



## SAKARYA MESLEK YÜKSEKOKULU

### BİLGİSAYAR PROGRAMCILIĞI PROGRAMI

#### Algoritma ve Programlamaya Giriş Ders Notları

#### Ders 8: Visual Studio Değişken ve Veritürü Kavramı

*Visual Studio'da Değişken Kavramı*

*Visual Studio'da Veritürü Kavramı*

*Kullanılan Veritürleri*

*Hazırlayan*

**Öğr. Gör. İsmail ÖYLEK**

*Sakarya*

*2022*

## Visual Studio Değişken ve Veritürü Kavramı

**Değişken:** Değişkenler ihtiyaç duyulan bilgilerin saklanmasını ve işlenmesini sağlar. Tanımlanan bir değişken için program yürütüldüğünde bellekte bir alan ayrılır. Bilindiği üzere RAM bellek, programlarda işlenecek olan bilgilerin geçici olarak depolanmasını sağlayan birimdir.

Değişken tanımlama işlemi üç adet parametrenin bir araya gelmesi ile gerçekleştirilir. Bu parametreler aşağıda anlatılmıştır.

**1-Tür:** Saklanacak olan verinin hangi türde ve hangi aralıkta olacağını belirleyen kavramdır. Örneğin; 0-255 arasında pozitif tam sayı değerleri için **byte** türünün kullanılması uygun olur.

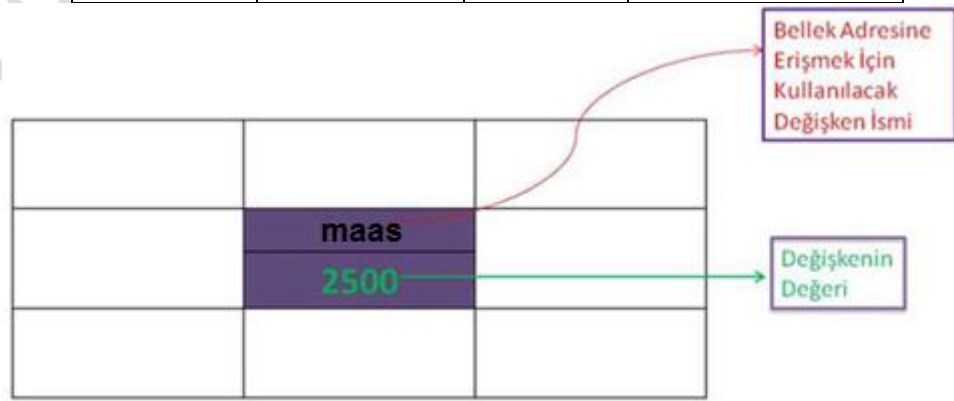
**2-İsim:** Verinin bellekte saklanacağı bölümün etiketi yani ismidir denilebilir. Örneğin; bir işyerinde her bir çalışanın odasının kapısı üzerinde etiketi mevcut ise aradığımız görevliye bu etiket isimlerinden kolaylıkla ulaşabiliriz. Benzer şekilde bellekteki her verinin bir adresi bulunur. Bu adresleri kullanarak belleğe veri yazabilir veya veriyi çağırıp kullanabiliriz.

**3- Değer:** Tür ve isim belirlendikten sonra bellekte veri için uygun bir alan ayrılır. Bellekte ayrılan bu alana daha sonra saklanacak olan bilgi yazılır. Değer verme işlemi tanımlama yapılan komut satırında gerçekleştirilebileceği gibi programın akışı esnasında sonradan da yapılabilir.

Örnek olarak; **sbyte** -128 ile +127 arasında değer alan, bellekte 1 byte alan kaplayan tam sayı değişken türüdür. String ise 2 milyar karaktere kadar değer alabilir. String değişken bellekte **karakter sayısı x 2 byte** yer kaplar.

Tablo 1. Örnek değer atama işlemi

Tür	Değişken Adı	Atam Op.	Değer
sbyte	sicaklik	=	-5;
string	tcno	=	"11122233347";
int	maas	=	2500;



## Değişkenler neden farklı türlerde tanımlanır?

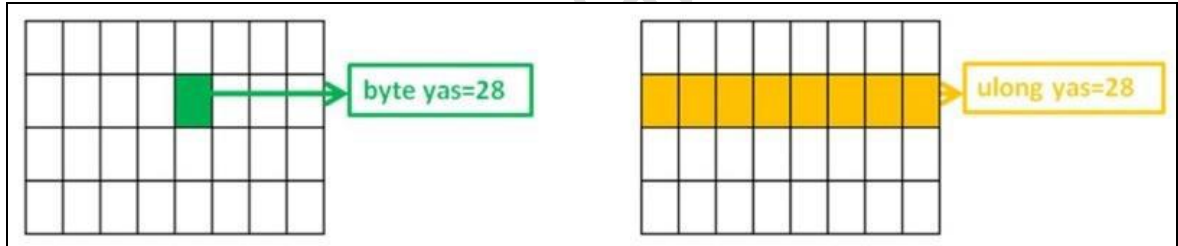
Programlama mimarisinin tasarlanması esnasında farklı veritürlerinin kullanılmasındaki amaç belleğin **en verimli şekilde kullanımını sağlamaktır**. RAM bellek (*Random Access Memory*) bilgisayarın en önemli donanım birimlerinden biridir ve bellek alanı kapasitesi düşük olmasına rağmen fiyatı oldukça yüksektir. Veri türlerinin sayısı daha az olsaydı belleği istediğimiz şekilde dizayn etme şansımız da olmazdı. Bu durumda bellek kapasitesinin gereksiz yere işgal edilme ihtimali ortaya çıkardı.

Bir örnekle açıklanacak olursa;

**byte**: 0-255 arasındaki pozitif tam sayı değerlerini alır. Bellekte 1 Byte alan kaplar.

**ulong**: 0-18.446.744.073.709.551.615 arasındaki tam sayı değerlerini alır. Bellekte 8 Byte alan kaplar.

Yazılacak olan programda bir kişinin yaşı üzerinde işlem yapılacak ise yaş verisini saklamak için byte değişken türü yeterli olacaktır. Aynı zamanda int, short ve ushort değişken türlerini de kullanmak mümkündür. Ancak byte yerine ulong değişken türünü kullanmak bellekteki 7 byte boyutundaki alanın gereksiz yere rezerve edilmesine sebep olacaktır. Bu durum aşağıdaki bellek benzetimi ile görsel olarak da ifade edilebilir. Aşağıda yas=28 verisini saklayan byte ve ulong veri türleri için bellekte ne kadar yer tahsis edildiği görülmektedir.



Şekil 1. 28 sayısının byte ve ulong kullanımına göre farkları

## Değişken Türleri

Visual Studio Console programlama diline ait değişken türleri (veri tipleri) sayısal, metinsel (*string*) ve mantıksal (*boolean*) olmak üzere üç ana grupta değerlendirilmektedir.

### 1. Sayısal Değişken Türleri

Sayısal veri türleri **byte, sbyte, short, ushort, int, uint, long, ulong, float, decimal** ve **double** olmak üzere **11 adettir**. Listedeki fark edilebileceği üzere **sbyte** değişken türü **byte** değişken türünün başına **s** harfi eklenerek oluşmuştur. **ushort, uint** ve **ulong** değişken türleri sırasıyla short, int ve long değişken türlerinin önüne **u** harf eklenerek oluşmuştur.

**S (Signed)**: Türkçe'de **işaretli** anlamına gelir. Bilindiği üzere matematiksel işlemlerde pozitif tamsayıların önüne işaret koyulmaz ancak negatif tamsayıların önüne **-** işareti koyulur.

## Algoritma ve Programlamaya Giriş Ders Notları

Dolayısıyla başına **s** eklenen **sbyte** değişken türü pozitif değerler ile birlikte **negatif** değerleri de tutabilmektedir.

**U (Unsigned):** Türkçe’de işaretsiz anlamına gelir. U ile başlayan ushort, uint, ulong gibi değişken türleri yalnızca pozitif tam sayı değerlerini tutabilir.

Bellekte saklanacak olan sayısal verinin negatif ya da pozitif olup olmayacağı durumuna ve değer aralığına göre tür seçimi yapılmalıdır. Bazı sayısal tipler yalnızca pozitif değerler almakta iken bazıları hem negatif hem pozitif değerler alabilir. Yine bazı sayısal tipler yalnızca tam sayı değerleri almakta iken bazıları ondalıklı değerleri de saklayabilir. Bu hususlar dikkate alınarak veri saklama ihtiyacını karşılayan ancak bellekte de en az yer kaplayan bir sayısal veri tipi seçilmelidir.

### Tam Sayı Değişken Tipleri

Veri Türü	Sayı Aralığı	Bellekte Kapladığı Alan
byte	0 ile 255 arasında	1 byte
sbyte	-128 ile 127 arasında	1 byte
short	-32768 ile 32767 arasında	2 byte
ushort	0 ile 65535 arasında	2 byte
int	-2.147.483.648 ile 2.147.483.647 arasında	4 byte
uint	0 ile 4.294.967.295 arasında	4 byte
long	-9.223.372.036.854.775.808 ile 9.223.372.036.854.775.807 arasında	8 byte
ulong	0 ile 18.446.744.073.709.551.615 arasında değer alabilir.	8 byte

### Virgüllü Sayı Değişken Tipleri

Ondalık bir deyişle virgüllü, reel veri tipleri **float**, **double** ve **decimal** veri tipi olmak üzere 3 adettir.

**float:**  $\pm 1.5 \times 10^{-45}$  ile  $\pm 3.4 \times 10^{38}$  arasında değer alabilir. Basamak hassasiyeti 7’dir. Bellekte 4 byte yer kaplar. (Yazılan sayının sonuna *f/F* yazılmalıdır. 3.12f)

**double:**  $\pm 5.0 \times 10^{-324}$  ile  $\pm 1.7 \times 10^{308}$  arasında değer alabilir. Basamak hassasiyeti 16’dır. Bellekte 8 byte yer kaplar.

**decimal:** Decimal ile tanımlanan bir değişken 29 basamaklı bir sayıyı tutabilir. Bellekte 16 byte yer kaplar. (Yazılan sayının sonuna *m/M* yazılmalıdır. 3.12M)

## Algoritma ve Programlamaya Giriş Ders Notları

**NOT-1:** Bir sayısal değişkene değer aktarımı yapılırken **=** atama operatöründen sonra veri yalın halde yazılır. Metinsel türlerde olduğu gibi veri tek tırnak veya çift tırnak işaretleri arasına yazılmaz. Aşağıda bu hususa ilişkin örnek tanımlamalar görülmektedir.

\* **int** agirlik = 83;                      \* **double** maas = 2580.75;                      \* **sbyte** sicaklik = -8;

**NOT-2:** Aşağıdaki örneklerde olduğu gibi **float** değişken türünde tanımlanan bir değişkene verilen değer sonuna büyük veya küçük **f**, **F** harfi eklenmelidir. Aynı zamanda **decimal** değişken türünde de değer sonuna büyük veya küçük **m**, **M** harfi eklenmelidir. Aksi takdirde programın çalışması esnasında hata oluşacaktır.

```
float agirlik = 5.75f;  
float maas = 2740.68F;  
decimal agirlik = 84540.988M;
```

## 2. Metinsel Değişken Türleri

Metinsel değişken türleri **string** ve **char** değişkenler olmak üzere ikiye ayrılır.

**string:** Bir karakter boyutunu geçen metinsel veriler için kullanılan string veri türü 2 milyar karaktere kadar değer saklayabilen ve her bir karakter için bellekte 2 byte yer kaplayan veri tipidir. Üzerinde sayısal işlem yapılmayacak olan *TC Kimlik No*, *Posta Kodu* gibi rakamsal veriler için ve *Ad*, *Soyad* gibi metinsel veriler için kullanılabileceği gibi özgeçmiş benzeri uzun metinler için de kullanılabilir.

String bir değişkene değer verilirken veri çift tırnak içerisine yazılmalıdır. Aksi takdirde ilgili kod satırında hata oluşur.

```
string tcno = "11122233312";  
string ozgecmis = "1982 yılında Sakarya'nın ilinin Taraklı ilçesinde  
dünyaya geldi. İlkokulu Kurtuluş İlköğretim Okulu'nda tamamladı.  
Ortaokulu 21 Haziran Ortaokulu'nda okudu. Daha sonra Konya Meram Fen  
Lisesi'ni kazanarak büyük bir başarıya imza attı. Lisans eğitimini İTÜ  
Bilgisayar Mühendisliği'nde başarı ile tamamladı.";
```

**char:** **Character** kelimesinin ilk dört harfinden oluşur. İsminden de anlaşılacağı üzere tek bir karakterlik veriyi saklayabilir.

Bellekte 2 byte boyutunda bir yer kaplar. Bir soruya verilecek olumlu ve olumsuz karakter sonuçlarını saklamak için kullanılabilir. Örneğin; *Evli için: E*, *Bekâr için: B* veya *Evet için: E*, *Hayır için: H* gibi klavyeden girilebilecek tek karakterlik veriler için kullanılabilir. Benzer şekilde bir menüden seçenek seçiminde de char veri tipinden faydalanılabilir. Örneğin; **1-Toplama**, **2-Çıkarma**, **3-Bölme**, **4-Çarpma** işlemi gibi menü seçiminde kullanılabilir. Char veri tipine değer aktarımı gerçekleştirilirken veri tek tırnak içerisine yazılmalıdır.

```
char cinsiyet = 'E';  
char menu = '2';
```

### 3. Mantıksal Değişken Türü

Visual Studio Programlama ortamında mantıksal değişken türü olarak **bool** tipi kullanılır. İngilizce **boolean** kelimesinin ilk dört harfinden oluşmaktadır. Bu veri tipi yalnızca *true* (doğru) ve *false* (yanlış) değeri alabilir ve bellekte 1 byte yer kaplar. İki seçenekten ibaret olan bir veri üzerinde işlem yapılması istenildiğinde kullanılabilir.

```
bool sonuc = true;  
bool cevap = false;
```

#### Değişken Tanımlama Kuralları

- Değişken tanımlanmadan önce hangi türden veri içereceğine karar verilerek, belleğin verimli şekilde kullanılması sağlanır.
- Bellekte saklanacak olan verinin sayısal mı yoksa metinsel bir veri mi olduğuna karar verilmelidir. Üzerinde matematiksel işlem veya mantıksal kıyaslama işlemi yapılması gereken *ders notu*, *maaş* gibi veriler için **sayısal** türler kullanılır. Ancak *Adres*, *Ad*, *Soyad* gibi veriler için **metinsel** tür kullanılır. *TC Kimlik Numarası*, *Posta Kodu* gibi bir verinin yalnızca rakamlardan oluşuyor olması, o verinin mutlaka sayısal türlerden birinde saklanacağı anlamına gelmez. Burada dikkat edeceğimiz husus üzerinden matematiksel işlem yapıp yapılmayacağı ile ilgilidir. Örneğin; TC kimlik numarası rakamlardan ibarettir ancak bu veri üzerinden matematiksel işlem veya mantıksal kıyaslama yapılmayacağı için sayısal veri türleri veya metinsel veri türlerinden birisi ile tanımlanması mümkündür.
- Tanımlama yapılırken önce tür, daha sonra boşluk karakterinden sonra değişken ismi yazılır. Eğer değişkene tanımlandığı satırda değer aktarılacaksa = operatörü, ardından uygun bir değer yazılır. Değer aktarılması işlemi programın akışı sırasında her hangi bir komut satırı ile doğrudan veya klavyeden alma yöntemiyle de yapılabilir.

#### Örnek:

```
byte sicaklik;  
string tcno = "22255566678";  
int stokadeti = 486;
```



#### Değişkene İsim Verilmesi

- Bellekte tutulacak değere göre tür belirlendikten sonra isim tanımlaması yapılmalıdır.
- Değişkene verilecek olan isim ile saklanacak veri birbiriyle ilişkili olmalıdır. Bu şekildeki bir kullanım programın anlaşılabilirliğini kolaylaştırır. Anlaşılır isimlerin kullanılmaması ilerleyen zamanlarda oluşabilecek hataların çözümünü ve yapılacak düzenlemeleri zorlaştırır.
- Örneğin; maaş verisi için **double m=2560** yerine **double maaş=2560** şeklinde tanımlama yapılması daha doğru olur.

## Algoritma ve Programlamaya Giriş Ders Notları

---

**NOT:** C# programlama dili, Türkçe karakter desteklemesine rağmen, isimlendirme yapılırken Türkçe'ye özgü karakterlerin yerine İngilizce karakterlerin kullanılması daha yerinde olacaktır. Her ne kadar C# programlama dili Türkçe karakter desteği sunuyor olsa da bazı programlama dilleri buna izin vermez. Programlama sektörüne yeni girenlerin bu tavsiyeye uymalarında fayda vardır. Çünkü ilerleyen zamanlarda farklı programlama dillerine yönelme söz konusu olabilir.

SUBÜ - Bilgisayar Programcılığı

**Kaynak:**

- 1- Örneklerle Algoritma ve C# Programlama, E. ARI, Seçkin Yay. Ankara, 2015
- 2- Algoritmalar, S. Uzunköprü, Kodlab Yay. 2017
- 3- <https://bidb.itu.edu.tr/sevir-defteri/blog/2013/09/06/c-'-ta-de%C4%9Fer-ve-referans-tipleri>
- 4- <https://avys.omu.edu.tr/storage/app/public/emre.dunder/132938/2.%20Hafta.pdf>