İlk programınız, yazılım hayatınız boyunca en zorlandığınız program olarak kalacaktır.

İlk Program

String Manipülasyonu

Oğuzhan Karagüzel 14.12.2024

İçindekiler

ŞEKİLLER DİZİNİ

İLK PROGRAM VE STRİNG MANİPÜLASYONU

İlk programımızı yazacağız ve şimdiye kadar öğrendiklerimizi kullanarak basit string manipülasyonu yapacağız. Bu dersin sonunda, kendi yazdığınız kodun çalıştığını görmeyi ve bunun keyfini çıkarmayı umuyorum.

Girdi Almak ve String Manipülasyonu

Öncelikle, bugün basit bir şekilde kullanıcıdan isim bilgisi alacağız ve bu ismi string manipülasyonu yaparak ekrana farklı biçimlerde yazdıracağız. Bu süreçte hem toplama operatörünü (`+`) hem de string interpolasyonu (“$"" ifadesiyle kullanılır) yöntemlerini kullanacağız.

Programın Amacı ve Detaylar

Programımız kullanıcıdan bir isim alıp bu ismi farklı biçimlerde ekrana yazdıracak ve son olarak bu ismi tersine çevirecek. Ters çevirmeyi yaparken, `Array.Reverse()` metodunu kullanarak string'in içerisindeki harflerin sıralamasını tersine çevireceğiz. Bu metodun kullanımıyla birlikte basit algoritmalar üretmeyi öğreneceğiz.

Metotları henüz tam olarak bilmediğinizin farkındayım. Ancak bir program yazabilmek için şu an metod kullanmak zorundasınız. Bundan dolayı verdiğim metotların tam olarak ne yaptıklarını ve nasıl kullandıklarını detaylı bir biçimde anlatacağım. Bu detaylara dikkat edin. Yazdığınız programı anlamanızı kolaylaştıracaktır.

Ayrıca eğer bir yazılımcı olacaksanız, ne işe yaradığını bilmediğiniz kod bloklarını sıklıkla kullanacaksınız. Bundan dolayı yazdığınız programdaki her bir satırın ne işe yaradığını bilmemeniz oldukça normal bir durum.

Kodlama Aşaması

Aşağıda adım adım ilerleyerek ilk kodumuzu yazacağız. Visual Studio’da ilk olarak bir console uygulaması projesi açın. İlk derslerden bunu nasıl yapacağınızı biliyorsunuz. .Net Framework olması gerektiğini unutmayınız.

Adım 1: Girdi Alma

Kullanıcıdan veri almak için `Console.ReadLine()` metodunu kullanacağız. Bu metod, kullanıcıdan girdi almak için kullanılan en temel metotlardandır.

```
static void Main(string[] args)
{
    Console.WriteLine("Lütfen Adınızı Giriniz");
    string isim = Console.ReadLine();
    Console.Read();
}
```

Şekil 1 Girdi Alma

- `Console.WriteLine("Lütfen Adınızı Giriniz")`: Bu satır kullanıcıya "Adınızı girin:" mesajını ekrana yazdırır. Böylece kullanıcı, ne yapması gerektiğini görebilir.
- `string isim = Console.ReadLine()`: Bu satır, kullanıcının klavyeden girdiği ismi alır ve `isim` adı verilen bir string değişkende saklar.
- Console.Read(); konsol uyuglamasını ayakta tutmaya yarayan standart kod bluğumuz. Bunu daha önceki derslerden öğrendiniz.

Console.WriteLine("Lütfen Adınızı Giriniz"); metodunu artık öğrendiğinizi var sayıyorum. İçine yazdığınız metni konsola yazar. Console.WriteLine() içerisine bir string yazdığınızı artık fark etmişsinizdir. İster metodun dışında bir string tanımlayıp metoda verin, ister doğrudan metodun içerisinde string ifadeyi oluşturun. İkiside aynı kapıya çıkacaktır.

Adım 2: String Manipülasyonu

Kullanıcıdan alınan ismi manipüle ederek çeşitli biçimlerde ekrana yazdıralım. Burada hem `+` hem de `$` işaretini kullanarak farklı yollarla string oluşturmanın örneklerini göreceğiz.

```
static void Main(string[] args)
{
    Console.WriteLine("Lütfen Adınızı Giriniz");
    string isim = Console.ReadLine();
    Console.WriteLine("Merhaba, " + isim + "! Programlama dünyasına hoş geldin.");
    Console.WriteLine($"Bugün seninle string manipülasyonu öğreneceğiz, Hazır mısın {isim}?");
    Console.Read();
}
```

Şekil 2 String Manipülasyonu

- `+` **Operatörü**: `+` operatörü, iki ya da daha fazla string'i birleştirmek için kullanılır. Örneğin, `"Merhaba, " + isim + "!"` kullanıcıdan alınan ismi bir mesajla birleştirir.
- `$` **String Interpolasyonu**: `$` işareti, string interpolasyonu yapmak için kullanılır. String interpolasyonu, string içerisinde değişkenlerin değerlerinin kolayca kullanılmasını sağlar. Örneğin, `$"Merhaba, {isim}!"` ifadesi kullanıcının ismini mesaja ekleyerek daha okunabilir bir kod yazmamıza olanak tanır.

Burada iki yöntemde aslında aynı işi yapmaktadır. Temelde ikisi arasında çok bir fark yoktur. Duruma göre hangisi ihtiyacınızı karşılıyorsa o yöntem kullanılır. Ancak ikisi arasında bir tercih yapabiliyorsanız "$" tercih edin. Bu şekilde kodun okunurluğu oldukça artar.

+ operatöründe her bir string ayrı ayrı yazılarak adeta toplanır. "ahmet" + "sınavdan" + sinavNotu + "aldı" şeklinde parça parça bir sürü string oluşturulur.

$" Ahmet sınavdan { sinavNotu } aldı" şeklinde tek bir string içerisinde işinizi halledebilirsiniz. + operatörünün kullanımını zaten içgüdüsel olarak anlamışsınızdır. $ operatörü ise sadece tırnakların başına eklenerek kullanılır. Ardından değişkenlerinizi süslü parantez {} içerisinde kullanabilirsiniz.

Adım 3: String'i Tersine Çevirme

Bir sonraki adımımızda kullanıcıdan aldığımız ismi tersine çevireceğiz. Bu işlem için öncelikle string'i bir `char` dizisine dönüştüreceğiz ve sonrasında `Array.Reverse()` metodunu kullanacağız.

Array.Reverse() Nedir?

- `Array.Reverse()`, dizilerle ("array") çalışırken sıklıkla kullanılan ve belirli bir dizinin elemanlarını tersine çeviren bir metottur.
- Bu metod, dizideki elemanların önceden belirlenmiş sıralamasının tam tersi olacak şekilde yer değiştirilmesini sağlar. Bu durumda dizideki ilk eleman sona, son eleman ise başa geçer.
- .Net nedir sorusuna bir cevap daha bulmuş oluyorsunuz. Sizin kullanmanız için yazılmış hali hazırda bulunan kodlar. Birini şimdi öğrenmiş oldunuz

. (Nokta) Operatörü ve ToCharArray() metodu

- `.` operatörü, bir nesne üzerinden metot ya da özelliklere erişim sağlamak için kullanılır. Bu operatör, nesnenin ya da verinin hangi özellik ya da fonksiyonlarla ilişkilendirildiğini belirtir. Bu konuyu metodlara geldiğimizde daha iyi anlayacaksınız.
- `ToCharArray()` metodu, bir `string` ifadeyi `char` dizisine dönüştürmek için kullanılır. Bu sayede string içerisindeki her bir karaktere kolayca erişim sağlayabiliriz. Normal şartlar altında string'in aslında bir char[] olduğunu söylemiştik. Ancak bu yaklaşımımız gereği böyledir. Bizler bunu içgüdüsel olarak ta anlayabiliriz. Fakat bilgisayar anlayamaz. Onun için char array char array'dir. String ise string'tir. Bundan dolayı değişkenin türünü bilgisayara söylememiz lazım. ToCharArray() diyerek metnimizi tam olarak bir char[]'e dönüştürmüş olduk.

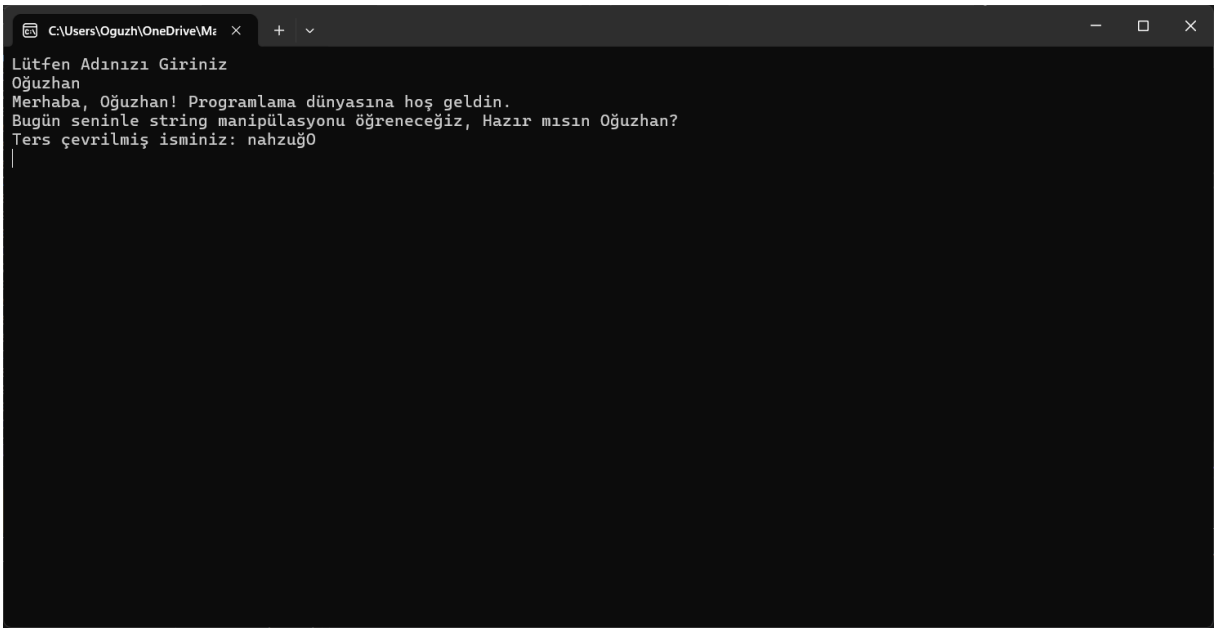
Aşağıda, kullanıcıdan alınan ismi tersine çevirmek için gerekli olan kodu görebilirsiniz:

```
static void Main(string[] args)
{
    Console.WriteLine("Lütfen Adınızı Giriniz");
    string isim = Console.ReadLine();
    Console.WriteLine("Merhaba, " + isim + "! Programlama dünyasına hoş geldin.");
    Console.WriteLine($"Bugün seninle string manipülasyonu öğreneceğiz, Hazır mısın {isim}?");

    char[] karakterler = isim.ToCharArray();
    Array.Reverse(karakterler);
    string tersIsim = new string(karakterler);
    Console.WriteLine($"Ters çevrilmiş isminiz: {tersIsim}");

    Console.Read();
}
```

Şekil 3 İsim Ters Çevirme

- `ToCharArray()`: `isim` değişkenini (`string`) karakterlerden oluşan bir diziye (“char array”) dönüştürür. Bu sayede her bir harfe erişim sağlayabiliriz.
- `Array.Reverse(karakterler)`: `karakterler` dizisinin elemanlarını tersine çevirir.
- - string tersIsim = new string(karakterler): Tersine çevrilen char dizisi tekrar bir string olarak saklanır ve Console.WriteLine() ile ekrana yazdırılır. Hatırlarsanız bir önceki derste yapıcı metoddan bahsetmiştik. Her bir değişkeni new anahtar kelimesi ile oluşturabileceğimizden bahsetmiştik. Ardından gelen metoda da yapıcı metod demiştik. Burada ise yapıcı metodu kullanıyoruz ve o metoda bir char array veriyoruz. Metod char array’i alarak bir string oluşturuyor.

Hadi programımızı çalıştırıp sonucu görelim.



Şekil 4 İsim ters çevirme programı çıktısı.

Gördüğünüz üzere programımız bizden ismimizi alıyor. Gerekli mesajları verdikten sonra ismimizi ters çeviriyor.

YORUM SATIRLARI VE YAZILIM GELİŞTİRME SÜRECİNDEKİ ÖNEMİ

Yorum Satırlarının Tanımı ve Kullanım Amaçları

Yazılım geliştirme sürecinde, yazılan kodu hem anlaşılırlığını artırmak hem de diğer geliştiricilere yol göstermek amacıyla kullanılan metinlere "yorum satırları" denir. Yorum satırları, programlama dillerinde genellikle derlenmeyen ve çalıştırılmayan metin bloklarıdır. Bu, yorumların yalnızca açıklama veya not bırakma amacı taşıdığı anlamına gelir.

Yorum satırlarının başlıca kullanım amaçları şunlardır:

- **Kodun Anlaşılabilirliğini Artırmak:** Kodun ne yaptığını ve neden belirli bir şekilde yazıldığını açıklamak için kullanılır.
- **Bakımı Kolaylaştırmak:** Yıllar sonra bile kodun kolayca anlaşılabilmesi için geliştiricilere rehberlik eder.
- **İşbirliğini Kolaylaştırmak:** Ekip üyelerinin birbirlerinin kodunu anlamalarına yardımcı olur.
- **Hataları Belirlemek ve Önlemek:** Kod üzerinde gelecekte yapılacak değişiklikler veya düzeltmeler için not bırakılır.

Örnek:

// Bu kod, bir kullanıcının yaşını hesaplar.

int yas = 2024 - dogumYili;

Bu örnekte, yorum satırı kodun amacını açıklamaktadır. Bu tür açıklamalar, kodu okuyan bir geliştirici için hızlı bir şekilde anlamayı sağlar.

Yorum Satırı Türleri

Programlama dillerinde birkaç farklı yorum satırı türü bulunmaktadır. C# dilinde bu türler şunlardır:

Tek Satırlık Yorumlar

Tek satırlık yorumlar, `//` ile başlar ve aynı satırdaki metni kapsar.

int yas = 30; // Kullanıcının yaşını saklayan değişken.

Bu kullanım, yalnızca tek bir satırı açıklamak için idealdir.

Çok Satırlı Yorumlar

Birden fazla satırı açıklamak için `/*` ve `*/` arasına alınan yorum blokları kullanılır.

```
/*
Bu metot, kullanıcının tam adını ve yaşını yazdırır.
Parametreler:
 - ad: Kullanıcının adı.
 - yas: Kullanıcının yaşı.
*/
void BilgiYazdir(string ad, int yas)
{
   Console.WriteLine($"Ad: {ad}, Yaş: {yas}");
}
```

Çok satırlı yorumlar, genellikle daha kapsamlı açıklamalar gerektiğinde tercih edilir.

XML Yorumları

`///` ile başlayan yorumlar, kodun belgelendirilmesini sağlar. Özellikle sınıf, metot veya değişkenler hakkında detaylı açıklama eklemek için kullanılır. Bu tür yorumlar, otomatik dokümantasyon oluşturmak için idealdir.

Henüz xml'in ne olduğunu bilmiyorsunuz. Bundan dolayı bu yorum türü size oldukça yabancı gelecektir. Ancak ilerleyen başlıklarda xml'den çok önce metodlara gireceğiz. Metod özetleri yazmak için ise bu yorum satırlarını kullanacağız.

```
/// <summary>
/// Kullanıcının yaşını hesaplar.
/// </summary>
/// <param name="dogumYili">Kullanıcının doğum yılı.</param>
/// <returns>Hesaplanan yaş.</returns>
int YasHesapla(int dogumYili)
{
   return 2024 - dogumYili;
}
```

XML yorumları, Visual Studio gibi IDE'lerde otomatik olarak tamamlanabilir ve dokümantasyon oluşturulabilir.

Yorum Satırlarının Doğru Kullanımı

Yorumlar kodun anlaşılırlığını artırsa da yanlış veya gereksiz yorum kullanımı kodun karmaşık görünmesine neden olabilir. Aşağıda, doğru ve yanlış yorum kullanımına dair bazı ipuçları verilmiştir:

İyi Bir Yorumun Özellikleri

- **Açık ve Net Olmalıdır**: Yorumu okuyanın, kodun amacını hızlı bir şekilde anlayabilmesi gerekir.
- **Kodu Tekrarlamamalıdır:** Kodun kendisinin anlattığı bir şeyi tekrar etmekten kaçınılmalıdır.
- **Amaca Odaklanmalıdır:** Yorum, kodun neden yazıldığını açıklamalıdır, ne yaptığını değil.

Kötü Yorum:

```
// Değişkeni 10 yap
int x = 10;
```

İyi Yorum:

```
// Kullanıcının başlangıç puanı.
int x = 10;
```

Yorumların Sıkça Yapılan Hataları

- **Gereksiz Yorumlar:** Çok açık bir kodu açıklamaya çalışmak gereksizdir.
- **Güncellenmeyen Yorumlar:** Kod değiştiği halde yorumlar güncellenmezse, yanlış bilgi verilebilir.
- **Fazla Uzun Yorumlar:** Yorumun çok uzun olması kodun okunabilirliğini düşürür.

TODO Kullanımı ve Yararları

`TODO` etiketi, yazılım geliştirme sırasında eksik bırakılan veya daha sonra tamamlanması gereken işleri belirtmek için kullanılır. `TODO`, genellikle yorum satırları içinde büyük harflerle yazılır ve birçok IDE tarafından özel olarak tanınır.

```
// TODO: Bu metodun hata durumlarını ele al.
void DosyaOku(string dosyaAdi)
{
    Console.WriteLine("Dosya okunuyor...");
}
```

TODO Etiketinin Avantajları

- **Eksik İşlerin Takibi:** Kodda eksik kalan işlerin kolayca bulunmasını sağlar.
- **IDE Desteği:** Çoğu IDE, `TODO` etiketlerini listeleyerek geliştiricilere rehberlik eder.
- **Proje Yönetimi:** Büyük projelerde, hangi işlerin tamamlanması gerektiğini görsel bir şekilde takip etmeye yardımcı olur.

Uygulamalı Örnek

```
class Program
{
    static void Main()
    {
        // TODO: Kullanıcıdan girdi alınacak bir sistem oluştur.
        Console.WriteLine("Merhaba, dünya!");
        // TODO: Verileri bir dosyaya kaydet.
    }
}
```

Bu tür etiketler, yazılım geliştirme sürecinde planlama ve görev takibini kolaylaştırır.

Sonuç

Yorum satırları, yazılım geliştirme sürecinin olmazsa olmaz bir parçasıdır. Kodun anlaşılabilirliğini artırır, bakımını kolaylaştırır ve işbirliğini destekler. Doğru kullanıldığında, yorumlar yazılımcının en güçlü araçlarından biri haline gelir. Ancak kötü yorum kullanımı kodu karmaşıklaştırabilir ve yanlış yönlendirmelere neden olabilir. `TODO` etiketleriyle işlerinizi düzenleyebilir ve gelecekte yapılacak işleri sistematik bir şekilde takip edebilirsiniz.

Bu bilgilerle artık yorumları etkili bir şekilde kullanabilir ve daha iyi kod yazabilirsiniz.

ÖDEV

Bu dersimizde, kullanıcıdan veri almayı, bu veriyi basit bir şekilde manipüle etmeyi ve son olarak basit bir algoritma ile veriyi tersine çevirmeyi öğrendik. Şimdi, bu bilgileri kullanarak aşağıdaki ödevi yapmanızı istiyorum. Yapay Zeka kullanmak serbest.

Kullanıcıdan iki sayı alın. Ardından bu iki sayının toplamını ekrana yazdıran bir program yazın. Kullanıcıdan alınan değer her zaman string olacaktır. Bundan dolayı bu değeri nasıl tam sayıya çevireceğinizi araştırıp bulmanız gerekiyor.

Ardından programınızı adım adım yorum satırları ile açıklayın.

Bu dersle ilgili her türlü sorunuz için bana ulaşabilirsiniz. Şimdiden hepinize kolay gelsin ve başarılar dilerim!

TEST

İsim ters çevirme programı tam olarak 9 satır koddan oluşmaktadır. Bir önceki dersin ödevi olan string kaçış elemanlarını hatırlayın. Eğer o ödevi yapmadıysanız ilk olarak o ödevi tamamlamanız gerekmektedir.

String kaçış elemanlarını kullanarak doğrudan 1 satır azaltabilirsiniz. Ardından program akışını düzenleyerek 1 satır daha azaltabilirsiniz.

Her azalttığınız satır için 25 puan alacaksınız. Bu testte 50 puan almanız yeterlidir.

Bu programı 4 satıra indirip 125 puan almak mümkün.

İpuçları:

- Bir önceki dersten string inşa (constructor) metodunu hatırlamaya çalışın. Gerekirse dokümanı tekrar edin.
- Bir char[] değişkenini elemanlarını terse çevirmenin başka yollarını arayın. Bu sayede programınızı 4 satıra düşürebilirsiniz.
- Yapay zeka kullanmak ya da araştırma yapmak serbest. Ancak her bir işlemi yorum satırı ile açıklama koşulu ile.
- $(f \circ g)(x)=f(g(x))$ fonksiyon birleşimini hatırlayı! Aynı kurallar olduğu gibi burada geçerlidir.

Satırları azaltmak yazılımcılar açısından çok hoş karşılanan bir durum değildir. (Gereksiz görülenler hariç) Kodun okunaklılığını azaltır. Ancak bu alıştırma algoritma yeteneğinizi arttıracaktır. Bundan dolayı bu alıştırmayı yapın. Nasıl yapacağınızı öğrenin. Ancak yapmaktan kaçının.