

Bölüm 2

C#

Dr. Öğr. Üyesi Murat TAŞYÜREK (kayubmprogramlama1@gmail.com)

28 Şubat 2024

Kayseri Üniversitesi, Bilgisayar Mühendisliği Bölümü

- Visual Studio Kurulum
- Console sınıfı
- Değişken tanımlama
- Dizi Tanımlama
- Döngüler
- Kontroller
- Ödev

Visual Studio Kurulum İndirme

Visual Studio Kurulum İndirme

https://www.google.com/search?client=firefox-b-d&q=download+visual+studio+community+2022

download visual studio community 2022

Tümü Görseller Videolar Haberler Alışveriş Daha fazla Araçlar

Yaklaşık 284.000.000 sonuç bulundu (0,70 saniye)

https://visualstudio.microsoft.com > community ▾

Visual Studio 2022 Community Edition – Son Ücretsiz Sürümü ...

12 Oca 2022 — Windows, Android ve iOS için modern geliştirici uygulamaları ...

Bu sayfayı 10.02.2022 tarihinde ziyaret ettiniz.

Visual Studio Paketler

Değiştiriliyor — Visual Studio Community 2022 — 17.0.6

İş yükleri Bağımsız bileşenler Dil paketleri Yükleme konumları

Web ve Bulut (4)

 ASP.NET ve web geliştirme
Docker desteği dahil olmak üzere ASP.NET Core, ASP.NET, HTML/JavaScript ve Kapsayıcıları kullanarak web uygulama...

☒

 Python geliştirme
Python için düzenleme, hata ayıklama, etkileşimli geliştirme ve kaynak denetimi.

☐

Masaüstü ve Mobil (5)

 .NET ile Mobil uygulama geliştirme
Xamarin kullanarak iOS, Android veya Windows için çoklu platform uygulamaları oluşturun.

☐

 .NET masaüstü geliştirme
.NET ve .NET Framework ile C#, Visual Basic ve F# dillerini kullanarak WPF, Windows Forms ve konsol uygulamaları...

☒

 C++ ile masaüstü geliştirme
Tercih ettiğiniz MSVC, Clang, CMake veya MSBuild gibi bir aracı kullanarak Windows için modern C++ uygulamaları...

☒

 Evrensel Windows Platformu geliştirme
C#, VB veya isteğe bağlı olarak C++ ile Evrensel Windows Platformu'na yönelik uygulamalar oluşturun.

☐

Yükleme ayrıntıları

.NET masaüstü geliştirme *

Eklendi

☒ .NET masaüstü geliştirme araçları
☒ .NET Framework 4.7.2 geliştirme araçları
☒ C# ve Visual Basic

İsteğe bağlı

☒ .NET için geliştirme araçları
☒ .NET Framework 4.8 geliştirme araçları
☒ Blend for Visual Studio
☒ Entity Framework 6 Tools
☒ .NET profil oluşturma araçları
☒ IntelliCode
☒ Anında Hata Ayıklayıcı
☒ Live Share
☒ ML.NET Model Builder
☐ F# masaüstü dil desteği
☐ PreEmptive Protection - Dotfuscator
☐ .NET Framework 4.6.2-4.7.1 geliştirme araç...
☐ .NET Taşınabilir Kitaplık Targeting Pack
☐ Windows Communication Foundation

Konum

C:\Program Files\Microsoft Visual Studio\2022\Community

Devam ederek, seçtiğiniz Visual Studio sürümünün [lisansını](#) kabul ediyorsunuz. Visual Studio ile başka yazılımlar indirme olanağı da sunuyoruz. Bu yazılımlar, [Üçüncü Taraf Bildiriminde](#) veya ürünle birlikte sağlanan lisansa belirtildiği gibi, ayrıca lisanslanır. Devam ederek bu lisansları da kabul etmiş olursunuz.

Gerekten toplam alan 4.63 GB

İndirilirken yükle

Değiştir

4/47

Visual Studio Paketler

Değiştiriliyor — Visual Studio Community 2022 — 17.0.6

İş yükleriBağımsız bileşenlerDil paketleriYükleme konumları

**Unity ile oyun geliştirme**
Güçlü bir çoklu platform geliştirme ortamı olan Unity ile 2B ve 3B oyunlar oluşturun.

☐

**C++ ile oyun geliştirme**
DirectX, Unreal veya Cocos2d tarafından desteklenen profesyonel oyunlar oluşturmak için C++'ın gücünden ta...

☐

**Veri depolama ve işleme**
SQL Server, Azure Data Lake veya Hadoop kullanarak veri çözümlerini bağlayın, geliştirin ve test edin.

☒

**Visual Studio Uzantısı geliştirme**
Visual Studio için yeni komutlar, kod çözümleyicileri ve araç pencereleri gibi eklenti ve uzantılar oluşturun.

☐

**C++ ile Linux geliştirme**
Linux ortamında çalışan uygulamalar oluşturun ve hatalarını ayıklayın.

☐

**C++ ile oyun geliştirme**
DirectX, Unreal veya Cocos2d tarafından desteklenen profesyonel oyunlar oluşturmak için C++'ın gücünden ta...

☐

**Veri bilimi ve analitik uygulamalar**
Python ve F# gibi dillerle veri bilimi uygulamaları oluşturmaya yönelik için diller ve araçlar.

☐

**Office/SharePoint geliştirme**
C#, VB ve JavaScript kullanarak Office ve SharePoint eklentileri, SharePoint çözümleri ve VSTO eklentileri oluşt...

☐

Yükleme ayrıntıları

- Visual Studio temel düzenleyicisi
- C++ ile masaüstü geliştirme *
- ASP.NET ve web geliştirme *
- .NET masaüstü geliştirme *
- Veri depolama ve işleme *
 - İsteğe bağlı
 - ☒ SQL Server Veri Araçları
 - ☒ Azure Data Lake ve Stream Analytics Araçları
 - ☒ .NET Framework 4.7.2 geliştirme araçları
 - ☐ F# masaüstü dil desteği

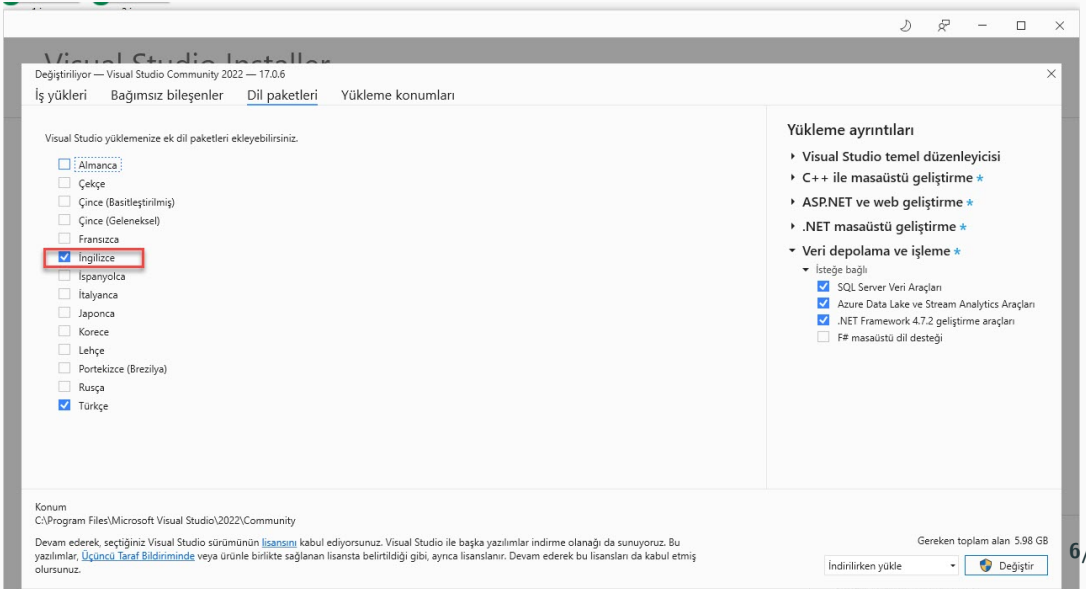
Konum
C:\Program Files\Microsoft Visual Studio\2022\Community

Devam ederek, seçtiğiniz Visual Studio sürümünün [lisansını](#) kabul ediyorsunuz. Visual Studio ile başka yazılımlar indirme olanağı da sunuyoruz. Bu yazılımlar, [Üçüncü Taraf Bildiriminde](#) veya ürünle birlikte sağlanan lisansa belirtildiği gibi, ayrıca lisanslanır. Devam ederek bu lisansları da kabul etmiş olursunuz.

Gerekten toplam alan 5.98 GB

İndirilirken yükle 

Visual Studio Dil Paketi



Console App Oluşturma

Create a new project

Recent project templates

- ASP.NET Web Application (.NET Framework) C#
- Console App (.NET Framework) C#
- Windows Forms App (.NET Framework) C#
- Console App C#
- Windows Desktop Application C++
- Console App C++
- Empty Project C++

console x

Clear all

C# All platforms Web

No exact matches found

Other results based on your search

Console App
A project for creating a command-line application that can run on .NET on Windows, Linux and macOS
C# Linux macOS Windows **Console**

Console App
A project for creating a command-line application that can run on .NET on Windows, Linux and macOS
Visual Basic Linux macOS Windows **Console**

Console App
Run code in a Windows terminal. Prints "Hello World" by default.
C++ Windows **Console**

TypeScript Console Application
A basic TypeScript **Console** application template which can be run with your local node installation.
TypeScript Web

.NET Framework ve .NET CORE

- .NET Framework ve .NET Core , **Microsoft** tarafından geliştirilen iki farklı yazılım geliştirme platformudur.
- .NET Framework, Windows işletim sistemi için tasarlanmış bir platformdur.
- .NET Core ise **Windows, Linux ve macOS** gibi farklı işletim sistemlerinde çalışabilen bir platformdur.
- .NET 5.0 ve öncesi .Net Framework olarak bilinmektedir, ancak **Microsoft .NET 5.0 ve öncesinden desteğini çekmiştir.**

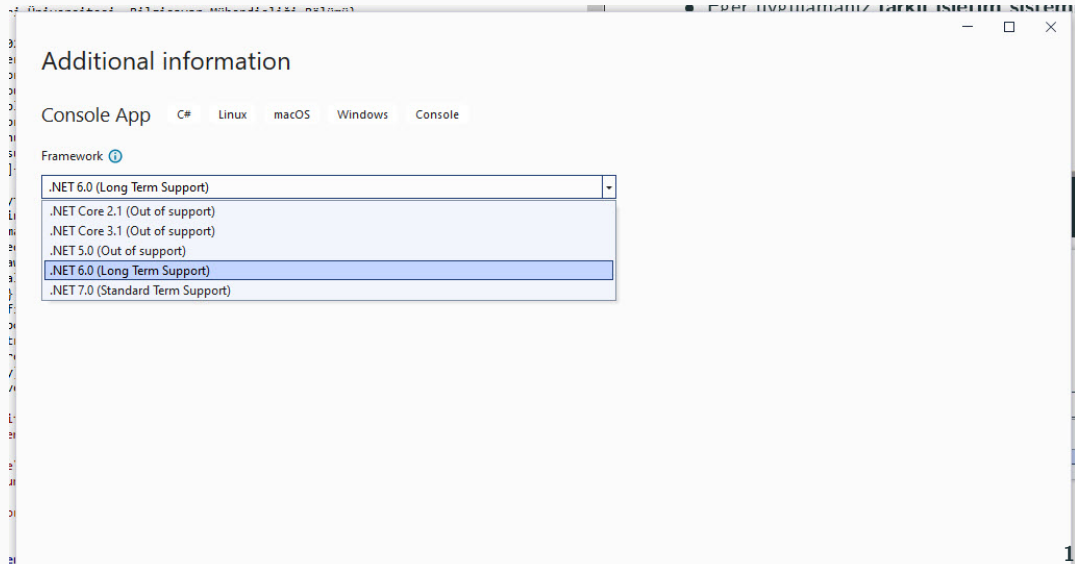
- **.NET Framework yalnızca Windows** işletim sistemini desteklerken, **.NET Core Windows, Linux ve macOS** işletim sistemlerini destekler.
- **.NET Framework açık kaynak kodlu değildir**, ancak **.NET Core açık kaynak kodlu** bir platformdur.
- **Açık kaynak kod (open source)**, yazılımın kaynak kodunun herkes tarafından erişilebilir ve değiştirilebilir olduğu bir yazılım geliştirme modelidir. Bu modelde, yazılım geliştiricileri, yazılımın iç işleyişini anlamak, geliştirmek, düzeltmek ve iyileştirmek için kaynak kodunu inceleyebilirler.

- **.NET Framework, monolitik** bir mimariye sahiptir.
- **Monolitik bir sistem**, tek bir büyük ve bütünsel bir uygulama olarak inşa edilmiş olan bir yazılım sistemidir. Bu sistemde, farklı işlevler ve bileşenler genellikle tek bir uygulama içinde bir araya getirilir ve birbirleriyle sıkı şekilde bağlantılıdır. Sistemin çalışması için bütün bileşenlerin eksiksiz çalışması gerekir.
- **.NET Core ise mikro hizmetler** mimarisine sahiptir.
- **Mikro hizmet mimarisi**, büyük ve karmaşık bir yazılım uygulamasını, küçük ve bağımsız hizmetlere bölerek tasarlamak ve geliştirmek anlamına gelir. Her bir hizmet, belirli bir işlevselliği sunar ve genellikle kendi veritabanıyla birlikte çalışır.

Hangi Platform Seçilmeli

- .NET Core, .NET Framework'e göre daha yüksek performans sunar.
- .NET Core, .NET Framework'e göre daha hızlı ve kolay bir geliştirme süreci sunar.
- Eğer uygulamanız **yalnızca Windows** işletim sistemi için geliştirilecekse, **.NET Framework** uygun bir seçenektir.
- Eğer uygulamanız **farklı işletim sistemlerinde** de çalışacaksa, **.NET Core** daha uygun bir seçenektir.

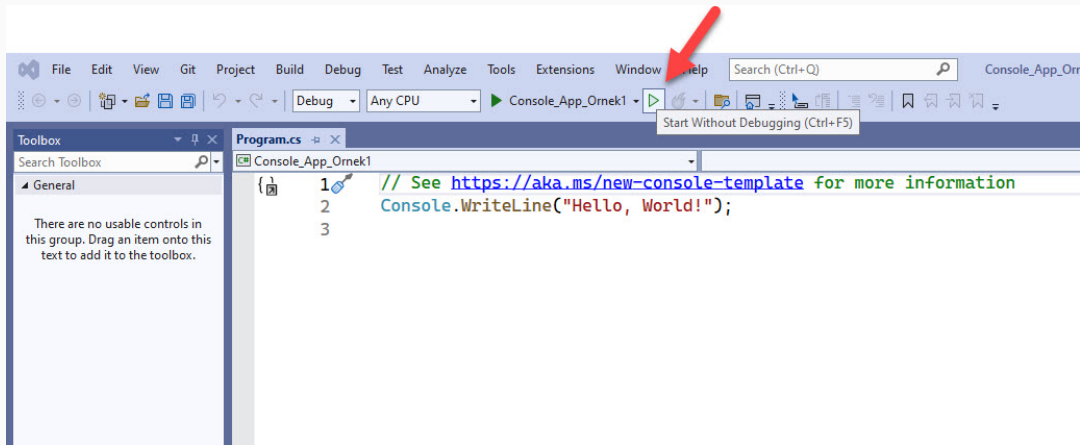
.Net Framework Seçimi



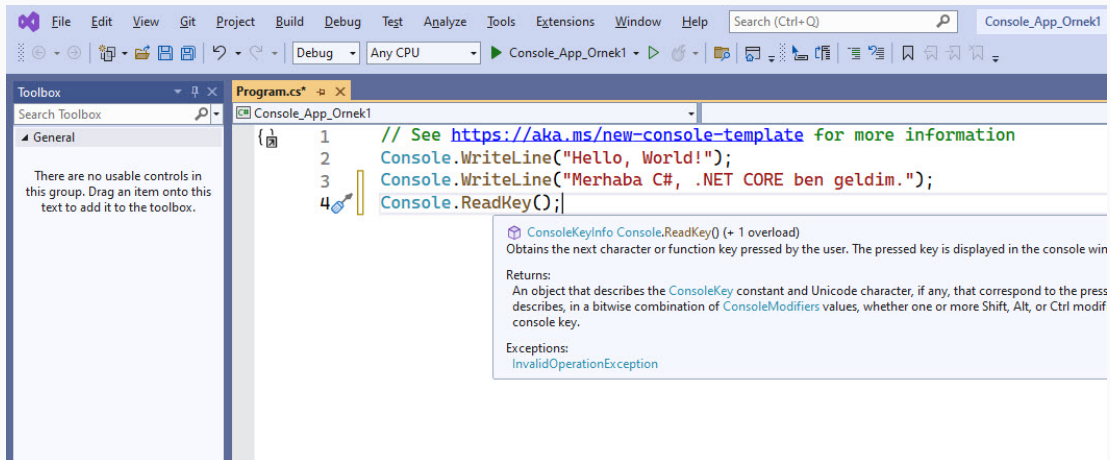
Hangi Platform Seçilmeli

- .NET Core, .NET Framework'e göre daha yüksek performans sunar.
- .NET Core, .NET Framework'e göre daha hızlı ve kolay bir geliştirme süreci sunar.
- Eğer uygulamanız **yalnızca Windows** işletim sistemi için geliştirilecekse, **.NET Framework** uygun bir seçenektir.
- Eğer uygulamanız **farklı işletim sistemlerinde** de çalışacaksa, **.NET Core** daha uygun bir seçenektir.

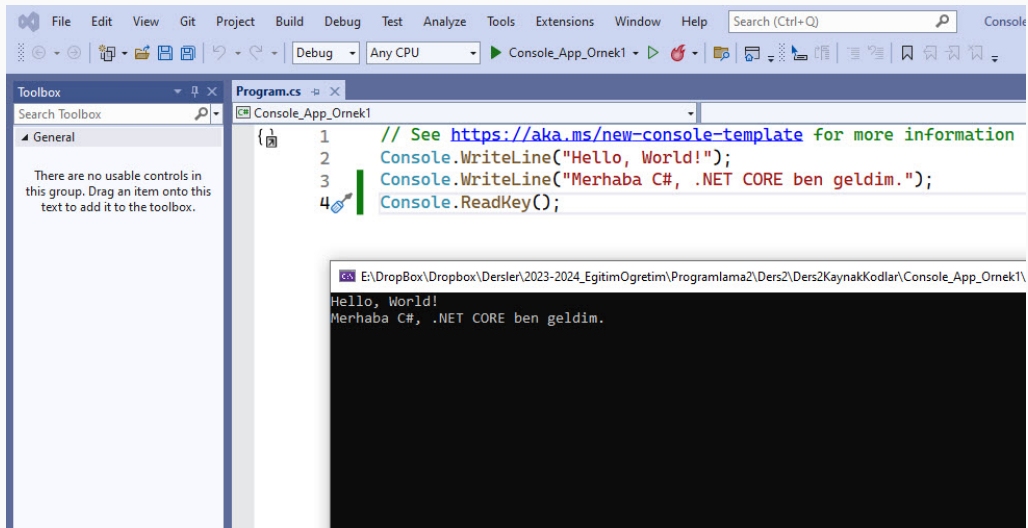
Consele App



Tab Otomatik Doldurma



İlk Console App

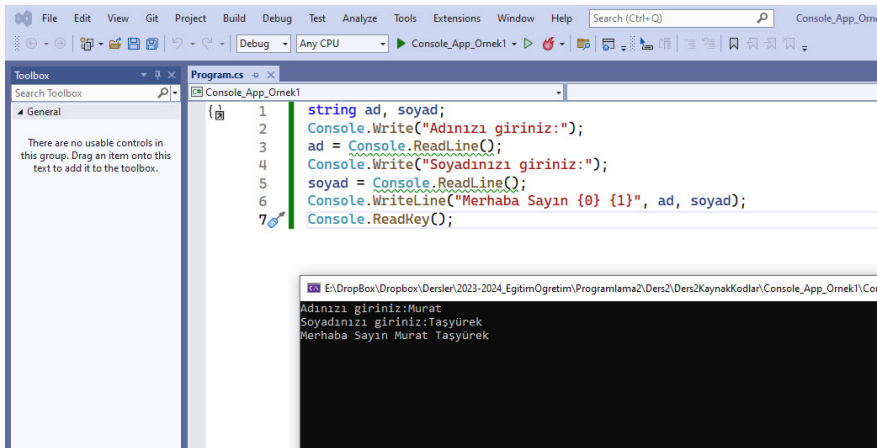


C# programlama dilinde değişken tanımlamak için aşağıdaki sözdizimini kullanabilirsiniz:

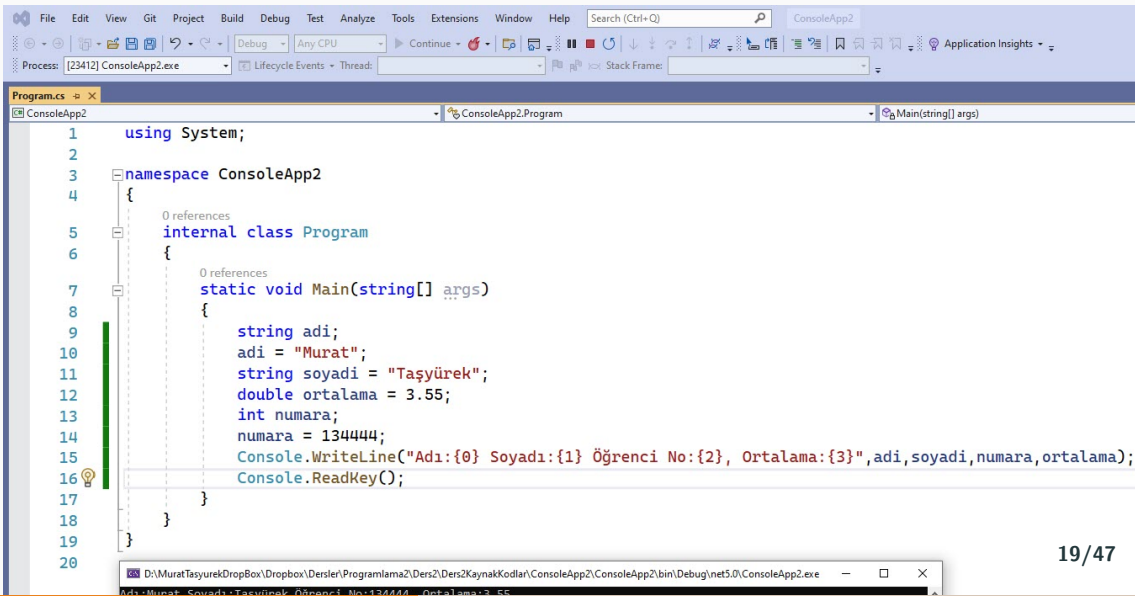
- `veri_tipi degisken_adi;`

Console uygulaması oluşturup değişkenlerimiz tanımlayalım.

Değişkeni Ekran Yazdırma



Değişken Tanımlama



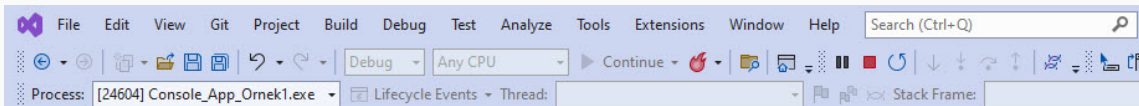
The screenshot shows the Visual Studio IDE with a C# program named `Program.cs` in the `ConsoleApp2` project. The code defines a namespace `ConsoleApp2` containing an internal class `Program`. Inside `Program`, there is a static method `Main` that declares and initializes several variables: `string adi`, `string soyadi`, `double ortalama`, and `int numara`. The `Main` method then uses `Console.WriteLine` to output the values of these variables in a formatted string, followed by `Console.ReadKey()`.

```
1 using System;
2
3 namespace ConsoleApp2
4 {
5     0 references
6     internal class Program
7     {
8         0 references
9         static void Main(string[] args)
10        {
11            string adi;
12            adi = "Murat";
13            string soyadi = "Taşyürek";
14            double ortalama = 3.55;
15            int numara;
16            numara = 134444;
17            Console.WriteLine("Adı:{0} Soyadı:{1} Öğrenci No:{2}, Ortalama:{3}", adi, soyadi, numara, ortalama);
18            Console.ReadKey();
19        }
20    }
```

The console output window at the bottom shows the execution of the program, displaying the output of the `Console.WriteLine` statement:

```
D:\MuratTasyurekDropBox\Dropbox\Dersler\Programlama2\Ders2\Ders2KaynakKodlar\ConsoleApp2\ConsoleApp2\bin\Debug\net5.0\ConsoleApp2.exe
Adı:Murat Soyadı:Taşyürek Öğrenci No:134444 Ortalama:3.55
```

Değişken Tanımlama



Program.cs

Console_App_Ornek1

```
{
1  string ad;
2  ad = "Murat";
3  string soyad = "Taşyürek";
4  double notOrtalamasi = 3.55;
5  int numara;
6  numara = 12456;
7  Console.WriteLine("Adınız:{0}, Soyadınız:{1}, Öğrenci No:{2}, Ortalama:{3}",
8      ad, soyad, numara, notOrtalamasi);
9  Console.ReadKey();
}
```

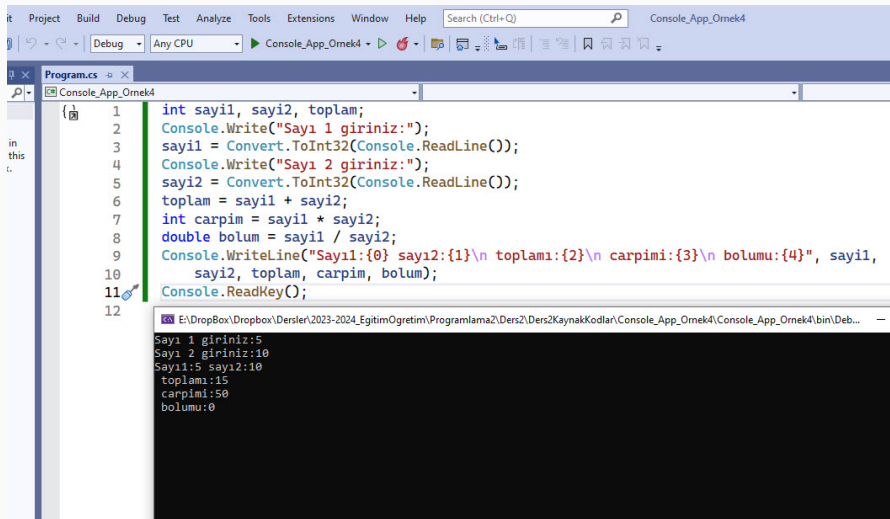
E:\DropBox\Dropbox\Dersler\2023-2024_EgitimOgretim\Programlama2\Ders2\Ders2KaynakKodlar\Console_App_Ornek1\Console_App_Ornek1

Adınız:Murat, Soyadınız:Taşyürek, Öğrenci No:12456, Ortalama:3.55

Kullanıcıdan İnteger Değer Alma

- C# konsole uygulamalarında kullanıcıdan değer almak için `Console.ReadLine()` fonksiyonu kullanılır.
- Ancak bu fonksiyon geriye metin (string) değer dönderir.
- Kullanıcıdan **int**, **double** ve benzeri değerler almak istiyorsanız öncelikle string olarak alırsınız ve daha sonra **convert** işlemi ile dönüşüm yaparsınız.
- Kullanıcıdan iki adet tam sayı değerleri alan bu değerlerin toplamı, bölümü ve çarpımı ekrana yazdıran Console uygulamasını kodlayalım.

Kullanıcıdan Değer Alma



The screenshot shows the Visual Studio IDE with a C# console application named 'Console_App_Ornek4'. The code in 'Program.cs' prompts the user for two numbers, reads them, and then calculates and displays the sum, product, and division. The output window shows the program's execution with the user input '5' and '10', resulting in a sum of 15, a product of 50, and a division result of 0.

```
1 int sayi1, sayi2, toplam;  
2 Console.Write("Sayı 1 giriniz:");  
3 sayi1 = Convert.ToInt32(Console.ReadLine());  
4 Console.Write("Sayı 2 giriniz:");  
5 sayi2 = Convert.ToInt32(Console.ReadLine());  
6 toplam = sayi1 + sayi2;  
7 int carpim = sayi1 * sayi2;  
8 double bolum = sayi1 / sayi2;  
9 Console.WriteLine("Sayı1:{0} sayı2:{1}\n toplamı:{2}\n carpimi:{3}\n bolumu:{4}", sayi1,  
10 sayi2, toplam, carpim, bolum);  
11 Console.ReadKey();  
12
```

Output:

```
Sayı 1 giriniz:5  
Sayı 2 giriniz:10  
Sayı1:5 sayı2:10  
toplamı:15  
carpimi:50  
bolumu:0
```

- Programlama dillerinde baştan sona bütün kodlamanın değil de şartı sağlayan kod satırlarının çalıştırılmasını sağlayan bileşenlere koşul ifadeleri adı verilir.

```
if (true)
{
    // statements executed if boolean-expression is true
}
```

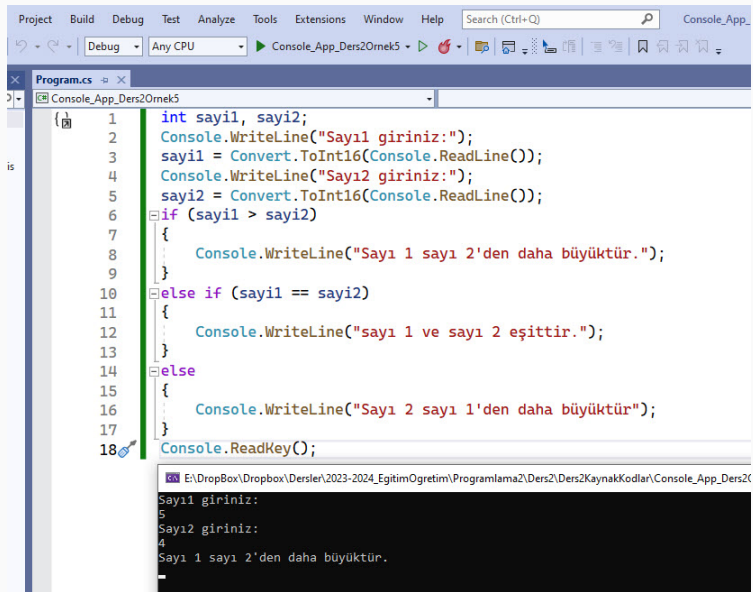
Kullanıcıdan Alınan Değerleri Karşılaştırma

- C# Console uygulamalarında kullanıcıdan iki adet tam sayı değeri alınız.
- Aldığınız bu değerleri **if else** then yapısı ve **and or** yapısı kullanarak karşılaştırınız.

If else

```
if (boolean - expression - 1)
{
    // statements executed if boolean-expression-1 is true
}
else if (boolean - expression - 2)
{
    // statements executed if boolean-expression-2 is true
}
else if (boolean - expression - 3)
{
    // statements executed if boolean-expression-3 is true
}
else
{
    // statements executed if all above expressions are false
}
```

If else Örnek



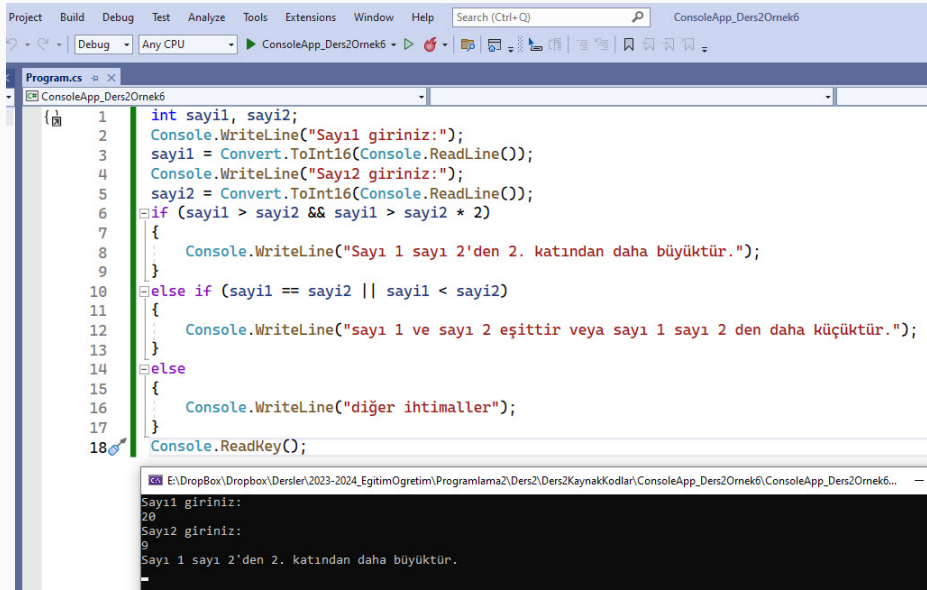
The image shows a Visual Studio IDE with a C# program named `Program.cs` in a project called `Console_App_Ders2Ornek5`. The code implements an if-else statement to compare two integers, `sayi1` and `sayi2`, entered by the user. The program prompts the user to enter two numbers and then prints a message based on the comparison: if `sayi1` is greater than `sayi2`, it prints "Sayı 1 sayı 2'den daha büyüktür."; if they are equal, it prints "sayı 1 ve sayı 2 eşittir."; otherwise, it prints "Sayı 2 sayı 1'den daha büyüktür.". The console output at the bottom shows the program's execution with the inputs 5 and 4, resulting in the message "Sayı 1 sayı 2'den daha büyüktür.".

```
1  int sayi1, sayi2;
2  Console.WriteLine("Sayı1 giriniz:");
3  sayi1 = Convert.ToInt16(Console.ReadLine());
4  Console.WriteLine("Sayı2 giriniz:");
5  sayi2 = Convert.ToInt16(Console.ReadLine());
6  if (sayi1 > sayi2)
7  {
8      Console.WriteLine("Sayı 1 sayı 2'den daha büyüktür.");
9  }
10 else if (sayi1 == sayi2)
11 {
12     Console.WriteLine("sayı 1 ve sayı 2 eşittir.");
13 }
14 else
15 {
16     Console.WriteLine("Sayı 2 sayı 1'den daha büyüktür.");
17 }
18 Console.ReadKey();
```

Console Output:

```
E:\DropBox\Dropbox\Dersler\2023-2024_EgitimOgretim\Programlama2\Ders2\Ders2KaynakKodlar\Console_App_Ders2C
Sayı1 giriniz:
5
Sayı2 giriniz:
4
Sayı 1 sayı 2'den daha büyüktür.
```

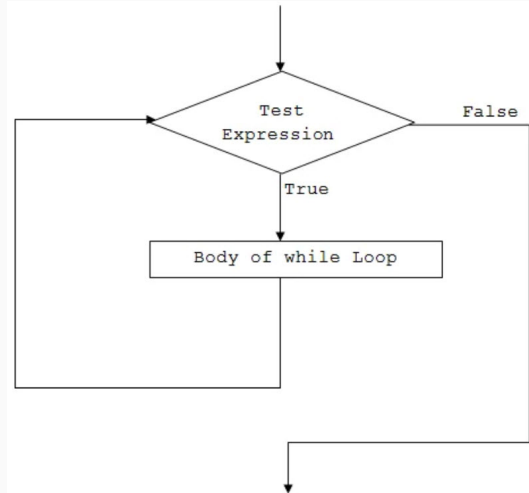
if else (and - or) örnek



```
Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) ConsoleApp_Ders2Ornek6
Debug Any CPU ConsoleApp_Ders2Ornek6
Program.cs
ConsoleApp_Ders2Ornek6
1 int sayi1, sayi2;
2 Console.WriteLine("Sayı1 giriniz:");
3 sayi1 = Convert.ToInt16(Console.ReadLine());
4 Console.WriteLine("Sayı2 giriniz:");
5 sayi2 = Convert.ToInt16(Console.ReadLine());
6 if (sayi1 > sayi2 && sayi1 > sayi2 * 2)
7 {
8     Console.WriteLine("Sayı 1 sayı 2'den 2. katından daha büyüktür.");
9 }
10 else if (sayi1 == sayi2 || sayi1 < sayi2)
11 {
12     Console.WriteLine("sayı 1 ve sayı 2 eşittir veya sayı 1 sayı 2 den daha küçüktür.");
13 }
14 else
15 {
16     Console.WriteLine("diğer ihtimaller");
17 }
18 Console.ReadKey();
E:\DropBox\Dropbox\Dersler\2023-2024_EgitimOgretim\Programlama2\Ders2\Ders2KaynakKodlar\ConsoleApp_Ders2Ornek6\ConsoleApp_Ders2Ornek6...
Sayı1 giriniz:
20
Sayı2 giriniz:
9
Sayı 1 sayı 2'den 2. katından daha büyüktür.
```

- Programlama dillerinde, bir döngü (loop), belirli bir işlemin belirli bir koşul sağlandığı sürece tekrarlanmasını sağlayan bir kontrol yapısıdır.
- Döngüler, belirli bir işi tekrar etmek ve tekrar eden işlemleri otomatikleştirmek için kullanılır.
- Genellikle programlama dillerinde kullanılan temel döngü türleri: **For Döngüsü**, **While Döngüsü**, **Do-While Döngüsü**

While



While Döngüsü İle Çift Sayısı

- C# Console uygulamalarında kullanıcıdan bir adet tam sayı değeri alınız.
- Sıfırdan aldığınız bu değerlere kadar olan çift sayıları **While** döngüsü kullanarak ekrana yazdırınız.
- Aynı örneği **do-while** döngüsü kullanarak kodlayınız.
- Aynı örneği **For** döngüsü kullanarak kodlayınız.

While Örnek

0 references

```
internal class Program
```

```
{
```

0 references

```
static void Main(string[] args)
```

```
{
```

```
    int sayi1;
```

```
    Console.Write("1. Sayı : ");
```

```
    sayi1 = Convert.ToInt32(Console.ReadLine());
```

```
    int count= 0;
```

```
    while (count < sayi1)
```

```
    {
```

```
        if (count % 2 == 0 && count >0)
```

```
        {
```

```
            Console.WriteLine("{0} sayısı çift sayıdır.", count);
```

```
        }
```

```
        count++;
```

```
    }
```

```
    Console.ReadKey();
```

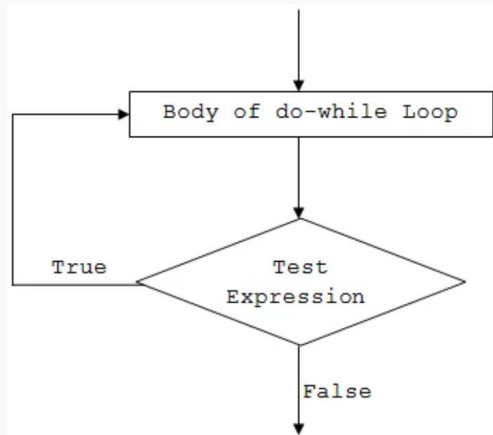
```
}
```

```
}
```

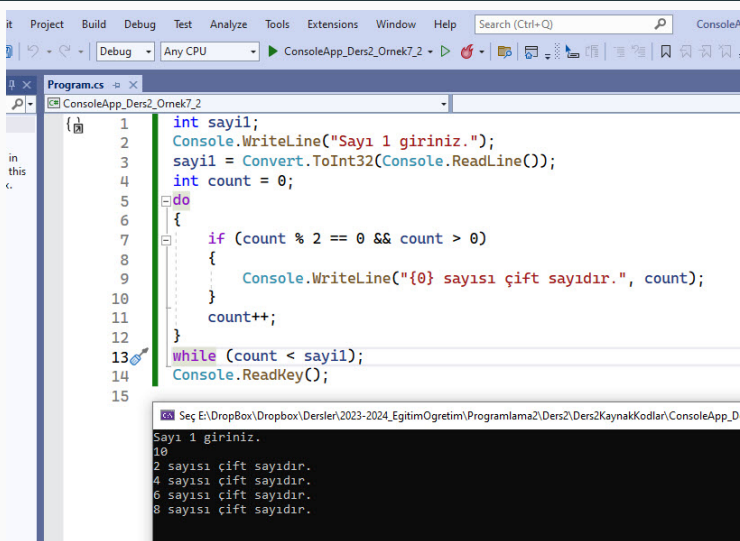
D:\MuratTasyurekDropBox\Dropbox\Dersler\Programlama2\Ders2\Ders2KaynakKodlar\ConsoleApp2\ConsoleApp2\bin\Debug\net5.0\Con

```
1. Sayı : 15
2 sayısı çift sayıdır.
4 sayısı çift sayıdır.
6 sayısı çift sayıdır.
8 sayısı çift sayıdır.
10 sayısı çift sayıdır.
```

Do While



Do While Örnek



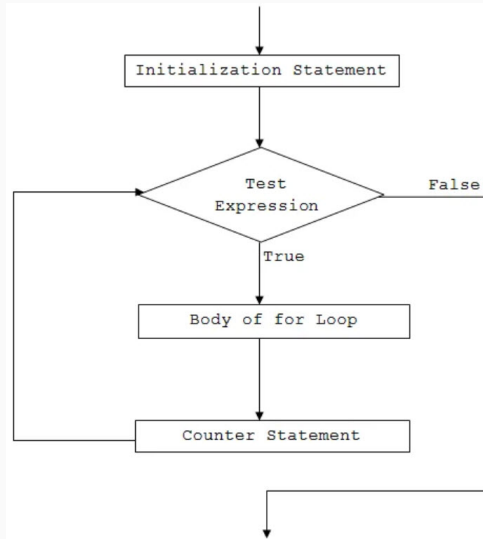
```
1  int sayi1;  
2  Console.WriteLine("Sayı 1 giriniz.");  
3  sayi1 = Convert.ToInt32(Console.ReadLine());  
4  int count = 0;  
5  do  
6  {  
7      if (count % 2 == 0 && count > 0)  
8      {  
9          Console.WriteLine("{0} sayısay çift sayıdır.", count);  
10     }  
11     count++;  
12 }  
13 while (count < sayi1);  
14 Console.ReadKey();  
15
```

Seç E:\DropBox\Dropbox\Dersler\2023-2024_EgitimOgretim\Programlama2\Ders2\Ders2KaynakKodlar\ConsoleApp_D

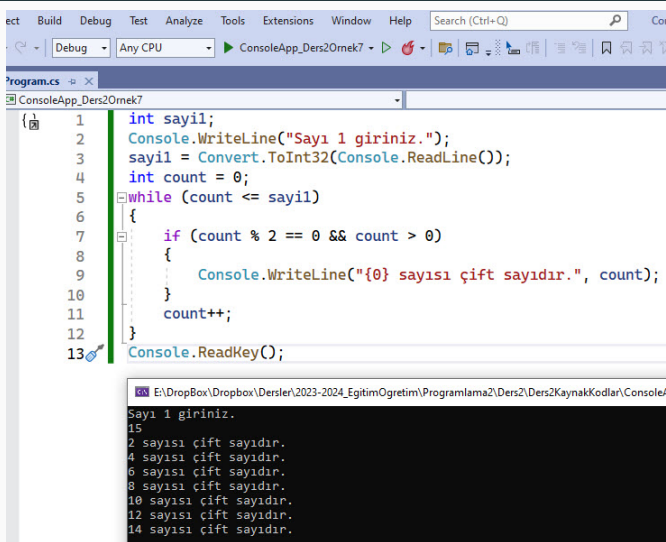
Sayı 1 giriniz.
10
2 sayısay çift sayıdır.
4 sayısay çift sayıdır.
6 sayısay çift sayıdır.
8 sayısay çift sayıdır.

```
for (initialization; condition; iterator)
{
    // body of for loop
}
```

For Döngüsü



For Döngüsü



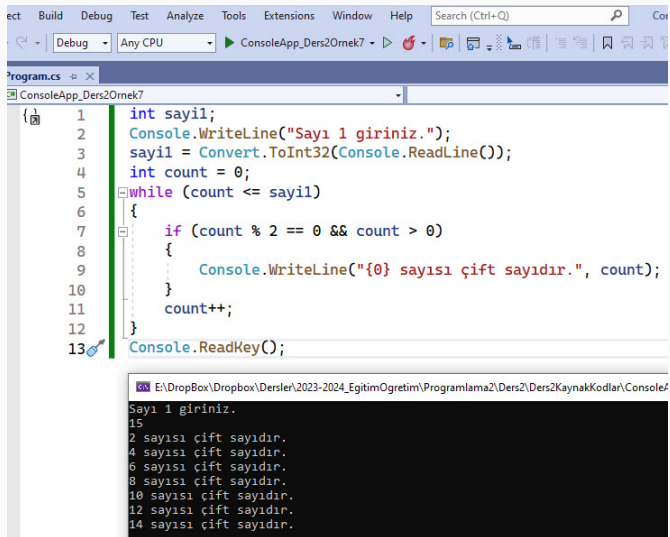
The screenshot shows a C# program in Visual Studio. The code is as follows:

```
1 int sayi1;  
2 Console.WriteLine("Sayı 1 giriniz.");  
3 sayi1 = Convert.ToInt32(Console.ReadLine());  
4 int count = 0;  
5 while (count <= sayi1)  
6 {  
7     if (count % 2 == 0 && count > 0)  
8     {  
9         Console.WriteLine("{0} sayısı çift sayıdır.", count);  
10    }  
11    count++;  
12 }  
13 Console.ReadKey();
```

The output window shows the following results:

```
Sayı 1 giriniz.  
15  
2 sayısı çift sayıdır.  
4 sayısı çift sayıdır.  
6 sayısı çift sayıdır.  
8 sayısı çift sayıdır.  
10 sayısı çift sayıdır.  
12 sayısı çift sayıdır.  
14 sayısı çift sayıdır.
```

For Döngüsü Örnek



The screenshot shows a C# program in Visual Studio. The code is as follows:

```
1 int sayi1;  
2 Console.WriteLine("Sayı 1 giriniz.");  
3 sayi1 = Convert.ToInt32(Console.ReadLine());  
4 int count = 0;  
5 while (count <= sayi1)  
6 {  
7     if (count % 2 == 0 && count > 0)  
8     {  
9         Console.WriteLine("{0} sayısı çift sayıdır.", count);  
10    }  
11    count++;  
12 }  
13 Console.ReadKey();
```

The console output shows the program's execution:

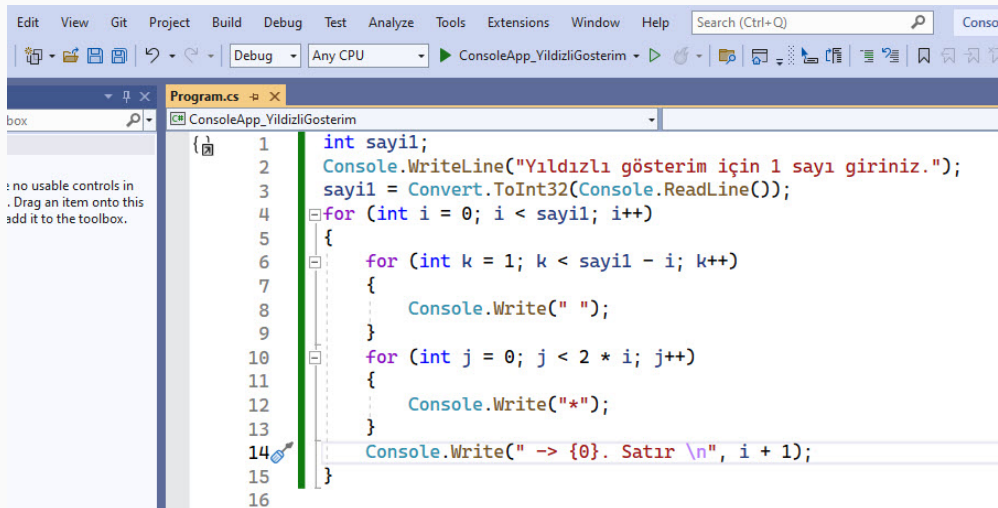
```
Sayı 1 giriniz.  
15  
2 sayısı çift sayıdır.  
4 sayısı çift sayıdır.  
6 sayısı çift sayıdır.  
8 sayısı çift sayıdır.  
10 sayısı çift sayıdır.  
12 sayısı çift sayıdır.  
14 sayısı çift sayıdır.
```

For Döngüsü İçin Çıktı

Görüntüde bulunan çıktıyı sağlayan for döngüsünü veya döngülerini kodlayalım

```
C:\ D:\MuratTasyurekDropBox\Dropbox\Dersler\Programlama2\Ders2\Ders2K
1. Sayı : 10
      *      -> 1. Satır
     ***     -> 2. Satır
    ***** -> 3. Satır
   ********* -> 4. Satır
  *********** -> 5. Satır
 *****     -> 6. Satır
*****       -> 7. Satır
*****       -> 8. Satır
*****       -> 9. Satır
*****       -> 10. Satır
```

İç İçe For Döngüsü

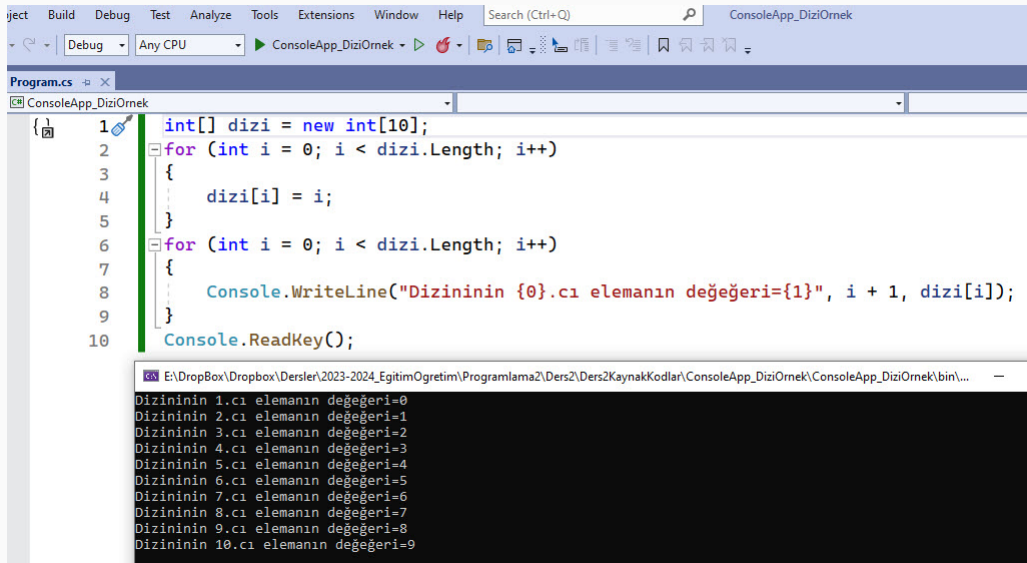


- **Dizi** (array), aynı türden birden fazla veriyi tek bir değişken altında depolamak için kullanılan veri yapısıdır.
- **Diziler**, genellikle sabit boyutta ve sıralı bir koleksiyon olarak düşünülürler.
- Her bir eleman, bir indeks numarası kullanılarak erişilebilir.
- Değişken tanımlarken aşağıda gösterildiği gibi köşeli parantez kullanılır.

```
string[] cars;
```

- integer değerleri alan ve 10 tane ögesi olan bir dizi tanımlayalım.
- Dizi'nin ögeleri sırasıyla 1'den 10'a kadar olan sayılar olsun.
- Daha sonra bu değerleri ekrana yazdıralım.
- İlgili örneği C#'ta kodlayalım.

Dizi Tanımlama



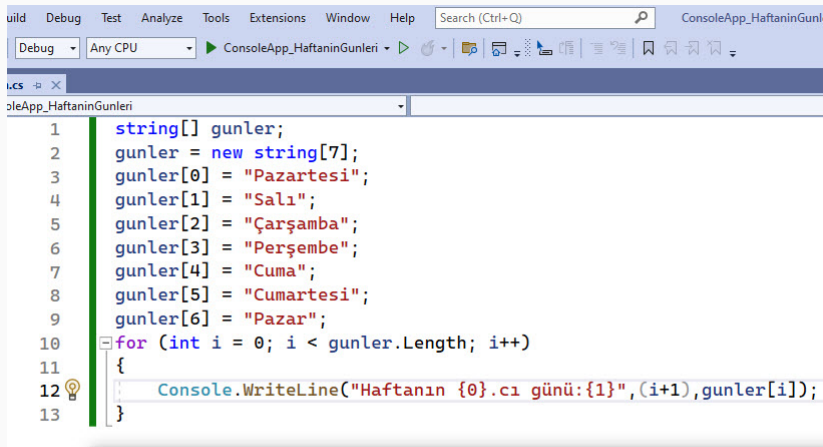
The screenshot shows a Visual Studio IDE with a C# console application named 'ConsoleApp_DiziOrnek'. The code in 'Program.cs' defines an integer array 'dizi' of size 10, fills it with values from 0 to 9, and prints each element with its index. The output in the console window shows the array contents: 'Dizininin 1.cı elemanın değeri=0' through 'Dizininin 10.cı elemanın değeri=9'.

```
1 int[] dizi = new int[10];  
2 for (int i = 0; i < dizi.Length; i++)  
3 {  
4     dizi[i] = i;  
5 }  
6 for (int i = 0; i < dizi.Length; i++)  
7 {  
8     Console.WriteLine("Dizininin {0}.cı elemanın değeri={1}", i + 1, dizi[i]);  
9 }  
10 Console.ReadKey();
```

E:\DropBox\Dropbox\Dersler\2023-2024_EgitimOgretim\Programlama2\Ders2\Ders2KaynakKodlar\ConsoleApp_DiziOrnek\ConsoleApp_DiziOrnek\bin\...
Dizininin 1.cı elemanın değeri=0
Dizininin 2.cı elemanın değeri=1
Dizininin 3.cı elemanın değeri=2
Dizininin 4.cı elemanın değeri=3
Dizininin 5.cı elemanın değeri=4
Dizininin 6.cı elemanın değeri=5
Dizininin 7.cı elemanın değeri=6
Dizininin 8.cı elemanın değeri=7
Dizininin 9.cı elemanın değeri=8
Dizininin 10.cı elemanın değeri=9

- Dizinler string veya başka türdede de olabilir.
- Haftanın gün bilgilerini tutan bir dizi tanımlayalım.
- Günleri dizide tutalım ve değerlerini yazdıralım.

Dizi Tanımlama



The screenshot shows the Visual Studio IDE with a C# file named `ConsoleApp_HaftaninGunleri`. The code defines a string array `gunler` containing the days of the week in Turkish. A `for` loop iterates over the array, printing each day to the console. The code is as follows:

```
1 string[] gunler;  
2 gunler = new string[7];  
3 gunler[0] = "Pazartesi";  
4 gunler[1] = "Salı";  
5 gunler[2] = "Çarşamba";  
6 gunler[3] = "Perşembe";  
7 gunler[4] = "Cuma";  
8 gunler[5] = "Cumartesi";  
9 gunler[6] = "Pazar";  
10 for (int i = 0; i < gunler.Length; i++)  
11 {  
12     Console.WriteLine("Haftanın {0}.cı günü:{1}", (i+1), gunler[i]);  
13 }
```

Microsoft Visual Studio Debug Console

```
Haftanın 1.cı günü:Pazartesi  
Haftanın 2.cı günü:Salı  
Haftanın 3.cı günü:Çarşamba  
Haftanın 4.cı günü:Perşembe  
Haftanın 5.cı günü:Cuma  
Haftanın 6.cı günü:Cumartesi  
Haftanın 7.cı günü:Pazar
```

- Aynı örneği **foreach** döngüsü kullanarak yapalım.
- C#'da foreach döngüsü, bir koleksiyonun (dizi, liste, vb.) her bir elemanını işlemek için kullanılan bir döngü yapısıdır.
- foreach döngüsü, belirtilen koleksiyonun her elemanını sırayla alır ve belirtilen işlemi her eleman üzerinde uygular.
- Bu döngü, genellikle koleksiyon üzerinde gezinmek ve her elemanla işlem yapmak için tercih edilir.

Foreach Kullanım

```
1 string[] gunler = { "Pazartesi", "Salı", "Çarşamba", "Perşembe", "Cuma", "Cumartesi", "Pazar" };  
2 foreach(string s in gunler)  
3 {  
4     Console.WriteLine("{0}", s);  
5 }  
6  
7 Console.WriteLine("\n\n");  
8  
9 int[] rakamlar = { 0, 1, 3, 4, 5, 6 };  
10 foreach (int i in rakamlar)  
11 {  
12     Console.WriteLine("{0}", i);  
13 }
```

Microsoft Visual Studio Debug Console

Pazartesi
Salı
Çarşamba
Perşembe
Cuma
Cumartesi
Pazar

0

1

3

4

5

No IS

- Bir müşteri birlikte çalıştığı yazılım firmasından kişilere özel bir takvim geliştirmesini istemektedir.
- Öncelikle kişiselleştirilmiş takvim sisteminde kullanıcılardan adı, soyadı, numarası, cinsiyeti ve doğum tarihi bilgileri alınmaktadır.
- Takvim sisteminde bayanlar için hafta 5 günden oluşmakta, baylar için ise hafta 8 günden oluşmaktadır.
- 18 yaşından gün almış olanlar (doğum tarihinden ay gün hesaplamalı) için aylar 5 hafta diğerleri için ise 6 hafta olmaktadır.
- Daha sonra kullanıcılar dinamik olarak x gün sonra bir işinin olduğunu girecektir. İlgili yazılım Sayın " ... isimli ... soy isimli .. bey/hanım (cinsiyet bilgisine göre yazmalı) ... yaşındasınız. .. ayının ... gününde işiniz bulunmaktadır" bilgisini vermelidir.
- ay isimleri: 1. ay, 2. ay 3. ay...., gün isimleri: 1. gün, 2. gün
- ilgili yazılımı C# dilinde **.NET Core** ve **.NET framework** olarak **ayrı ayrı** Console App formatında kodlayıp kaynak kodları ve çıktıları ile birlikte 6 Mart 07.59'a kadar gönderilenler **8+2** puan, vizeye kadar gönderenler **8** puan (**kayubmprogramlama1@gmail.com** adresine gönderilecek).