

## Nesne Yönelimli Programlama Tekniği

Nesne Yönelimli Programlama Tekniği 'nin (NYPT) tek bir cümleyle tam bir tanımını yapmak mümkün değildir. Fakat kısaca "sınıflar kullanarak program yazma tekniğidir" denebilir. Aslında NYPT birtakım anahtar kavramların birleşiminden oluşmaktadır. Bu anahtar kavramlar bütünüyle birbirinden ayrıktır. Bu kavramlar içiçe geçmiştir. Bu anahtar kavramları birbirleriyle kesişen daireler olarak düşünebiliriz. Tüm bu anahtar kavramların temelinde kodun daha iyi algılanması ve yönetilmesi vardır. Örneğin, benzer işi yapan metotlara aynı ismi vermek iyi bir tekniktir. Çünkü bu durumda o metotları kullanan kişiler "çok şey varmış" duygusundan uzaklaşıp "az şey var" gibi algı oluştururlar.

NYPT insanın doğayı algılayış biçimini model almaktadır. İnsanlar herşeyi nesne biçiminde ele alıp onları kullanırlar. Örneğin: biz kalemler farklı olsa bile konuşurken hepsine kalem deriz. Gerekirse onun başka özelliklerini söyleriz.

## Bilgisayarın Kısa Tarihi

Elektronik düzeyde bugün kullandığımız bilgisayarlara benzer ilk aygıtlar 1940'lı yıllarda geliştirilmeye başlanmıştır. Ondan önce hesaplama işlemlerini yapmak için pek çok mekanik aygıt üzerinde çalışılmıştır. Bunların bazıları kısmen başarılı olmuştur ve belli bir süre kullanılmıştır. Mekanik bilgisayarlardaki en önemli girişim Charles Babbage tarafından yapılan "Analytical Engine" ve "Difference Engine" aygıtlarıdır. Analytical Engine tam olarak bitirilememiştir. Fakat bunlar pek çok çalışmaya ilham kaynağı olmuştur. Hatta bir dönem Babbage'in asistanlığını yapan Ada Lovelace bu Analytical Engine üzerindeki çalışmalarından dolayı dünyanın ilk programcısı kabul edilmektedir. Şöyle ki: Rivayete göre Babbage Ada'dan Bernolli sayılarının bulunmasını sağlayan bir yönerge yazmasını istemiştir. Ada'nın yazdığı bu yönergeler dünyanın ilk programı kabul edilmektedir. (Gerçi bu yönergelerin bizzat Babbage'in kendisi tarafından yazılmış olduğu neredeyse ispatlanmış olsa bile böyle atıf vardır.) Daha sonra 1800'lü yılların son çeyreğinde itibaren elektronikte hızlı bir ilerleme yaşanmıştır. Bool cebri ortaya atılmış, çeşitli devre elemanları kullanılmaya başlanmış ve mantık devreleri üzerinde çalışmalar başlatılmıştır. 1900'lü yılların başlarında artık yavaş yavaş elektromekanik bilgisayar fikri belirmeye başlamıştır. 1930'lu yıllarda Alan Turing konuya matematiksel açıdan yaklaşmış ve böyle bir bilgisayarın hangi matematik problemleri çözebileceği üzerine kafa yormuştur. Turing bir şerit üzerinde ilerleyen bir kafadan oluşan ve ismine "Turing Makinası" denilen soyut makina tanımlamıştır ve bu makinanın neler yapabileceği üzerinde kafa yormuştur. ACM Turing'in anısına bilgisayarın Nobel ödülü gibi kabul edilen Turing ödelleri vermektedir.

Dünyanın ilk elektronik bilgisayarının ne olduğu konusunda bir fikir birliği yoktur. Bazıları Konrad Zuse'nin 1941'de yaptığı Z3 bilgisayarını ilk bilgisayar olarak kabul ederken bazıları Harvard Mark 1, bazıları da ENIAC'ı kabul etmektedir.

İlk bilgisayarlarda transistör yerine vakum tüpler kullanılıyordu. (Vakum tüpler transistör görevi yapan büyük, ısınma problemi olan lambaya benzer devre elemanlarıdır). Modern bilgisayar tarihi 3 döneme ayrılarak incelenebilir:

- 1) Transistör öncesi dönem (1940-1950'lerin ortalarına kadar)
- 2) Transistör dönemi (1950'lerin ortalarından 1970'lerin ortalarına kadar)
- 3) Entegre devre dönemi (1970'lerin ortalarından günümüze kadarki dönem)

Transistör icad edilince bilgisayarlar transistörlerle yapılmaya başlandı ve önemli aşamalar bu sayede kaydedildi. Bilgisayar devreleri küçüldü ve kuvvetlendi. O zamanların en önemli firmaları IBM, Honeywell, DEC gibi firmalardı.

Transistörü bulan ekipten Shockley bir şirket kurarak yanına genç mühendisler aldı. Bu ekipteki Noyce ve arkadaşları ilk entegre devreleri yaptılar ve Intel firmasını kurdular. Böylece Entegre devre devrine geçilmiş oldu.

Dünyanın entegre olarak üretilen ilk mikroişlemcisi Intel'in 8080'i kabul edilmektedir. Intel daha önce 4004, 8008 gibi entegreler yaptıysa da bunlar tam bir mikroişlemci olarak kabul edilmemektedir. O yıllara kadar dünyadaki bilgisayarlar sayılabilecek kadar azdı. Bunlar yüzbinlerce dolar fiyatı olan dev makinalardı ve IBM gibi şirketler çoğu kez bunları kiraya verirdi. Kiilerin evine bilgisayar alması uçuk bir fikirdi.

Intel 8080'i yaptığından bundan bir kişisel bilgisayar yapılabileceği onların aklına gelmemiştir. Kişisel bilgisayar fikri Ed Roberts isimli bir girişimci tarafından ortaya atılmıştır. Ed Roberts 8080'i kullanarak Altair isimli ilk kişisel bilgisayarı yaptı ve "Popular Electronics" isimli dergiye kapak oldu. Altair makina dilinde kodlanıyordu. Roberts buna Basic derleyicisi yazacak kişi aradı ve Popular Electronics dergisine ilan verdi. İlane o zaman Harvard'ta öğrenci olan Bill Gates ve Paul Allen başvurular. Böylece Altair daha sonra Basic ile piyasaya sürüldü. Gates ve Allen okuldan ayrıldılar Microsoft firmasını kurdular. (O zamanlar bu yeni kişisel bilgisayarlara mikrobilgisayarlar denilmekteydi). Amerika'da bu süreç içerisinde bilgisayar kulüpleri kuruldu ve pek çok kişi kendi kişisel bilgisayarlarını yapmaya çalıştı. Steve Jobs ve Steve Wozniak Apple'ı böyle bir süreçte kurmuştur.

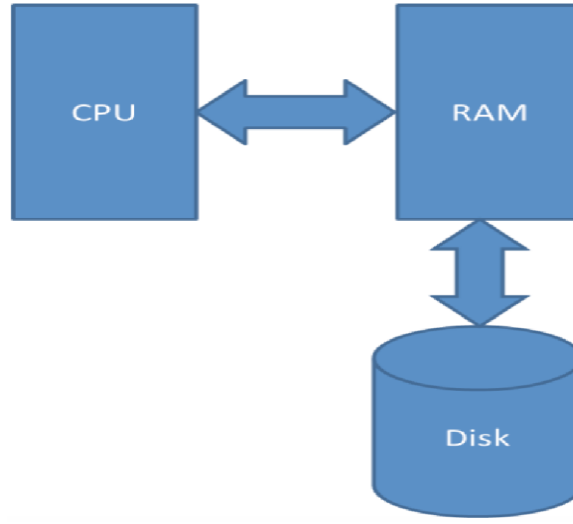
IBM kişisel bilgisayar konusunu hafife aldı. Fakat yine de bir ekip kurarak bugün kullandığımız PC'lerin donanımını IBM tasarlamıştır. Ancak IBM küçük iş olduğu gerekçesiyle bunlara işletim sistemini kendisi yazmadı taşeron bir firmaya yazdırmak istedi. Microsoft IBM ile anlaşarak DOS işletim sistemini geliştirdi. İlk PC'lerin donanımı IBM tarafından, yazılımı Microsoft tarafından yapılmıştır. Microsoft IBM'le iyi bir anlaşma yaptı. IBM uzağı göremedi. Anlaşmaya göre başkalarına DOS'un satışını tamamaen Microsoft yapacaktı. IBM PC için patentleri ihmal etti. Pek çok firma IBM uyumlu daha ucuz PC yaptılar. Fakat bunların hepsi işletim sistemini Microsoft'tan satın aldı. Böylece Microsoft 80'li yıllarda çok büyüdü.

İlk devirlerde bilgisayar programları ancak birkaç sayfa uzunluğunda oluyordu. Sonra transistör devrinde onbin satırdan oluşan projeler yazılmaya başlandı. Sonra yüzbin satırlara çıkıldı. PC'lerin başlarında donanım yetersizdi. PC projeleri genellikle onbinlerle ölçülen satırlarda kalıyordu. Ancak donanımlar iyileştikçe yazılımlarda kod büyümesi yaşanmaya başladı. O zamanlar kullanılan prosedürel tekniğin artık yetersiz kaldığı görülmüştür. İşte NYPT donanımların gelişmesiyle yazılımlarda ortaya çıkan kod büyümesi ile algısal olarak mücadele etmek için geliştirilmiştir. NYPT'nde artık fonksiyonlarla değil sınıflarla konuşulur. Böylece "çok şey var" duygusundan uzaklaşarak "az şey var" duygusuna kapılır.

NYPT yazılım endüstrisine 90'lı yıllarda girmiştir. Fakat ilk denemeler 80'lerin başlarında yapılmıştır. Bugün yazılım endüstrisinde ağırlıklı olarak NYPT kullanılmaktadır.

## **Adres Kavramı**

Bir bilgisayar sisteminde CPU (Central Processing Unit), RAM (Random Access Memory) ve Disk en önemli üç birimdir. CPU entegre devre biçiminde üretilmiş olan mikroişlemcidir. Bütün hesaplamalar ve karşılaştırmalar CPU tarafından yapılmaktadır. RAM'ler de bir kart üzerinde monte edilmiş entegre devre modüllerinden oluşur. CPU, RAM olarak elektriksel olarak bağlantılıdır. Disk bilgisayar kapatıldığında bilgilerin saklandığı birimdir. Disk ile RAM arasında da bağlantı vardır.



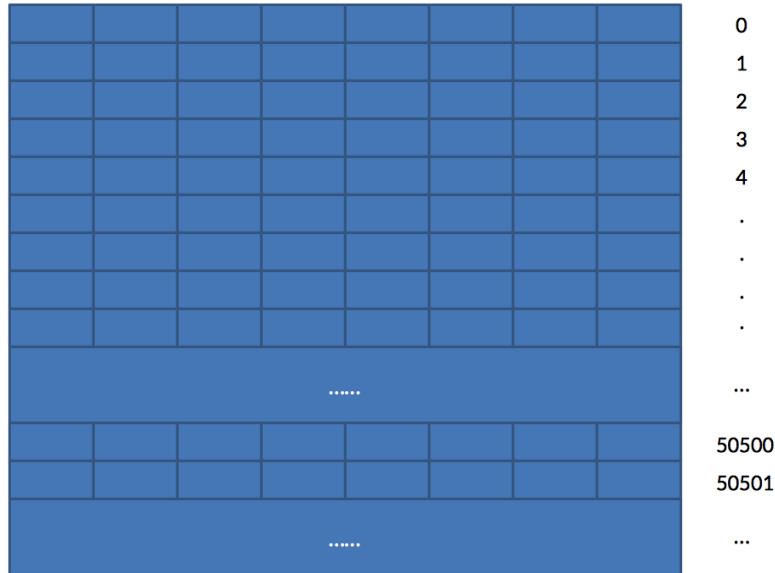
Programlama dillerinde değişkenler RAM'de tutulur. İşlemler CPU'da yapılır. Örneğin:

```
a = b + c;
```

ifadesinde a, b, c RAM'dedir. Fakat toplama ve atama işlemi CPU tarafından yapılır. CPU RAM'e erişir oradan b'yi ve c'yi alır. Bunu elektrik devreleriyle toplar ve sonucu da a'ya atar.

Dosyalar diskte bulunurlar. Bilgisayarı kapattığımızda dosyalar kaybolmazlar. İşletim sistemlerinin dosya işlemleri ile uğraşan bölümüne dosya sistemi denir.

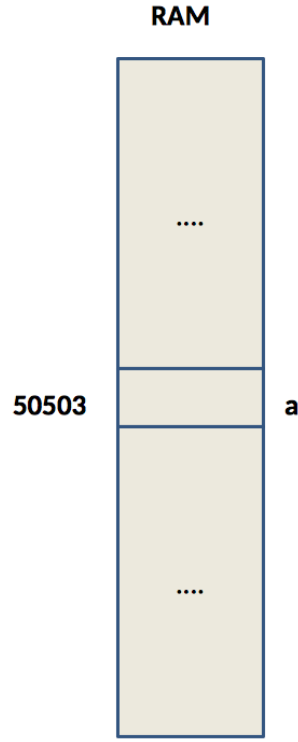
Bellek (RAM) byte'lardan byte'lar da bitlerden oluşur. 1 byte 8 bittir. Bit (binary digit) 0 ya da 1 değerini tutan bellek hücreleridir.



Bellekteki her bir byte'a ilki sıfır olmak üzere artan sırada bir sayı karşılık getirilmiştir. Bu sayılara adres denilmektedir. Değişkenler bellekte olduğuna göre onların da birer adresi vardır. Örneğin:

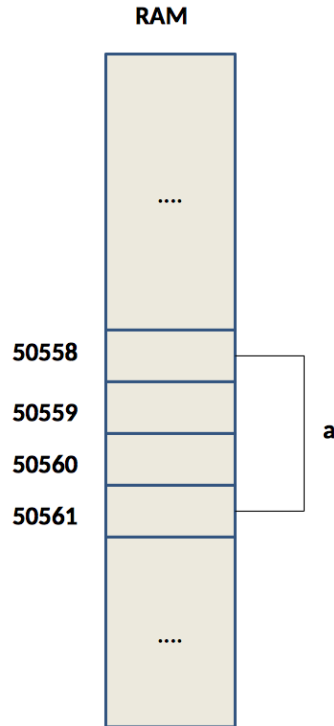
```
byte a;
```

Burada a 1 byte yer kaplayan bir değişkendir. a'nın da bir adresi vardır. Örneğin:



Burada a'nın adresi 50503'dür. Bir byte'tan daha uzun olan değişkenlerin adresleri onların yalnızca en düşük numaralı adresleriyle ifade edilir. Örneğin:

```
int a;
```



Burada a 4 byte uzunluğundadır ve a'nın adresi 50558'dir.

### Değişkenlerin Ömürleri

Çalıştırılabilen (executable) bir dosya diskte bulunur. Çalıştırılmak istendiğinde işletim sistemi tarafından diskten alınarak RAM'e yüklenir. Programın çalışması bittiğinde program RAM'den boşaltılır.

Bir değişkenin bellekte yer kapladığı zaman aralığına ömür denilmektedir.

## Yerel Değişkenlerin Ömürleri

Yerel değişkenler sürekli bellekte tutulmazlar. Akış yerel değişkenin bildirildiği noktaya geldiğinde yerel değişken yaratılır, akış yerel değişkenin bildirildiği bloktan çıktığında o blokta bildirilmiş tüm yerel değişkenler yok edilir. Yani bir programda binlerce yerel değişken olsa da belli bir anda onların hepsi bellekte yer kaplamazlar. Yani toplamda bellekte kapasitenin çok üstünde yer ayrılmıştır. Ancak yaratılıp yok edildikleri için bellek bu şekilde tüm yerel değişkenler için yeterli olmaktadır.

Yerel değişkenlerin yaratıldığı alana "stack" denilmektedir. Stack işletim sistemi tarafından RAM'de organize edilen bir bölümdür. Stack'in yeri ve uzunluğu sistemden sisteme değişebilmektedir. Stack'te yaratma ve yoketme işlemleri tek bir makine komutuyla yani çok hızlı bir biçimde yapılmaktadır.

Yerel bir değişkene onun bildirildiği bloğun dışından erişilememesinin asıl nedeni o değişkenin o blok dışında ömrünün bitmesindendir. Değişken bildirilmeden önce de yaşamıyor durumdadır:

```
{
    //...
    a = 10; //error: a henüz bildirilmediği için bellekte (stack'de) değil
    {
        int a;
        //....
    }
    //...
    a = 8; //error: a diye bir değişken artık bellekte (stack'de) değil.
}
```

## Metot Parametre Değişkenlerinin Ömürleri

Metodun parametre değişkenleri metot çağrıldığında yaratılır, metot bittiğinde yok edilir. Metot parametre değişkenleri sanki metodun başında bildirilen yerel değişkenler gibidir ve stack' de yaratılmaktadırlar:

```
package csd;

class App {
    public static void foo(int val)
    {
        //...
    }

    public static void main(String[] args)
    {
        foo(20); // Parametre değişkeni val stack te yaratıldı
        //foo çağırması bittiğinde val yok edildi

        {
            //...
            int a; //Stack de yaratıldı
            //..
        } // a yok edildi
    }
}
```