# Değişken

Yazılım geliştirmenin temelini bilgi oluşturur. Bilgisayardaki herşey bilgidir. Bilgiler okunur, değerlendirilir ve yazılır. Tüm mesele bundan ibarettir. Bilgilere göre uygulama belli bir yol izler ve bilgiler doğrultusunda sonuçlar verir.

Evet, tüm işimiz bilgilerimizi yönetmektir. Şimdi aklınıza gelecek ilk soruyu tahmin ediyorum, mutlaka soracaksınızdır “hangi bilgiler? ne bilgisi?”

Bu çok güzel bir soru ve bu soruyu yanıtlamak oldukça güç. Bu yüzden bir bilgisayar programının nasıl çalıştığını anlamanız gerekiyor.

## Bir Bilgisayar programı nedir?

Bunu bir örnekle anlatmaya çalışacağım. Evden çıkıcaksınız ve eşiniz elinize bir kağıt parçası sıkıştırır ve bunda dönüşte alışverişte almanız gereken şeyler yazılıdır.

Fakat siz kapıdan çıkarken henüz işin yolunu tutuyorsunuz. Evet, bu yolda ilk adımınızı attınız ve üzerinizi giyinip, ayakkabılarınızı da giyerek, kapıdan dışarı çıktınız.

Sonra merdivenlerden basamak basamak dış kapıya indiniz ve kapıya yönelerek açtınız ve binadan çıktınız.

Bitmedi! Arabanıza doğru yol aldınız ve birkaç adım sonra arabaya vardınız. Anahtarınızı cepten çıkararak arabanın kapısını açıp, direksiyon tarafına geçtiniz. Sonra arabayı çalıştırıp yola çıktınız.

İşe varana kadar birkaç ampülde durmak zorunda kaldınız, bazı ara sokaklardan geçmek durumunda oldunuz ve bunun için en kısa yol hangisi ise onu kararlaştırıp o yoldan gittiniz.

İşinize vardınız, işinizi yine titizlikle adım adım, tane tane yaptınız. Sonra işten çıktınız ve yine arabanıza yöneldiniz. Sonra yakındaki marketi tespit edip o tarafa doğru yola çıktınız.

Markete varınca eşinizin size vermiş olduğu kağıdı çıkarıp sırayla Limon, Yumurta, Tomates, Fasulye, Şekertozu, Biberon, Çöp Poşeti gibi gerekenler nelerse aldınız ve sonra eve geçtiniz.

Farkındaysanız, sizi bıktırmamak için daha çok ayrıntıya girmek istemedim. Ama burada anlattığım hikaye örneğin sizin bugünlük programınızdı.

Bilgisayar programlarıda aslında böyle adımlardan oluşuyor. Mesela fareyi ekranda gezdiriyorsunuz, sizin gözünüzde ekranda bulunan küçük bir işaret ekran üzerinde hareket ediyor. Bir pencereden bir başka pencereye varıyor, bir menüye tıklıyor, açılan menü içinde bir öğeye iniyor ve ona tıklıyor.

Bu çok sırdan gibi gözüken olayın ardında onu gerçekleştirmek için bir çok işlem yapılması gerekiyor. Mesela fare simgesinin eski pozisyondan silinip yeni pozisyona çizilmesi gibi. Ondan önce farenin yeni pozisyonunun tespiti gibi. Hangi fare tuşuna basıldığını tespit etmek gibi.

Bir program kısacası bir çok işlemin ardı ardına gerçekleşmesi sonucunda ortaya çıkan çözümdür.

İşlemlerden söz ediyoruz ya, bu işlemlerin gerçekleşebilmesi için bilgiye ihtiyaç var.

# Bilgi nedir?

Bilgisayar adına layık aslında sadece sayısal işlemler yapmaktadır. Yani söz ettiğimiz bilgi tamamen rakamlardan oluşur. Fakat sizin karşınıza bu rakamlar zaman zaman bir mektup, bir makale, bir video, bir müzik, bir resim, bir tablo, bir uçak bileti rezervasyonu, bir bankacılık işlemi gibi farklı şekillerde çıkıyor.

Kod geliştirmek ise işte bu tür sonuçları gerçekleştirmenin sanatıdır. Bunun için bilgileri değerlendiririz. Mesela “Merhaba” sizce bilgisayar dilinde nedir? Hiç düşündünüz mü?

Bilgisayara göre “Merhaba” aslında bir dizi rakamdır:

| M | e | r | h | b | a | |
|---|---|---|---|---|---|---|
| 77 | 101 | 114 | 104 | 97 | 98 | 97 |

Ekranda gördüğünüz göz kamaştırıcı renkler ekran işlemcisi tarafından ekrana yansıtılan ışık kırma dereceleridir. Bu dereceleri tarif eden rakam ise kırmızı, yeşil ve maviyi temsil edip her biri için 0 ile 255 arasında bir değer içeren RGB-modelinde bir rakamdır. Mesela siyahın karşılığı 0 iken beyazın sayısal değeri ise 16777215.

Örnek:

| RGB / KYM | 0 0 0 | > | 0 | > | Siyah |
|---|---|---|---|---|---|
| RGB / KYM | 255 255 255 | > | 16777215 | > | Beyaz |

# Gelelim asıl meseleye

Peki değişken, dizilim ve işaret nedir? Bunların her birini tek tek tanıyacağız. İlk olarak hepsinin temelini oluşturan Veri konseptini öğrenelim.

# Veri konsepti nedir?

İnsanoğlu makinadan daha akıllı olsada, sadece rakamlarla resim çizip, beste yapamaz. İnsana daha somut ve işini kolaylaştırıcı olanaklar gerek.

Bunun için 50inci yıllardan bu yana yazılım dilleri geliştirilirken birçok yöntem denenmiş ve belli bir veri konsepti netleşmiş ve benimsenmiştir.

Makina ile insan arasında tercümanlık yapan yazılım dilleri insanın anlayabileceği, fakat bilgisayarın dilinede mümkün oldukça yakın bir yapıya sahip bu konsepti otomatik olarak gerçekleştirmektedir ve böylece yazılımcı asıl rakamsal değerlerle uğraşmak zorunda kalmamaktadır.

Bilgisayarın anladığı en ufak bilgi “Elektrik akımı var (1)” veya “Elektrik akımı yok (0)” bilgisidir. Bu en ufak bilginin adı Bit (Binary Information Target), Türkçe karşılığı ile “İkili Bilgi Hedefi” veya Adresi. Aslında basit bir şalterdir.

Bu yetersiz olduğu için Bilgisayar işlemcileri Bitlerle tek tek değil, Bitlerden oluşan akım zincirleriyle çalışırlar. Bilgisayarın en az bilgi olarak kabul ettiği Bit zincirinin adı Bayttır. Bir Bayt 8 adet Bitten oluşur. Bit zincirleri sağdan sola doğru ikili sistemin üzerinden hesaplanır.

Örnek:

| | Bit 4 | Bit 3 | Bit 2 | Bit 1 | | |
|---|---|---|---|---|---|---|
| Model: | 8 | 4 | 2 | 1 | | |
| | 0 | 0 | 0 | 1 | = | 1 |
| | 0 | 0 | 1 | 0 | = | 2 |
| | 0 | 1 | 0 | 0 | = | 4 |
| | 1 | 0 | 0 | 0 | = | 8 |
| | 0 | 0 | 1 | 1 | = | 2 + 1 = 3 |
| | 0 | 1 | 1 | 0 | = | 4 + 2 = 6 |
| | 1 | 1 | 0 | 0 | = | 8 + 4 = 12 |

Görüldüğü gibi bir Bit zinciri sayesinde farklı değerlere ulaşılabiliniyor. Modern İşlemcilerin tanıdığı zincirler şöyledir:

| | |
|---|---|
| Bayt (Byte) | 8 Bitten oluşur |
| ÇiftBayt (Word) | 16 Bitten oluşur |
| Dört Bayt (DWord) | 32 Bitten oluşur |
| Sekiz Bayt (QWord) | 64 Bitten oluşur |

Bir program geliştirirken her amaçladığın bir bilgi için belli bir hafıza kapasitesine ihtiyacın olacaktır. Bilgisayarın hafızası insan hafızası gibi sınırsız değildir. Bu yüzden kullanmak istediğin kapasiteyi çok iyi hesaplaman gerekmektedir.

Bu işi kolaylaştırmak için yazılım dilleri Bit zinclerine ve Bit zinclerinden oluşan dizilimlere isimler vermiştir. Bu isimler aynı zamanda Veri türleridir.

# Basitin Veri Türleri

Basit profesyonel bir yazılım dili olarak bazı temel veri türlerini sunar ve bunun dışında yeni veri türleri oluşturabilir. Bu bölümde temel veri türlerini tanıyacağız:

| **Veri Türü** | **Bit_Adeti** | **Minimum** | **Maksimum** | **Amacı** |
|---|---|---|---|---|
| YarıBayt | 8 | -128 | +127 | En düşük sayısal değerleri karşılamak içindir |
| Bayt | 8 | 0 | 255 | En düşük, fakat pozitif olan sayısal değerleri karşılamak içindir |

| Veri Türü | Bit_Adeti | Minimum | Maksimum | Amacı |
|---|---|---|---|---|
| Ascii | 8 | 0 | 255 | Ascii Tablosunun karakterlerini desteklemek içindir (Özellikle C fonksiyonlarıyla veri alışverişinde bulunurken gereklidir) |
| ÇiftBayt | 16 | 0 | 65535 | Geniş bir banda sahip olup, sadece pozitif sayısal değerleri karşılamak içindir |
| SafKüçükRakam | 16 | 0 | 65535 | Geniş bir banda sahip olup, sadece pozitif sayısal değerleri karşılamak içindir |
| KüçükRakam | 16 | -32768 | +32767 | Geniş bir banda sahip olup, hem pozitif, hemde negatif sayısal değerleri karşılamak içindir |
| Renk | 32 | 0 | 4294967295 | KYM (RGB) ve KYMŞ (RGBA) renklerini karşılamak içindir |
| Saat | 32 | 0 | 4294967295 | Saat, Dakika ve Saniyeyi karşılamak içindir |
| Rakam | 32 | -2147483648 | +2147483647 | Standart sayısal değerleri karşılamak içindir |
| SafRakam | 32 | 0 | 4294967295 | Standart geniş bir banda sahip olup, sadece pozitif sayısal değerleri karşılamak içindir |
| EsnekRakam | 32 | -2147483648 | +2147483647 | 32 Bit derlemede Rakam ile aynıdır |
| SafEsnekRakam | 32 | 0 | 4294967295 | 32 Bit derlemede SafRakam ile aynıdır |
| EsnekRakam | 64 | -9223372036854775808 | +9223372036854775807 | 64 Bit derlemede GenişRakam ile aynıdır |
| SafEsnekRakam | 64 | 0 | 18446744073709551615 | 64 Bit derlemede SafGenişRakam ile aynıdır |

| Veri Türü | Bit_Adeti | Minimum | Maksimum | Amacı |
|---|---|---|---|---|
| GenişRakam | 64 | -9223372036854775808 | +9223372036854775807 | Çok geniş bir banda sahip olup, hem pozitif, hemde negatif sayısal değerleri karşılamak içindir |
| SafGenişRakam | 64 | 0 | 18446744073709551615 | Çok geniş bir banda sahip olup, sadece pozitif sayısal değerleri karşılamak içindir |
| KüçükVirgül | 32 | -1175494e-38 | +3402823e+38 | Normal boyuttaki virgüllü rakamlar içindir |
| Virgül | 64 | -22250738585072013e-308 | +17976931348623157e+308 | Büyük virgüllü rakamlar içindir |
| İz | 32 | 0 | 4294967295 | Türü belirsiz, işaretli adresleme içindir |
| Tarih | 32 | -2147483648 | +2147483647 | Sadece Tarih bilgisini içermek içindir |
| Tarih | 64 | -9223372036854775808 | +9223372036854775807 | Sadece Tarih bilgisini içermek içindir |
| TarihSaat | 32 | -2147483648 | +2147483647 | Tarih ve Saat bilgisini içermek içindir |
| TarihSaat | 64 | -9223372036854775808 | +9223372036854775807 | Tarih ve Saat bilgisini içermek içindir |
| İşaret | 32 | 0 | 4294967295 | Bellekte bir fiziksel alana işaret eder. İşaret aynı zaman özel bir kuraldır. |
| İşaret | 64 | 0 | 18446744073709551615 | Bellekte bir fiziksel alana işaret eder. İşaret aynı zaman özel bir kuraldır. |
| ProsedürBağı | 32 | 0 | 4294967295 | Türsüz bir prosedür bağı oluşturur ve başka bir ProsedürBağı'na veya bir Prosedüre veya bir Metota işaret eder. ProsedürBağı aynı zaman özel bir kuraldır. |

| Veri Türü | Bit_Adeti | Minimum | Maksimum | Amacı |
|---|---|---|---|---|
| AsciiMetni | * | Bayt | | Bu tür C kütüphaneleri ile veri alışverişi için özel olarak sunulur. Basit standart olarak Metin türünü kullanır ve tüm karakterleri Unikod modelinde Metin türünde tutar |
| Metin | * | ÇiftBayt | | Basitin sunduğu standart karakter türüdür. Tüm metinsel veriler bu türle tutulur |
| Esnek | 32 | 0 | 4294967295 | Esnek özel bir türdür. Tüm diğer türleri destekler. Bazen gerekli olduğu için sunulmaktadır. Fakat standart olarak asla kullanılmamalıdır. Sadece gerçekten gerektiği yerde kullanılması doğrudur. Daha ayrıntılı bilgi için tıkla. |

| Veri Türü | Bit_Adeti | Minimum | Maksimum | Amacı |
|---|---|---|---|---|
| Esnek | 64 | 0 | 18446744073709551615 | Veri Türü “Esnek” özel bir türdür. Bu türün özelliği ise, diğer türlerin hepsinden olabilir. Esnek ihtiyaç durumunda yeniden yapılandırılır. Esnek türünün kullanılmasının gerekli olduğu durumlar mutlaka vardır. Ama önerilen o ki, mümkün oldukça bu türden az kullanılmasıdır. Çünkü “Esnek” normalden çok daha fazla hesaplama ve işlem adımları gerektiriyor. Bu durum programın yavaşlamasına ve ağır çalışmasına sebebiyet verebilir. Tabloya bakıldığında kapasite açısından aynı kapasiteye sahip türler ile karşılaşıyoruz. Kapasiteleri aynı olsa bile kullanım durumlarında farklılıkları ortaya çıkmaktadır. Daha ayrıntılı bilgi için tıkla. |

# Değişken nedir?

Bir değişken bir veri türünden oluşan adrestir. Bu adrese Basit sayesinde bir isim veririlir, o isim altında o adrese erişebilir, oradaki veriyi okuyabilir ve değiştirebilirsin.

Bir değişkeni kullanmadan önce onu tanımlamak gerekiyor. Bunu yapmak için Basit iki farklı yöntem sunuyor:

```
Değişken  KüçükRakam  YaşımKaç
```

Veya

```
Belirle  KüçükRakam  YaşımKaç
```

Değişkeni, lokal birimde olması şartıyla tanımlarken ona veri girebilirsin:

```
Değişken  KüçükRakam  YaşımKaç  =  36
```

Birçok değişkeni aynı anda tanımlamak işini kolaylaştırır:

```
Değişken  KüçükRakam  a;  b;  c;  d;  e
```

Farklı türlerden tanımlamakta mümkün:

```
Değişken  KüçükRakam  a;  b;  c;  Virgül  d;  e;  Metin  met1;  met2
```

# Değişken nasıl okunur ve değişkene nasıl yazılır?

"İşlem Operatörleri" bölümü bu konuda daha kapsamlı bilgi vermektedir. Burada sadece kısa örneklerle yetineceğiz.

Örnek 1:

```
Değişken  Rakam  a  ? Değişkeni tanımlıyoruz

a  =  5  ? Değişkene veri giriyoruz

Denetim  a  ? Değişkeni okuyoruz
```

Örnek 2:

```
Değişken  Rakam  a;  b;  c  ? Değişkenler tanımlıyoruz

a  =  5  ? veri
b  =  3  ? giriyoruz
c  =  a  +  b  ? yazarken, kaynakları okuyoruz

Denetim  c  ? Sonucu okuyoruz
```

Örnek 3:

```
Değişken  Rakam  a;  b;  c  ? Değişkenler tanımlıyoruz

a;  b;  c  =  8  ? Tümüne 8 değerini giriyoruz

Denetim  a  &&  b  &&  c  ? Sonuçları okuyoruz
```

# "Değişken" ve "Belirle" arasındaki fark nedir?

“Değişken” komutunun sunduğu tüm imkanları “Belirle” aynen sunmaktadır. Fakat “Belirle” daha çok imkan sunmaktadır ve bunları yeri geldiğince yavaş yavaş izah edeceğim.