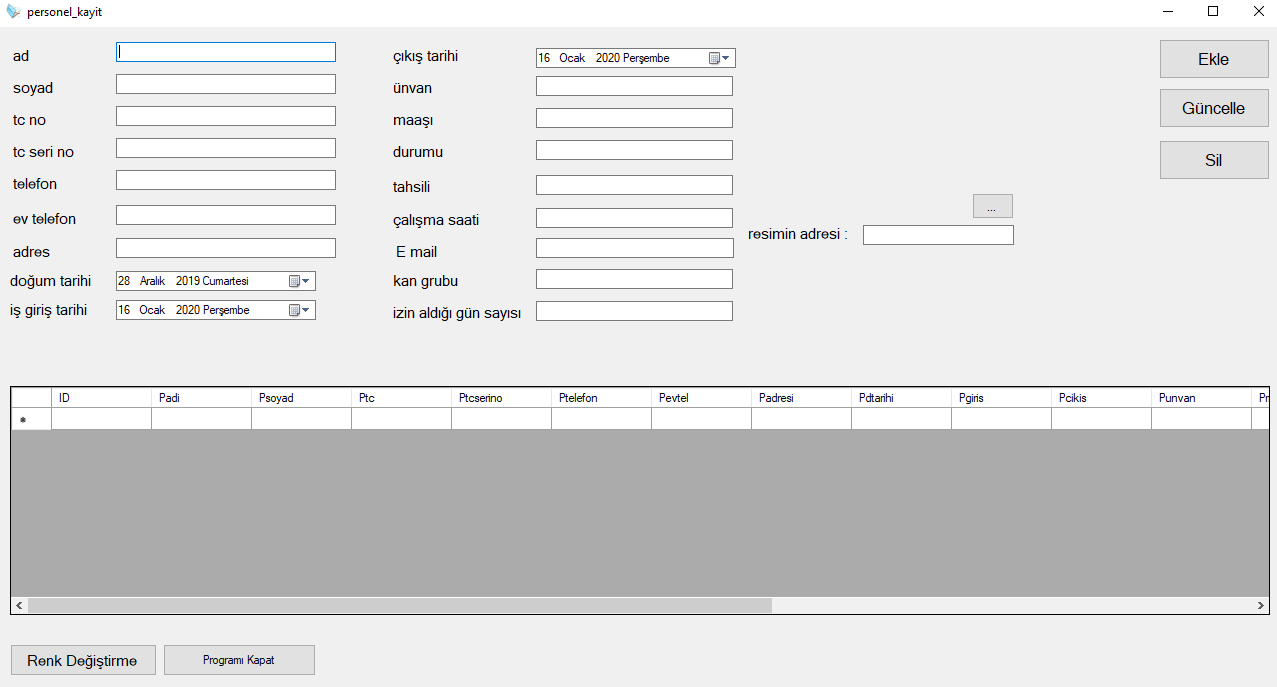
**EK4-**



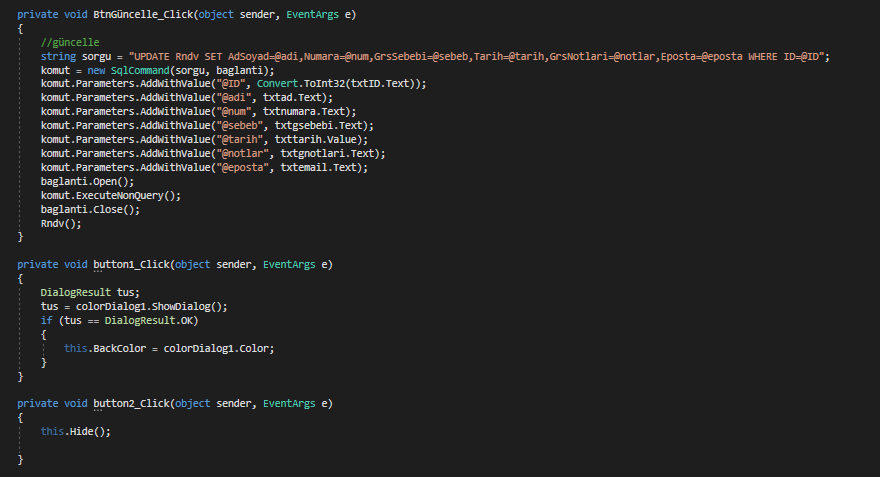
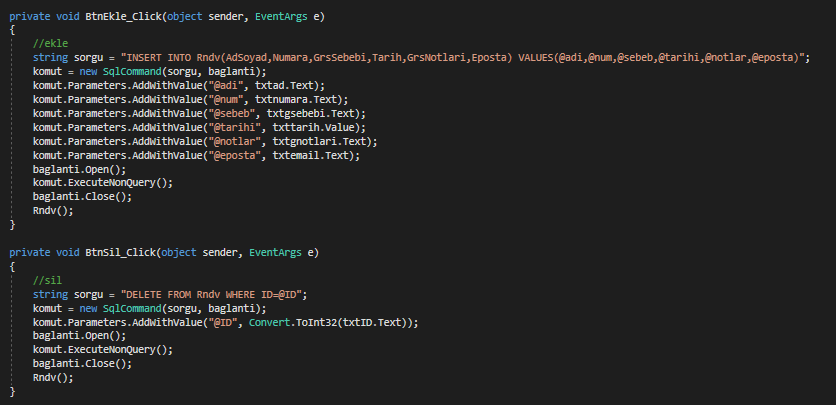
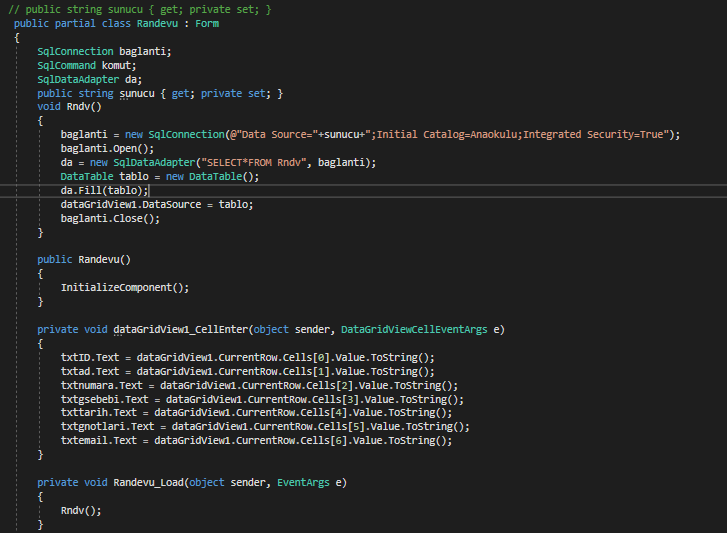
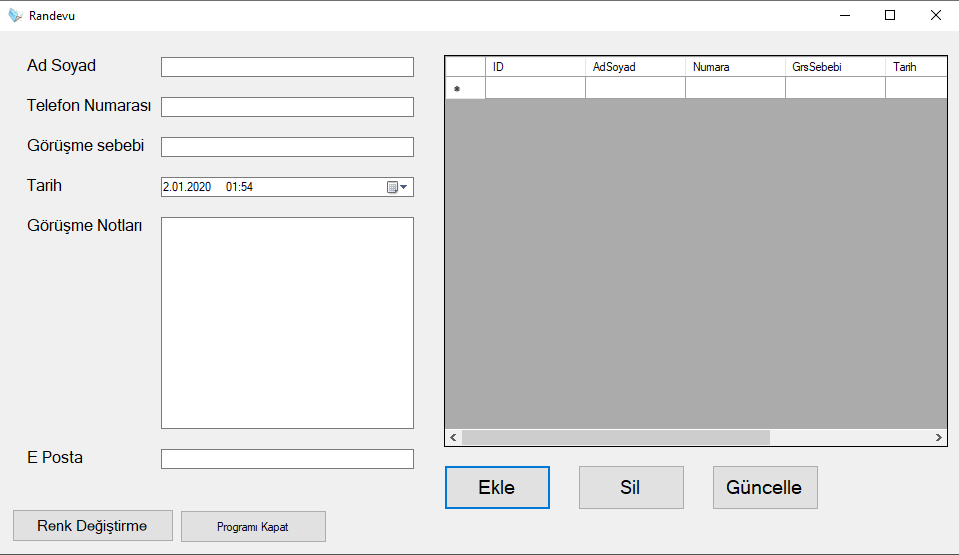
**EK5-**



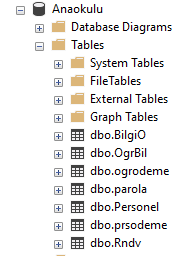
**EK6-**



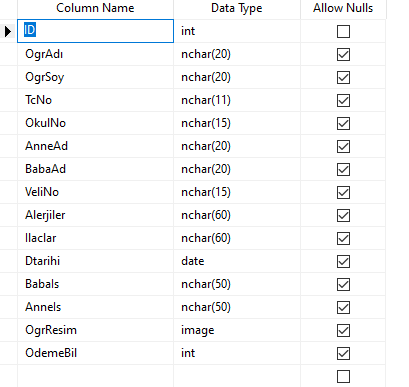
**EK7-**



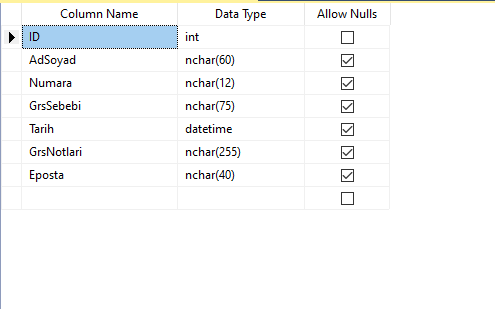
EK8-



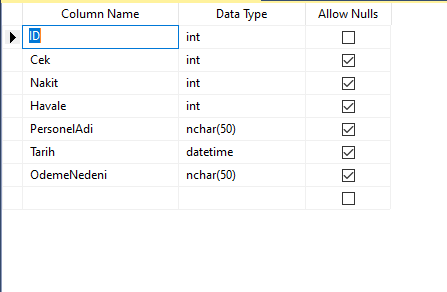
EK9-



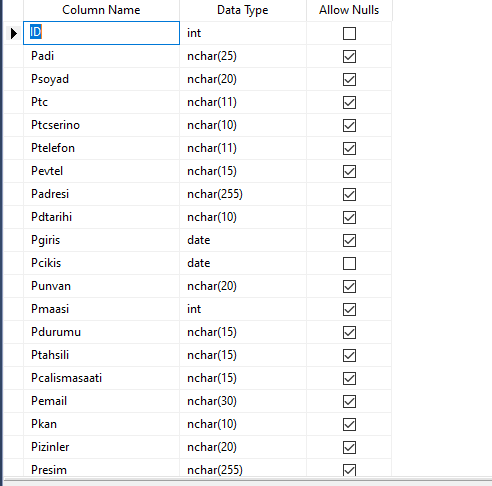
EK10-



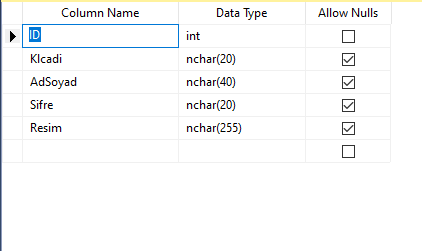
EK11-



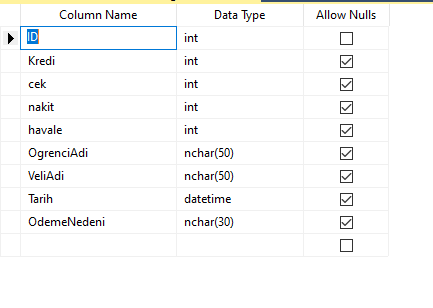
EK12-



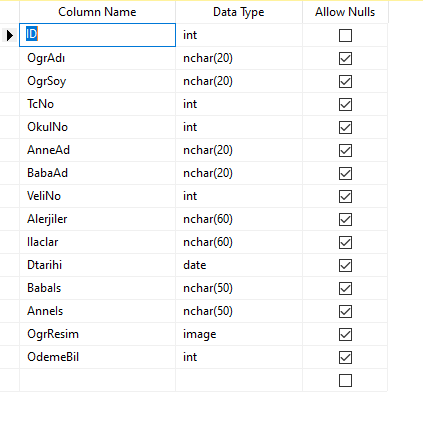
EK13-



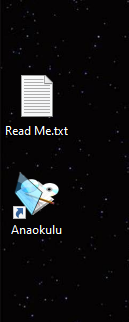
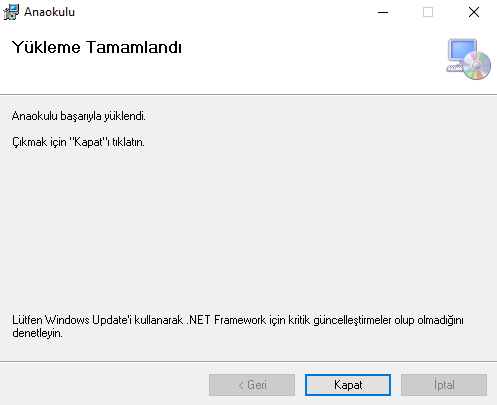
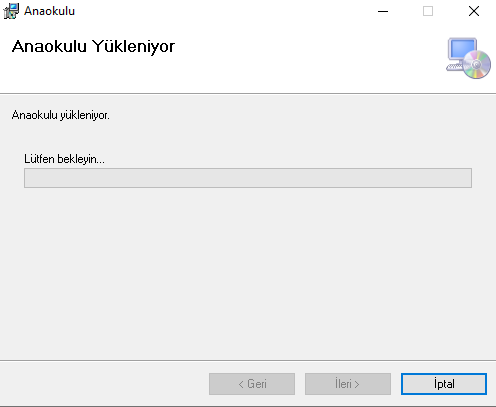
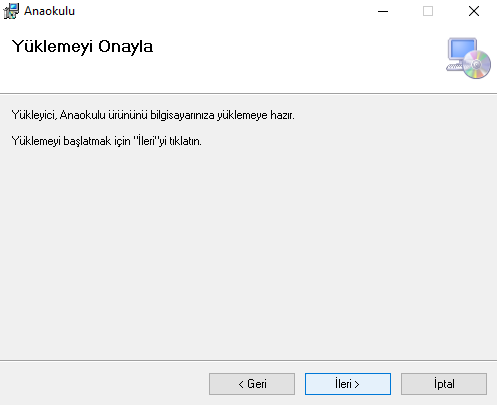
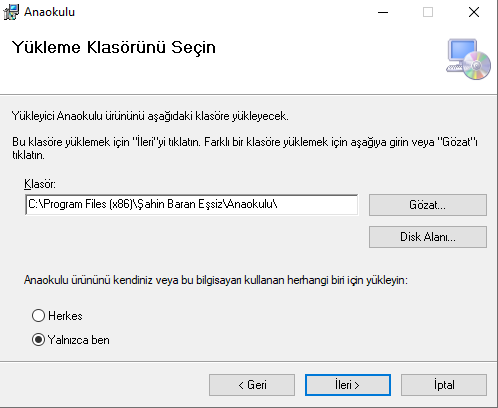
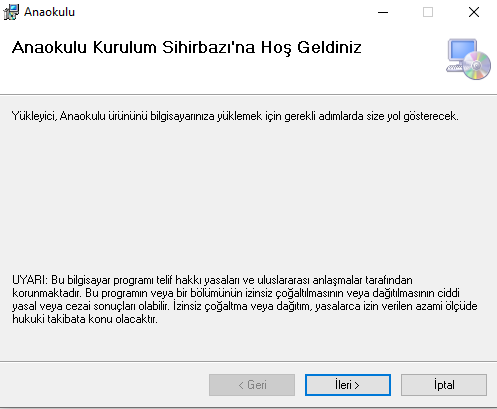
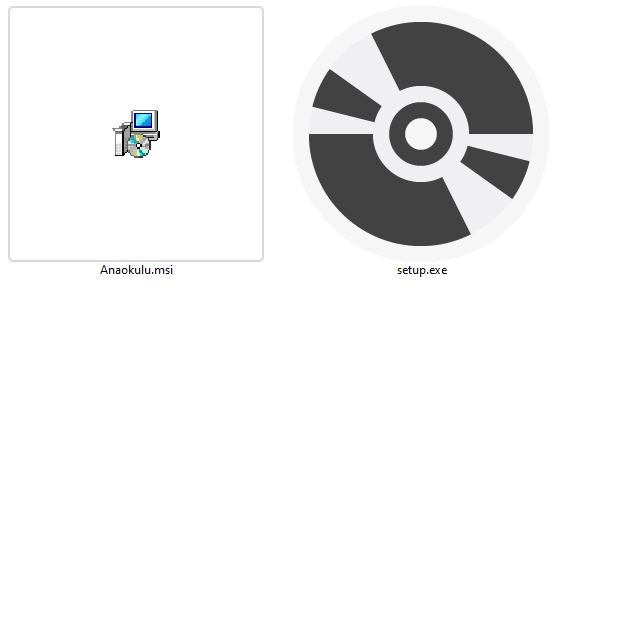
EK14-



EK15-



EK16-



**64. ÖZGEÇMİŞ**

1999 yılında Nisan ayında İSTANBUL/Eyüp’de doğdu. İlk okulu ve ortaokulu Kartal’da, liseyi Maltepede’da tamamladı. 2018 yılında üniversite yerleştirme sınavında başarılı olarak Kocaeli Üniversitesi, Kocaeli Meslek Yüksekokulu Bilgisayar Programcılığı bölümünü okumaya hak kazanmıştır. 2. Sınıf Bilgisayar Programcılığı öğrencisi olarak devam ediyor.