Variables

Değişkenler bilgisayar belleğinde veri saklamak için kullanılır. C#’da değişkenler aşağıda gösterildiği gibi tanımlanmaktadır.

Example

//Syntax : type name = value

string firstName = "John";

* Tek satırda birden fazla değişken tanımlanabilir.
* Değişkenler tanımlandıkları scope dışında kullanılamazlar.
* Metod içinde tanımlanan değişkenler erişim belirteçlerine sahip olamaz.
* Değişkenler static anahtar kelimesi ile tanımlanamaz ancak static alan(field) tanımlanabilmektedir.
* Değişkenler readonly anahtar kelimesi ile tanımlanamaz ancak readonly alan tanımlanabilmektedir.
* Değişken tanımlarken tiplere verilen takma adlar yerine struct ve class isimleri(Common Type System’deki karşılıkları) kullanılabilir.

Example

System.String name = "John Doe";

System.Byte age = 30;

System.Int32 salary = 4000;

* struct ve class’lardan nesne oluşturularak değişken tanımlanabilir.

Example

System.String city = new System.String("İstanbul".ToCharArray());

System.Int32 number = new System.Int32();

number = 34;

* Değişkenlere varsayılan değerleri aktarılırken default anahtar kelimesi kullanılabilmektedir. [C# 7.1]

(Constant Variables)

Sabit değişkenler const anahtar kelimesi ile tanımlanmaktadır.

Example

const double PI = 3.14;

* Sabit değişkenlere ilk değerleri verilmelidir.
* Sabitler tanımlandıktan sonra değer değişikliği yapılamaz.
* class içinde tanımlanan sabitler statik alanlar gibi class’ın örneği alınmadan kullanılır.

Example

class Class1

{

public const double PI = 3.14;

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine(Class1.PI);

Console.ReadLine();

}

}

(var Keyword)

Example

var year = 2022;

* var anahtar kelimesi ile field tanımlanamaz.
* İlk değer ataması zorunludur.
* Her türde bilgi saklayabilirler ancak değer ataması yapıldıktan sonra değişkenin tipi değiştirilemez.
* null değer tutamazlar.
* Değişkenin içerdiği bilginin türü GetType() metodu ile öğrenilebilmektedir.
* C#’da tamsayıların int virgüllü sayıların ise double tipinde olduğu varsayılmaktadır. Bu nedenle var keyword ile tanımlanan değişkenlere değer aktarmadan önce tip dönüşümü yapabilirsiniz.

(dynamic Keyword)

Example

dynamic year = 2022;

* İlk değer ataması zorunlu değildir.
* Her türde bilgi saklayabilirler.

(Fields)

class ve struct içinde tanımlanan değişkenlere **alan(field)** adı verilir.

Example

class Employee

{

public string firstName;

public int salary;

}

* Erişim belirteçlerine sahip olabilirler. Erişim belirteci olmayan alanların private oldukları varsayılır.
* Başlangıç değeri verilebilir.
* var anahtar kelimesi ile alan tanımlanamaz.
* dynamic anahtar kelimesi ile alan tanımlanabilir.
* static anahtar kelimesi ile alan tanımlanabilir.
* const anahtar kelimesi ile alan tanımlanabilir.

(readonly Fields)

Bazen programın yazımı sırasında değil de nesne oluşturulurken içeriği belirlenebilen sabitlere gerek duyulur. Bu durumda readonly alanlar kullanılır.

* Değer ataması yalnızca class’ın yapıcı metodu içinden yapılabilir.
* Başlangıç değeri verilebilir.
* public olarak tanımlanmışsalar class dışından değerleri okunabilir.

|  |
| --- |
| **NOT**  İsimlendirme kuralları :   * İsimler rakam ile başlayamaz ancak rakam içerebilirler. * \_ dışındaki özel karakterler kullanılamaz. * Türkçe’ye özel karakterlerin kullanılması tavsiye edilmez. * Anahtar kelimeler, deyimler ve rezerve edilmiş kelimeler isim olarak kullanılamaz. |