

**İSTATİSTİKSEL VERİLERİN
İKİ BOYUTLU GRAFİKSEL GÖSTERİMİ**

YÜKSEK LİSANS TEZİ

H. Naci YANANLI

Tezin Enstitüye Verildiği Tarih : 27 Mayıs 1996

Tezin Savunulduğu Tarih : 11 Haziran 1996

Tez Danışmanı

: Yrd. Doç. Dr. Celal TUNCER

Diğer Jüri Üyeleri

: Prof. Dr. Erol ÜÇDAL

Doç. Dr. Fikret BALTA

[Signatures]

HAZİRAN 1996

ÖNSÖZ

Günümüzde ekonomik, siyasal ve sosyal olayların incelenmesinde, teknik konularda verilerin değerlendirilmesinde kullanılan istatistikler ve bunların bilgisayardaki uygulamaları bu araştırmaların yapılmasında büyük kolaylık sağlamaktadır. Bilimsel deneyler sonucu elde edilen verilere göre veya verilen bir matematiksel ifadeye göre grafiğin çizimi istatistiksel olarak bu alandaki araştırmalara yön vermektedir.

Bu tez çalışması sırasında yönlendirmeleri ve yardımları ile her türlü desteği bana sağlayan tez danışmanım Yrd. Doç. Dr. Celal TUNCER'e ve hayatım boyunca maddi ve manevi her türlü desteği sağlayan sevgili annem ve babam'a şükranlarımı sunmayı bir borç bilirim.

HAZİRAN 1996

H.Naci YANANLI

İÇİNDEKİLER

ÖZET	VI
SUMMARY.....	VII
BÖLÜM 1. GİRİŞ	1
BÖLÜM 2. GRAFİK CİHAZLAR.....	2
2.1 Storage Tube Grafik Ekranları.....	2
2.2 Kaligrafik Yenilemeli Grafik Ekranlar.....	3
2.3 Raster Refresh Grafik Ekranlar	5
2.4 Katod Işın Tübü Temelleri.....	11
2.5 Renkli CRT Raster Scan Temelleri	12
BÖLÜM 3. İKİ BOYUTLU GRAFİKSEL DÖNÜŞÜMLER	14
3.1 Noktaların Gösterimi	14
3.2 Çevrimler ve Matrisler	15
3.3 Noktaların Dönüşümü	15
3.3.1 Düz Çizgilerin Dönüşümü	18
3.4 Döndürme.....	19
3.5 Yansıma.....	22
3.6 Ölçekleme.....	24
3.7 Birleşik Dönüşümler	25
3.8 Dönüşümler ve Homojen Koordinatlar	26
3.9 Keyfi Bir Nokta Etrafında Döndürme	27
BÖLÜM 4. VERİLERİN İSTATİSTİKSEL DEĞERLENDİRİLMESİ.....	30
4.1 Ortalamalar	30
4.1.1 Aritmetik Ortalama	32
4.1.2 Geometrik Ortalama.....	32
4.1.3 Harmonik Ortalama	33

4.1.4 Kareli Ortalama.....	34
4.2 Dağılıma	34
4.2.1 Ortalama Sapma	35
4.2.2 Standart Sapma.....	35
4.2.3 Değişim Katsayısı.....	36
4.3 İstatistik İlişkiler	36
4.3.1 Bağımlılık İndeksi	36
4.3.2 Bağımlılık Katsayısı.....	37
4.3.3 Korelasyon Katsayısı	38
BÖLÜM 5. EĞRİ VE OBJE ÇİZİMİ	40
5.1 Nesneye Yönelik Programlama (Object Oriented Programming). 41	
5.1.1 Nesne Nedir ?	41
5.1.2 Nesneye Dayalı Programlamadaki Farklar	42
5.1.3 Visual Basic ve NYP	43
5.1.4 Hiyerarşi İlişkisi	46
5.2 İki Boyutlu Eğri Çizimi	47
5.3 Grafik Elemanları	48
5.3.1 Grafik Başlığı.....	48
5.3.2 Yatay Eksen Başlığı	52
5.3.3 Dikey Eksen Başlığı	52
5.3.4 Seri Başlıkları.....	52
5.3.5 Grafik Çizim Alanı	53
5.3.5.1 X Eksen Alanının Ölçeklenmesi ve Etiketlenmesi ...	53
5.3.5.1 Y Eksen Alanının Ölçeklenmesi ve Etiketlenmesi ...	54
5.3.6 XY Dağılım Grafikleri (XY scatter).....	55
5.3.7 Pay (Pasta) Grafikler	56
BÖLÜM 6. UYGULAMA	58
6.1 Menüler ve Dosyalamalar	58
6.2 Menülerin Tanıtımı.....	59
6.3 İstagraf Fonksiyonları	61
6.3.1 Matematiksel Fonksiyonlar.....	61
6.3.2 İstatistik Fonksiyonları	63

6.4 Grafik Penceresinin Kullanımı	64
SONUÇ ve ÖNERİLER.....	66
KAYNAKLAR	67
EK A	68
ÖZGEÇMİŞ.....	117



ÖZET

Bu çalışmada, eğri ve nesne çizimlerine yönelik olarak, verilerin oluşturulması, analizi ve oluşturulan bu değerlerden yararlanılarak, eğri ve objelerin çizimi ile ilgili çeşitli algoritmalar kullanılmıştır.

Verilerin oluşturulması ve analizi aşamasında; bir serinin alacağı değerlerin oluşturulması, bu değerlerin analizinde yardımcı olacak istatistiksel hesaplamaların yapılması ile ilgili yöntemler uygulanmıştır. Aynı zamanda verilen bir fonksiyonun, bilgisayar tarafından incelenmesi (parse) ve yorumlanması da yapılmıştır.

Çalışmada ayrıca, eğri çizimi ile ilgili özellik ve yöntemlerle, obje çizimi ile ilgili matematiksel dönüşümler kullanılarak, bunların bir görüntü ortamında çizimini yapan algoritmalar gerçekleştirilmiştir.

Bu çalışma sırasında geliştirilen uygulamada, nesneye yönelik programlama yardımcı bir araç olarak kullanılmış ve daha sonraki grafik çizimi ve veri analizi uygulamalarına temel teşkil edebilecek bir sistem kurulmaya çalışılmıştır.

SUMMARY

TWO-DIMENSIONAL REPRESENTATIONS OF STATISTICAL DATAS

In this study, a computer program is developed in order to show some statistical data, graphical on computer screen or printer. For this purpose, some commercial computer program package can be found, but the program developed in this study is different from them. It is very specific, easy to use, and user-friendly.

The thesis is divided into five main parts.

1. Introduction
2. Computer sceen components
3. Graphics transformation methods for two-dimensional case
4. Statistical data and their relations
5. Drawing series and objects

Computer Screen Components

The display medium for computer graphics-generated pictures has become widely diversified. Typical examples are cathode ray tube (CRT) displays, flat panel displays, pen-and-ink plotters, dot matrix, electrostatic or laser printer plotters and film. In addition to display devices, image capture devices are becoming of increasing importance.

The three most common types of CRT display technologies are direct-view storage tube (line drawing), calligraphic (line drawing) refresh and raster scan (point plotting) refresh displays. The most common types of flat panel displays include plasma-gas discharge, electroluminescent, liquid crystal and light-emitting diode technologies. With recent advances, an individual display may incorporate more than one technology. In discussing the various displays we take a user's, or conceptual, point of view; i.e., we are generally concerned with functional capabilities.

Storage Tube Graphics Displays

The direct-view storage tube¹ is conceptually the simplest of the CRT displays. The storage tube displays, also called a bistable storage tube, can be considered a CRT

¹ Storage tube graphics displays are no longer manufactured. However, there are still literally thousands of them in use, hence the discussion.

with a long-persistence phosphor. A line or character remains visible (up to an hour) until erased.

Calligraphic Refresh Graphics Displays

In contrast to the storage tube displays, a calligraphic (line drawing or vector) refresh CRT display uses a very short persistence phosphor. These displays are frequently called random scan displays. Because of the short persistence of the phosphor, the picture painted on the CRT must be repainted or refresh many times each second. The minimum refresh rate is at least 30 times each second, with a recommended rate of 40 to 50 times each second. Refresh rates much lower than 30 times each second result in a flickering image. The effect is similar to that observed when a movie film is run too slowly. The resulting picture is difficult to use and disagreeable to look at.

The basic calligraphic refresh display requires two elements in addition to the CRT. These are the display buffer and the display controller. The display buffer is contiguous computer memory containing all the information required to draw the picture on the CRT. The display controller's function is to repeatedly cycle through this information at the refresh rate. Two factors which limit the complexity (number of vectors displayed) of the picture are the size of the display buffer and the speed of the display controller.

Raster Refresh Graphics Displays

Both the storage tube CRT display and the calligraphic refresh displays are line drawing devices. That is, a straight line can be drawn directly from any addressable point to any other addressable point. In contrast, the raster CRT graphics display is a point plotting device. A raster graphic device can be considered a matrix of discrete cells, each of which can be made bright. It is not possible, except in special cases, to directly draw a straight line from one addressable point or pixel (picture element), in the matrix to another addressable point or pixel. The line can be approximated by a series of dots (pixels) close to the path of the line.

The most common method of implementing a raster CRT graphics device uses a frame buffer. A frame buffer is a large, contiguous piece of computer memory. As a minimum there is a one memory bit for each pixel in the raster. This amount of memory is called a bit plane. A 512x512 element square raster requires 2^{18} ($2^9 = 512$; $2^{18} = 512 \times 512$) or 262,144 memory bits a single bit plane. The picture is built up in the frame buffer 1 bit at a time. Since a memory bit has only two states (binary 0 or 1) a single bit plane yields a black-and-white (monochrome) display. Since the frame buffer is a digital device, while the raster CRT is an analog device, conversion from a digital representation to an analog signal must take place when information is read from the frame buffer and displayed on the raster CRT graphics device. This is accomplished by a digital-to-analog converter (DAC). Each pixel in the frame buffer must be accessed and converted before it is visible on the raster CRT.

Cathode Ray Tube Basics

A frame buffer as described above is not itself a display device. It is simply used to assemble and hold a digital representation of the picture. The most common display device used with a frame buffer is a video (TV) monitor. An understanding of the raster displays, and to some extent line drawing refresh displays, requires a basic understanding of CRT's display techniques.

It is at this point that line drawing displays, both storage and refresh, and raster scan displays differ. In a line drawing display the electron beam may be deflected directly from an arbitrary position to any other arbitrary position on the face of the CRT. A perfectly straight line results. In contrast, in a raster scan display the beam is deflected in a set, rigidly controlled pattern. This pattern comprises the video picture.

Color CRT Raster Scan Basics

A color raster scan CRT is similar to the standard black-and-white CRT described in the previous section. In the color raster scan CRT there are three electronic guns, one for each of the primary colors, red, green and blue. The electronic guns are frequently arranged in a triangular pattern corresponding to a similar triangular pattern of red, green and blue phosphor dots on the face of the CRT.

Graphics Transformation Methods for Two-Dimensional Case

Points and lines which join them, along with an appropriate drawing algorithm, we used to represent objects or to display information graphically. When visualizing an object, it may be desirable to scale, rotate, translate, distort or develop a perspective view of the object.

Representation of Points

A point is represented in two dimensions by its coordinates. These two values are specified as the elements of a 1-row, 2-column matrix:

$$[X] = [x \ y]$$

In three dimensions a 1 x 3 matrix

$$[X] = [x \ y \ z]$$

is used. Alternately, a point is represented by a 2-row, 1-column matrix

$$[X] = \begin{bmatrix} x \\ y \end{bmatrix}$$

in two dimensions or by

$$[X] = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

in three dimensions. Row or column matrices are frequently called position vectors.

Transformation of Points

Consider the results of the multiplication of a matrix $[x \ y]$ containing the coordinates of a point P and a general 2×2 transformation matrix:

$$[X][T] = [x \ y] \begin{bmatrix} a & b \\ c & d \end{bmatrix} = [(ax + cy) \ (bx + dy)] = [x^\circ \ y^\circ]$$

This mathematical notations means that the initial coordinates x and y are transformed to x° and y° , where $x^\circ = (ax + cy)$ and $y^\circ = (bx + dy)$. We are interested in the implications of considering x° and y° as the transformed coordinates of the point P .

Transformation can be divided into several types, depending on parameter a, b, c, d .

For the case of $a = d = 1$ and $c = b = 0$. The transformation matrix $[T]$ the reduces to the identity matrix. Thus,

$$[X][T] = [x \ y] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [x \ y] = [x^\circ \ y^\circ]$$

and no change in the coordinates of the point P occurs.

Next consider $d = 1, b = c = 0$, i.e.,

$$[X][T] = [x \ y] \begin{bmatrix} a & 0 \\ 0 & 1 \end{bmatrix} = [ax \ y] = [x^\circ \ y^\circ]$$

which, since $x^\circ = ax$, produces a scale change in the x component of the position vector.

For the case of $b = c = 0$, i.e.,

$$[X][T] = [x \ y] \begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix} = [ax \ dy] = [x^\circ \ y^\circ]$$

This yields a scaling of both the x and y coordinates of the original position vector. If $a \neq d$, then the scalings are not equal.

If a and/or d are negative, reflections through an axis or plane occur. To see this, consider $b = c = 0$, $d = 1$ and $a = -1$. Then,

$$[X][T] = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -x & y \end{bmatrix} = \begin{bmatrix} x^0 & y^0 \end{bmatrix}$$

and a reflection through the y-axis results. If $b = c = 0$, $a = 1$ and $d = -1$, then a reflection through the x-axis occurs.

Transformation of Straight Lines

A straight line can be defined by two position vectors which specify the coordinates of its end points. The position and orientation of the line joining these two points can be changed by operating on these two position vectors. The actual operation of drawing a line between two points depend on the display device used. Here, we consider only the mathematical operations on the position vectors of the end points.

Statistical Data And Their Relations

The main purpose of the program developed is used for processing data and showing them in graphical form on a computer screen. Some statistical relations can be used directly in program. They are given below:

Average: One of the most important method is average in order to represent a data series or group. Average value of a data series will be the between minimum and maximum value of the data series. It can be divided into several types, but four of them used in program which are harmonic average, geometric average, arithmetical average and squared average.

Deviation: Although average is very useful to represent a data series, it is not sufficient. Because, two data series may have the same average value but different elements. Therefore, besides average, deviation of a data series should be considered. There is two types of deviation which are average deviation and standard deviation, used in the program.

Statistical Relations: Averages and deviations gives characteristic of a data series. When there were two series, we may want to learn if there is a relation between this two series. This can be find by computing some relation coefficients. In this section most known types of deviations which they are dependency index, dependency coefficient and correlation coefficients are discussed.

Drawing Object and Series

Drawing can be considered in two groups: showing statistical series and drawing objects.

Statistical Graphics: In this case, one or more data series can be shown as a graphic on a computer screen. Statistical graphics can be line, bar, area, xy scatter or pie. These graphics have different properties depending on data types. For example, pie graphs show the percentage of elements in the total value of the series. Therefore, it cannot be used for the second series in pie graphs. In this study, the graphics are considered as an object, and subdivided into some useful objects; graph titles, horizontal axis titles, vertical axis titles, series titles and graph area. Graph area is also subdivided into three objects which are x-axis, y-axis and series. In the developed program, each object draws itself, and chooses its own window location.



BÖLÜM 1. GİRİŞ

Bu çalışmada, bilgisayarda girilen verilere veya verilen bir matematiksel ifadeye göre grafik çizimi ve istatistik işlemleri oluşturan bir program geliştirilmiştir. Program nesneye yönelik programlamaya uygun Visual Basic programlama dilinde yazılmıştır.

Tezin ikinci bölümü, bilgisayarlarda kullanılan görüntü elemanlarını içermektedir. Görüntü elemanları tarihsel gelişim sırasıyla storage tube grafik ekranlardan günümüzde kullanılan raster refresh grafik ekranlara kadar olan ekran tipleri incelenmiştir.

Grafiklerin matematiksel ifadeleri ve bunlara dayanarak ölçekleme, döndürme, taşıma ve aynalama gibi dönüşümler üçüncü bölümde verilmiştir.

İstatiksel yöntemler tezin dördüncü bölümünü oluşturmaktadır. Korelasyon, standart sapma, varyans, ortalama, kareli ortalama ve bağımlılık indeksi gibi tanımlar incelenmiş ve bunlara ait matematiksel bağıntılar verilerek nasıl hesaplandığı anlatılmıştır.

Beşinci bölümde eğri çizimine yönelik özellikler anlatılarak, gerekli algoritmalar çeşitli grafik türleri için verilmeye çalışılmıştır. Ayrıca programlamaya dayalı temel kavramlar ve programın geliştirildiği Visual Basic programlama diline ait özellikler anlatılmıştır.

Altıncı bölümde uygulamanın kullanıcı arayüzü tanıtılmış ve kullanımı ile ilgili bilgiler verilmiştir.

BÖLÜM 2. GRAFİK CİHAZLAR

Bu bölümde bilgisayarlarda kullanılan bir grafik gösterim cihazı olarak ekranların fiziksel yapıları hakkında bilgiler verilecektir.

2.1 Storage Tube Grafik Ekranları

Storage tube türü ekranlar temel olarak basit bir CRT ekranıdır. Aynı zamanda çift-durumlu Storage tube olarak da adlandırılan Storage tube'ler uzun süreli fosfora sahip CRT ekranlar olarak da düşünülebilir. Bir satır veya karakter silinene kadar bir saate yakın süreyle görünür durumda kalır. Ekran üzerinde bir satır veya karakter çizmek ve fosforu parlak duruma 'storage state' getirebilmek için belirli bir miktar ışın ekrana gönderilir. [1], [2] Ekranı silmek ve fosforu sönük duruma 'dark state' getirebilmek için ekrana belirli bir miktar voltaj verilir. Silme işlemi yaklaşık 1/2 saniye sürer. Bu işlem sırasında tüm ekrana bu işlem uygulanması nedeniyle tüm ekrandaki satırlar ve karakterler silinir. Bu nedenle, istenen herhangi bir satır veya karakter silinemez ve dinamik görüntüler ve animasyonlar kullanmak mümkün değildir. Bir ara durum olan yazma anı modu (write-through mode) bazen, sınırlı bir yenileme (refresh) işlemi olarak kullanılır. Bir fosfor noktasına eşik değerin altında yapılan bir elektronik atış bu noktanın parlamasına yeterlidir. Ancak bu durumda parlama süresi sınırlıdır ve parlama işleminin devam etmesi için ilgili ekran bölümünün sürekli yenilenmesi veya yeniden oluşturulması gerekir.

Storage tube türü ekranlar sınırsız sayıda vektör basabilme yeteneğine sahiptirler. 10 bit'in kullanıldığı durumda 1024x1024 adreslenebilir vektör, 12 bitin kullanıldığı durumda ise 4096x4096 adreslenebilir vektör vardır.

Storage tube satır çizme veya rastgele taramalı bir ekrandır. Bir satır (vektör) adreslenebilir bir noktadan başka bir adreslenebilir noktaya kadar yazdırılabilir. Doğrudan yazma işlemi kolay ve hızlıdır. Calligraphic veya Raster Scan türü ekranlara göre programlaması daha kolaydır. Storage tube ekranlar seri veya paralel arayüzler kullanılarak grafik sistemlerle birleştirilebilir. Ekranın silme işlemlerindeki yavaşlık yüzünden etkileşimli grafik işlemleri yapılamadığından düşük hızlı arayüzler (paralel arayüzler) kullanılabilir.

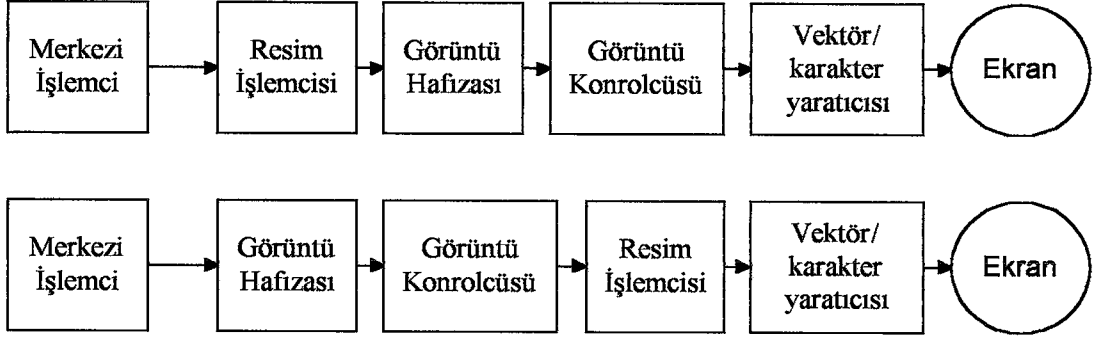
2.2 Kaligrafik Yenilemeli Grafik Ekranlar

Storage tüplerinin tersine Kaligrafik ekranlarda çok kısa saklama süreli fosforlar kullanılır. Bu tür ekranlar sıklıkla 'Rastgele Taramalı' olarak adlandırılır. Fosfor saklama süreleri kısa olduğunda dolayı bir saniye içinde birçok kere yenileme işlemi gerektirir. En düşük yenileme hızları saniyede 30 defadır. Fakat saniyede 40 veya 50 yenileme işlemi yapılması tercih edilir. Saniyede 30 kereden az olan yenileme hızlarında ekranda kırışmalar olur ve bakışı zorlaştırır.

Kaligrafik ekranlarda tartışılacak iki eleman mevcuttur. Bunlar görüntü hafızası ve görüntü kontrolcüsüdür. Görüntü hafızası, ekranda görünen resmi basabilmek için gerekli tüm bilgiyi içeren sürekli bir tampon hafıza bölümüdür. Görüntü kontrolcüsünün görevi yenileme hızına göre bu bilgileri sürekli olarak ekrana göndermektir. Görüntü kalitesini, bu iki etken 'tampon hafızanın büyüklüğü ve kontrolcünün hızı' belirler. Bunun dışında bir başka sorunda tampon hafızaya gelecek olan görüntü üzerinde yapılacak olan işlemlerdir. (kırpma, kaplama vs.)

Şekil 2-1'de yüksek-performanslı kaligrafik ekranların blok diyagramları gösterilmiştir. İki durumda da döndürme, ölçekleme, kırpma gibi görüntü dönüşümlerinin donanım üzerinde bulunan resim işlemcisi (picture processor) tarafından yapıldığı kabul edilmiştir. Birinci durumda resim işlemcisinin hızı (4000-5000 vektör) yenileme oranından daha düşüktür. Bu nedenle Merkezi İşlemci Birimi (MİB, Central Processing Unit, CPU) tarafından gönderilen grafik görüntüler, görüntü belleğinde (display buffer) saklanmadan önce işlenir. Görüntü belleğinde,

Vektör/karakter yaratıcısı tarafından kullanılacak değerler bulunur. Vektörler genellikle ekran koordinatlarının bilgilerini içerir. Görüntü kontrolcüsü, görüntü hafızasından aldığı bilgileri Vektör/karakter yaratıcısına gönderir. Görüntü hafızasının sonuna geldiğinde hafızanın başına döner ve işlemleri tekrar eder.



Şekil 2-1 Kaligrafik yenilemeli ekranların kavramsal yapısı

İlk düzenek aynı zamanda çift hafızalama ile ayrı güncelleme ve yenileme oranları kavramlarını da beraberinde getirir. Bu düzenekte resim işlemcisinin hızı bir yenileme döngüsü esnasında karmaşık bir resim yaratmaya veya varolan bir resim üzerinde değişiklik yapmaya yetmediği için görüntü hafızası iki bölüme ayrılır. Resim işlemcisi hafızanın bir yarısında işlem yaparken, görüntü kontrolcüsünde hafızanın diğer yarısını yenileme işleminde kullanır. Resmin güncelleme işlemi bittiğinde hafızalar yer değiştirir ve işleme devam edilir. Çift hafızalama işlemi sayesinde bir veya daha fazla yenileme zamanı içinde resmin oluşmasına imkân tanır.

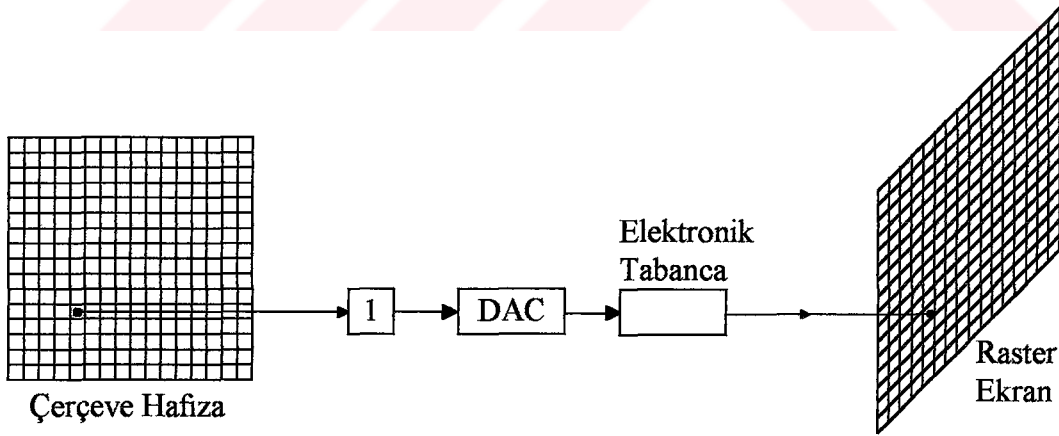
İkinci düzenekte resim işlemcisinin hızı yenileme oranından hızlıdır. Ana MİB tarafından gönderilen resim bilgileri doğrudan görüntü hafızası tarafından alınır. Bilgiler genellikle kayan ondalıklı sayılar şeklindeki dünya koordinatlarındadır. Görüntü kontrolcüsü hafızadan okuduğu bilgileri resim işlemcisine gönderir. Resim işlemciside bir yenileme döngüsü içinde işleyerek vektör yaratıcısına gönderir.

Her iki düzenekte de, vektör, karakter ve resme ait elemanlar görüntü hafızasında saklanır. Bu nedenle her bir eleman diğer elemanlardan bağımsız olarak değiştirilebilir. Yukarıdaki özellikler kısa süreli fosfor ile birleştiğinde dinamik görüntüler elde edilebilir.

2.3 Raster Refresh Grafik Ekranlar

Gerek Storage tüp ve gerekse kaligrafik ekranlar çizgi çizme cihazlarıdır. Adreslenebilir bir noktadan adreslenebilir başka bir noktaya kadar düz çizgi çizme prensibine dayanırlar. Bu tür ekranların tersine raster ekranlar nokta koyma esasına dayanırlar. Bu ekranlar bir matris şeklinde yanyana gelmiş hücreler olarak kabul edilebilir. Hücreler benek (pixel) olarak adlandırılırlar.

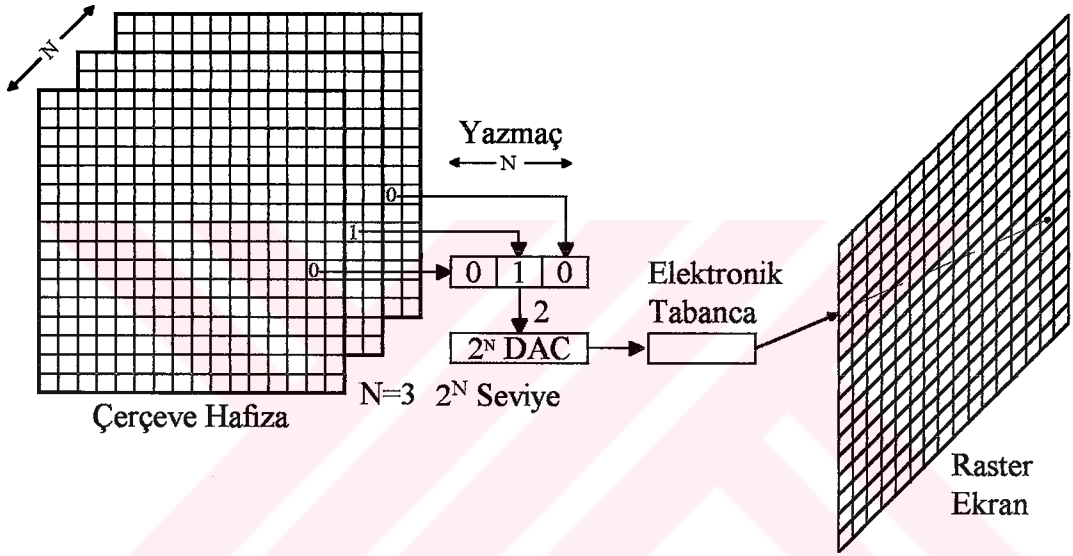
Raster CRT ekranlar genellikle çerçeve bellek (frame buffer) kullanılarak görüntü elde edilir. Çerçeve bellek büyük ve kesintisiz bir bellek parçasıdır. En küçük çerçeve bellek her benek için bir bitin kullanıldığı bellektir ve 'tek bit düzlemi' olarak adlandırılır. Tek bit düzleminde 512×512 benek için 2^{18} ($2^9 = 512$, $2^{18} = 512 \times 512$) 262,144 bellek bitine ihtiyaç vardır. Tek bit düzleminde her benek 1 bit ile ifade edilir. Her bit iki durumu ifade ettiği için (0 veya 1) tek bit düzlemi siyah-beyaz (monochrome) görüntü elde edilebilmesini sağlar. Çerçeve bellek dijital bir cihaz ve raster CRT analog bir cihaz olduğundan, çerçeve bellekten okunan bilginin gösterilmeden önce çevrimi gerekir. Bunu gerçekleştirmek için bir dijital-analog çevirici (digital-analog converter, DAC) bulunur. Bkz. Şekil 2-2.



Şekil 2-2 Tek-bit-düzlemli, siyah-beyaz çerçeve bellekli Raster grafik ekranı

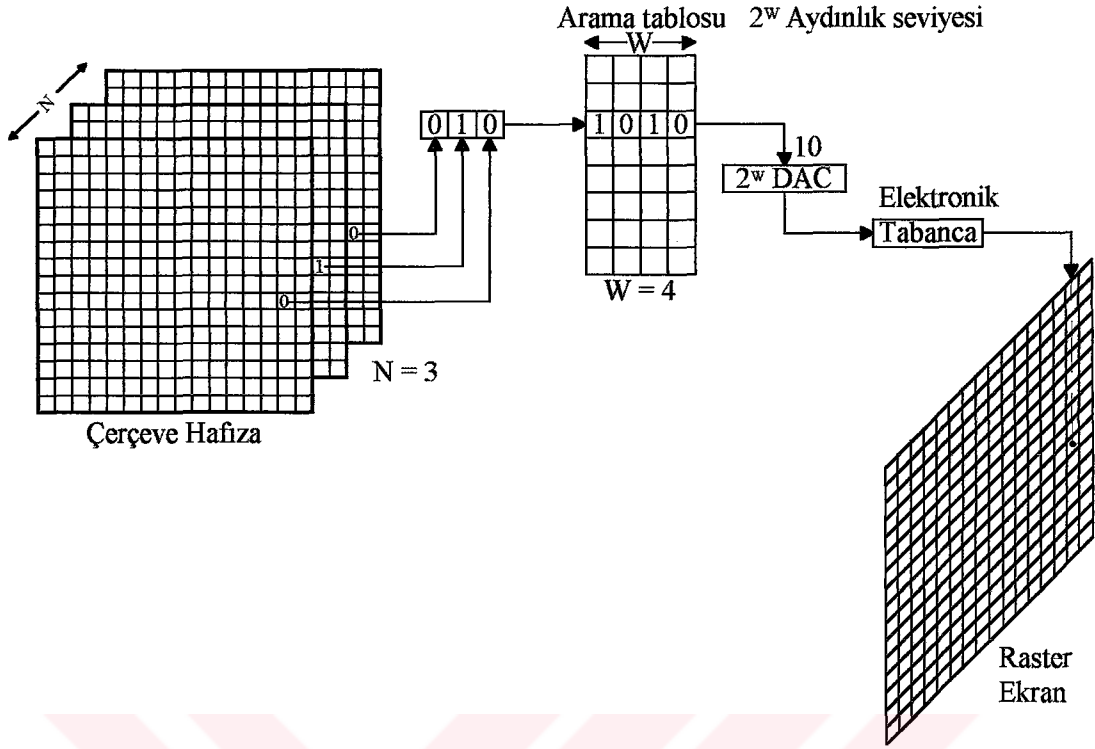
Renkli ve gri tonlu görüntüler bit düzlemi sayısının arttırılması ile elde edilir. Şekil 2-3 N bit düzleminde ve gri tonlamalı çerçeve belleğin kullanımını göstermektedir. Bir beneğin parlaklığı, N bit düzleminin beneğe karşılık gelen

bitlerinin birleştirilmesi ile bulunur. Bulunan bitler bir yazmaçta (register) toplanır. N tane bit düzlemi için 0'dan (karanlık, siyah) 2^N-1 (tam parlaklık, beyaz) arasındaki değerler beneğin parlaklığını ifade eder ve bu değerler DAC tarafından elektronik tabancanın 0 ile maksimum voltaj değerleri arasına ayarlanır. 2^N tane parlaklık seviyesi mevcuttur. Şekil 2-3'te 3 bit düzleminin bulunduğu ve dolayısıyla 8 (2^3) parlaklık seviyesinin bulunduğu bir raster ekranı gösterilmektedir. Verilen örnekte 512x512 boyutlu bir raster ekran için 786,432 ($3 \times 512 \times 512$) bellek bitine ihtiyaç vardır.



Şekil 2-3 N bit düzlemlili ve gri tonlamalı çerçeve bellek

Kullanılabilir parlaklık seviyeleri arttırılmak istendiğinde, ortaya çıkan doğrusal bellek artışı, bir arama tablosu ('palet' olarak da adlandırılır) kullanılarak azaltılabilir. Bkz. Şekil 2-4. Bu düzenekte çerçeve bellekteki bit düzlemlerinden okunan bilgi, arama tablosunun indeksi olarak kullanılmaktadır. Arama tablosundan elde edilen değer ise DAC tarafından elektronik tabancaya uygulanacak voltaj şiddetini verir. N tane bit düzlemi bulunduğunda, arama tablosunda 2^N eleman mevcuttur. Tablodaki her eleman W bit boyundadır. W 'nin değeri N 'den büyük olabilir. Bu durumda 2^W kullanılabilir parlaklık seviyesi bulunur, fakat aynı anda en fazla 2^N farklı parlaklık seviyesi kullanılabilir. Parlaklık değerlerini değiştirmek için arama tablosunun değiştirilmesi gerekir.

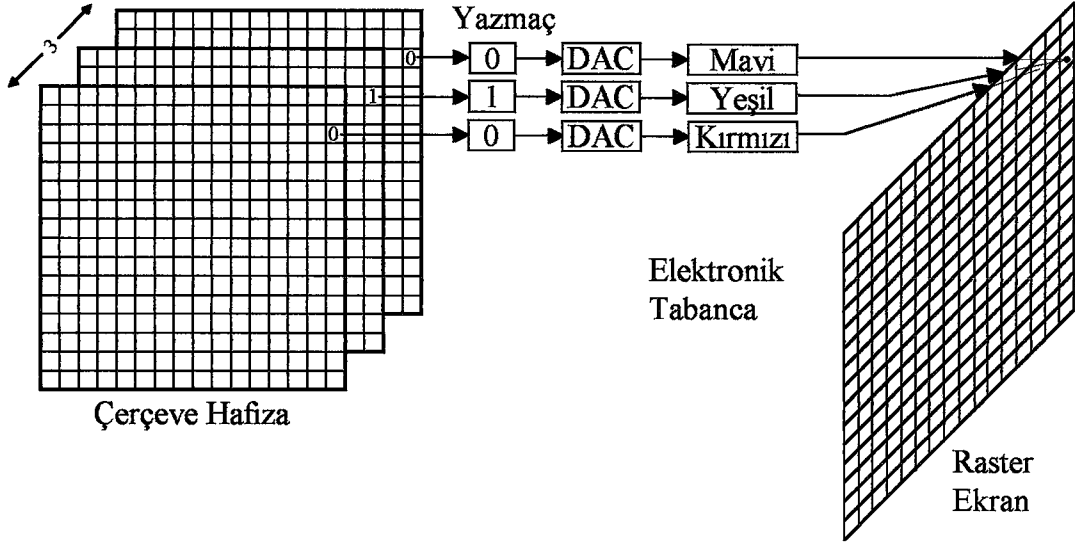


Şekil 2-4 N bit düzlemli W bit boyunda arama tablolu ve gri tonlamalı çerçeve bellek

Asli renk sayısı üç olduğu için renkli çerçeve bellek her renge karşılık gelmek üzere üçlü bit düzlemi kullanılarak elde edilir. Her bit düzlemi bir renk tabancasını yönetir. Bu üç asli renk (kırmızı, yeşil ve mavi) katod ekranda birleştirilerek sekiz renk Tablo 2-1'deki kombinasyonlarda birleştirilerek elde edilir. Şekil 2-5'te basit renkli çerçeve belleğin şeması verilmiştir.

Tablo 2-1 Basit üç bit düzlemli çerçeve bellek renk kombinasyonu

	Kırmızı	Yeşil	Mavi
Siyah	0	0	0
Kırmızı	1	0	0
Yeşil	0	1	0
Mavi	0	0	1
Sarı	1	1	0
Koyu mavi	0	1	1
Mor	1	0	1
Beyaz	1	1	1



Şekil 2-5 Basit renkli çerçeve bellek

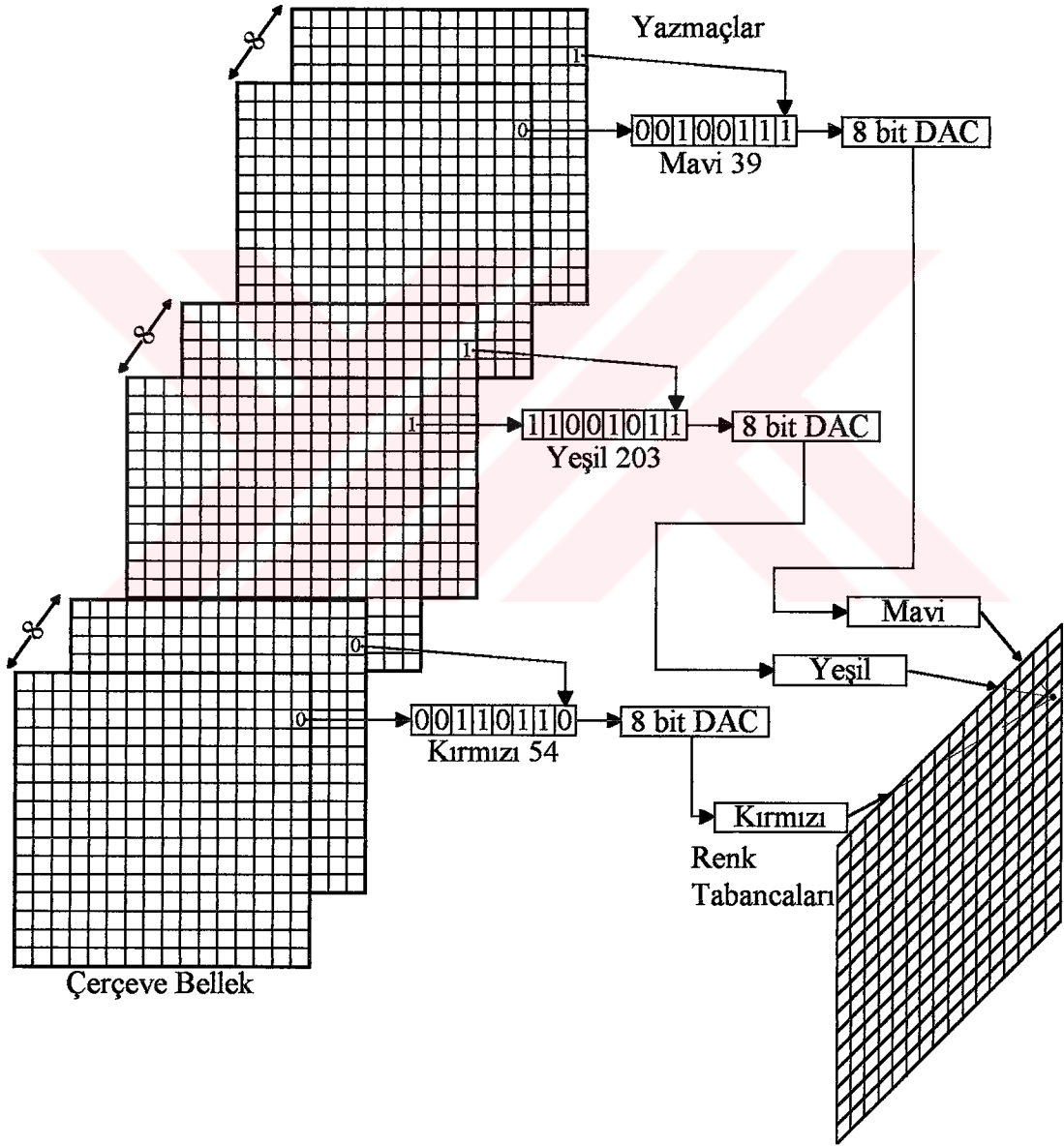
Bit düzlemlerinin her biri üç renk tabancasından birine karşılık gelir. Her renk için 8 bitin kullanıldığı çoklu bit düzlemine sahip, renkli çerçeve bellek sisteminin yapısı Şekil 2-6’da gösterilmiştir. Her renk grubu 256 (2^8) renk seviyesi verir ve renk kombinasyonları 16,777,216’yı [$(2^8)^3 = 2^{24}$] bulur. Buna tam renkli çerçeve bellek denir.

Tam renkli çerçeve bellek, her renkteki bit düzlemi için bir arama tablosu konularak geliştirilebilir. Bu Şekil 2-7’de gösterilmiştir.

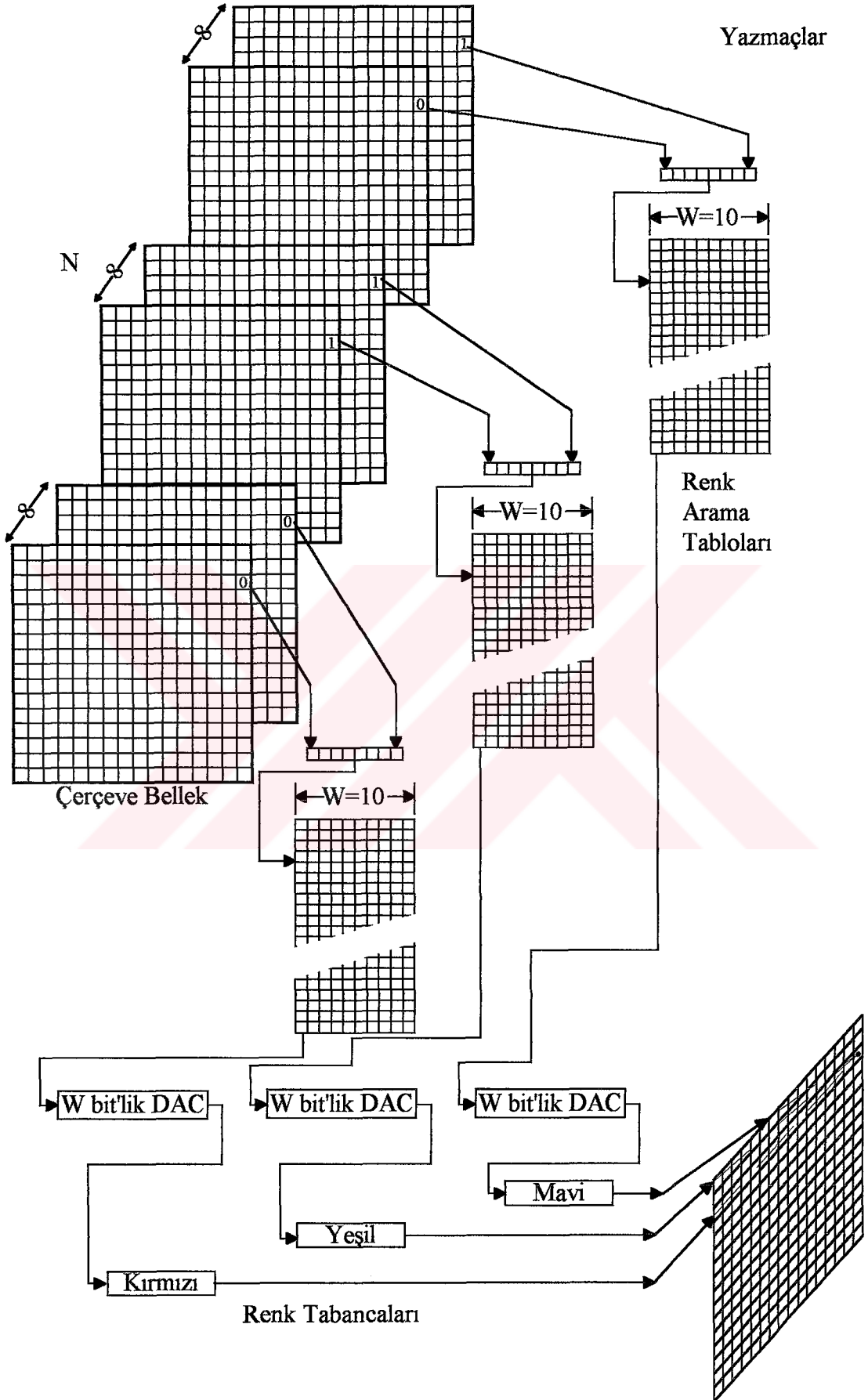
N tane bit düzlemi/renk ve W bit genişliğindeki arama tablosu, $(2^3)^W$ sayıdaki palet elemanından $(2^3)^N$ mümkün renk seçilebilir. 24 bit düzlemli (N=8) çerçeve bellek ve 10 bit genişliğindeki (W=10) arama tablosunda, 1,073,741,824 renk içinden 16,777,216 renk aynı anda kullanılabilir.

Raster ekranlarda çok sayıda benek üzerinde işlem yapılması gerektiğinden dolayı gerçek-zaman (real-time) performansını sağlamak ve kabul edilebilir yenileme oranını sağlamak oldukça zordur. Örnek olarak bir beneğe ortalama erişim zamanının 200 nanosaniye (200×10^{-9}) olduğu 512 x 512 boyundaki bir çerçevedeki tüm beneklere 0.0524 saniyede erişilir ki bu 19 çerçeve (resim) / saniye eder. Bu yenileme hızı kabul edilen sınırın (30 çerçeve / saniye) altındadır. 1024 x 1024 çerçeve bellek 1

milyon bit'in (1 megabit) biraz üzerinde eleman içerir ve 200 nanosaniyelik ortalama erişim zamanı ile tüm beneklere 0.21 saniyede erişilir. Bu da yaklaşık 5 çerçeve/saniye yenileme hızı demektir. 4096 x 4096 çerçeve bellekte 16,78 milyon bit vardır. Aynı erişim süreleri gözönüne alındığında 1 yenileme işlemi için yaklaşık olarak 3 saniye gerekir. Kabul edilebilir yenileme sayısını (30 çerçeve/saniye) sağlamak için ortalama benek erişim süresinin 2 nanosaniye olması gerekir. Bu kısa süre içinde ışık yaklaşık 5 santimetre yol alır.



Şekil 2-6 24 bit düzlemli renkli çerçeve bellek



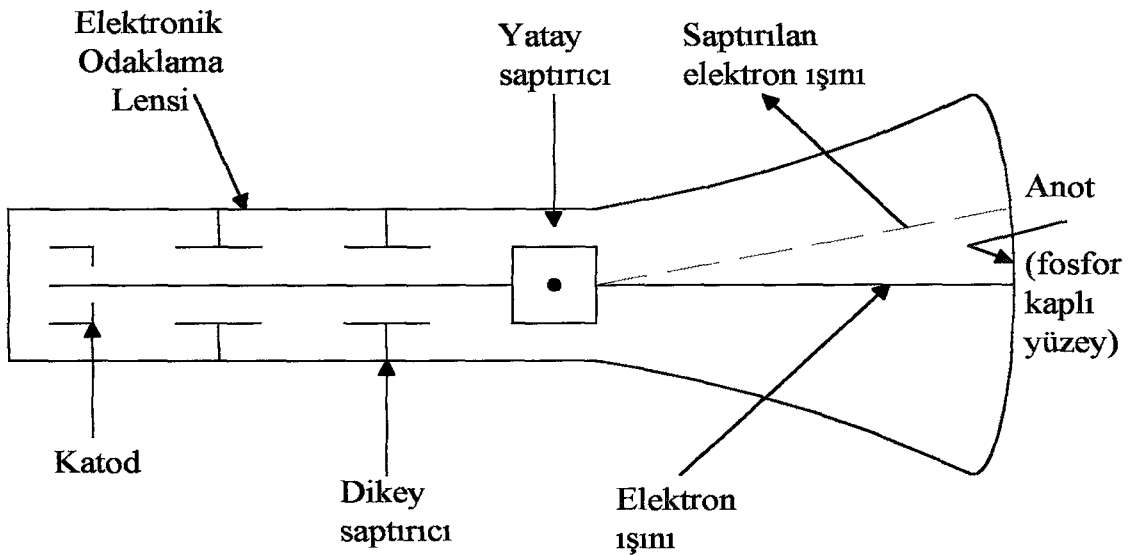
Şekil 2-7 10 bitlik arama tablolı 24 bit düzlemli renkli çerçeve bellek

Raster taramalı ekranlarda gerçek zaman performansı, beneklere 16, 32 veya 64'lük gruplar halinde ve eşzamanlı erişilerek sağlanır. Bit düzlemlerine birbirinden bağımsız olarak ulaşıldığı ve her beneğin en fazla 32 bit ile kadar temsil edildiği durumda bir beneğe erişim süresini 1600 nanosaniye kabul ettiğimizde 512 x 512 ve 1024 x 1024 bitlik çerçeve belleğin kabul edilebilir sınırlar içinde yenilenmesi mümkündür.

2.4 Katod Işın Tübü Temelleri

Çerçeve bellek resimlerin sayısal temsillerini toplamak ve tutmak için kullanılır. Çerçeve bellekle kullanılan en yaygın görüntü cihazı video monitördür. Görüntü cihazlarını anlamak için CRT ve video tekniklerini anlamak gerekir.

CRT Şekil 2-8'de şematik olarak gösterilen video monitörlerde kullanılır. Katod, (negatif olarak yüklü) elektronlar bir ayrışım bulutu içinde (elektronlar aynı yüke sahip oldukları için birbirini iter) kaynayıp ayrışmaya kadar ısıtılır. [3] Bu elektronlar bir pozitif yüklü anot tarafından çekilir. Bu CRT'nin sonunda; içindeki büyük yüzey üzerinde fosfor tabakasıdır. Elektronların kesilmeden devam etmesine izin verilirse parlak bir kızartı ile CRT'nin bütün yüzeyine akar. Bununla birlikte elektron



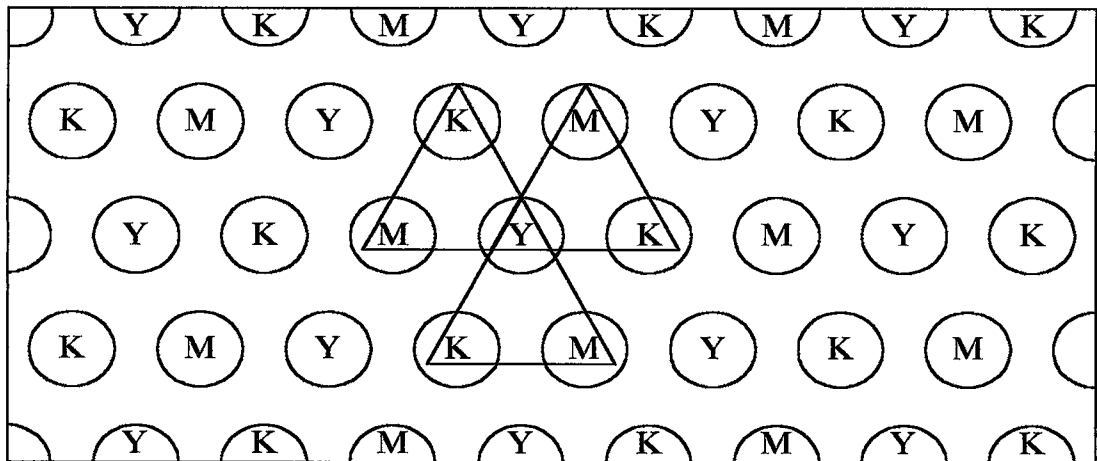
Şekil 2-8 Katod Işın Tüpü

bulutu bir elektron lens ile paralelleştirilmiş dar bir ışın demeti içine odaklanır. Bu noktada odaklanmış elektron demeti CRT'nin merkezinde tek bir parlak nokta oluşturur. Elektron demeti yatay ve dikey saptırma kuvvetlendiricileri vasıtasıyla saptırılır veya merkezin sağ veya soluna ve/veya merkezin altına veya üstüne yerleştirilir.

Bu noktada line drawing displayler, storage, refresh ve raster scan ekranlar farklıdır. Line drawing displaylerde elektron demeti keyfi bir pozisyonndan direkt olarak CRT (anot) yüzey üzerinde diğer bir keyfi pozisyona saptırılabilir. Mükemmel bir düz doğru elde edilir. Bunun aksine, raster scan ekranlarda elektron demeti sabit olarak kontrol edilmiş örüntü kümesi içinde saptırılır, Bu örüntü video resmini oluşturur.

2.5 Renkli CRT Raster Scan Temelleri

Renkli CRT raster scan ekranları çalışma prensibi siyah-beyaz CRT'ler ile benzerdir. Renkli CRT'lerde farklı olarak üç temel renk (kırmızı, mavi ve yeşil) için üç renk tabancası kullanılır. Elektronik tabancalar üçgen örüntü halinde bulunur ve ekran yüzeyinde de bunun benzeri renkli forfor örüntüleri bulunur. Bkz. Şekil 2-9. Bir elektron atışının doğru renkteki fosfor yüzeye çarpması (örnek olarak kırmızı tabancanın yalnızca kırmızı fosfor noktasına) tabancalar ile yüzeydeki noktalar arasında



konan delikli bir metal ızgara ile sađlanır. Buna yansıma perdesi (shadow mask) ve bu tür CRT ekranlara da yansıma perdeli CRT ekranlar adı verilir. Yansıma perdesi üzerindeki delikler ekran yüzeyi ile aynı üçgensel örüntüye sahip olarak düzenlenmiştir. Delikler arasındaki mesafeye adım (pitch) denir.

Bilgisayar ekranı üzerinde resim oluşturmada ekranların teknik yapısı hakkında verilen bu kısa bilgi sonucunda ekranın pixel (benek) denilen işlenebilir noktalardan oluşan bir ortam olduğu ve resmin bu noktaların yan yana veya alt alta bulunmasıyla elde edildiğini şeklinde özetlenebilir. Bu bölümü takip eden bölümlerde ekran üzerinde resim oluşturmada uygulanabilecek temel dönüşümlerin matematiksel yapıları hakkında bilgi verilecektir.



BÖLÜM 3. İKİ BOYUTLU GRAFİKSEL DÖNÜŞÜMLER

Bu bölümde bilgisayar grafiklerinin matematiksel temelleri incelenerek, noktalar veya çizgilerin gösterime hazırlanırken uygulanan dönüşümler (ölçekleme, döndürme, taşıma, aynalama) için gerekli matematiksel teknikler anlatılmaya çalışılacaktır.

3.1 Noktaların Gösterimi

İki boyutlu gösterimde, bir noktanın bulunduğu koordinatlar, iki boyutlu bir matrisle gösterilir. Bu iki değer matrisin bir satırında iki sütun değeri (Bu bölümde satır matris düzeninde gösterim tercih edilecektir.) olarak bulunur.

$$[X] = [\begin{matrix} x & y \end{matrix}]$$

Üç boyut ise 1 x 3 matris ile gösterilir.

$$[X] = [\begin{matrix} x & y & z \end{matrix}]$$

Bir nokta serisinin bilgisayarda kullanımı, bu matrislerin satırları veya noktaların bir dizisi olarak yapılır. Dolayısıyla ifade edilmesi gereken nokta sayısı, aynı zamanda matrisin satır sayısını verir.

Noktaların belirlenmesinde en sık kullanılan yöntem kafesleme yöntemidir. Bu yöntemde herhangi bir yüzeyi oluşturan uç noktalar belirlenir ve bu noktalar üzerine dönüşümler uygulanır. Noktalar arasına çizilen doğru ve yaylar ile yüzey elde edilir. Çizim sırasında iki uç nokta arasındaki (noktaları gösteren matrisin iki satırı) ilişkinin ayrıca bilinmesi gerekir.

3.2 Çevrimler ve Matrisler

Matris elemanları çeşitli düzenlerde bulunabilirler. Örnek olarak bir seri, bir ağ veya bir eşitliğin katsayıları olabilir. Matris cebri bu matrisler üzerinde yapılabilecek işlemleri belirler. Bir obje üzerinde yapılabilecek işlemler $[A][T]=[B]$ tarzında ifade edilebilir. $[A]$ objenin geometrik işlemden önceki noktalarından oluşan bir matrisi, $[B]$ objenin geometrik işlemden sonraki konumunu belirtir.

$[T]$ matrisinin diğer bir yorumu da geometrik işlemci olmasıdır. Bundan sonraki kısımlarda özel bir takım durumlardan bahsedilecektir.

3.3 Noktaların Dönüşümü

P noktasının koordinatlarını veren $[x \ y]$ matrisi ile 2×2 boyutlarındaki genel dönüşüm $[T]$ matrisini çarptığımızı düşünelim:

$$[X][T] = [x \ y] \begin{bmatrix} a & b \\ c & d \end{bmatrix} = [(ax + cy) \ (bx + dy)] = [x^0 \ y^0] \quad (3-1)$$

Bu matematiksel gösterim x ve y koordinatlarının x^0 ve y^0 koordinatlarına dönüşümünü verir. ($x^0 = (ax + cy)$ ve $y^0 = (bx + dy)$) Bu aşamada P noktasının x^0 ve y^0 koordinatlarındaki dönüşümden sonraki halini inceliyoruz. Birkaç özel durumu inceleyelim:

$$[X][T] = [x \ y] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [x \ y] = [x^0 \ y^0] \quad (3-2)$$

$a = d = 1$ ve $b = c = 0$ durumunda $[T]$ matrisi birim matris olur ve çarpımdan sonra P koordinatlarında bir değişiklik olmaz. Matris cebirinde bir matrisin birim matris ile çarpımı matrisi 1 ile çarpmak ile aynı olduğundan bu beklenen bir sonuçtur.

Şimdi $d = 1$ ve $b = c = 0$ kabul edelim;

$$[X][T] = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} a & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} ax & y \end{bmatrix} = \begin{bmatrix} x^0 & y^0 \end{bmatrix} \quad (3-3)$$

Çarpım sonucu sadece $x^0 = ax$ sonucu değişir ki, bu x eksenindeki bir ölçek değişikliği sonucunu üretir. Bkz. Şekil 3-1a. $b = c = 0$ durumunda;

$$[X][T] = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix} = \begin{bmatrix} ax & dy \end{bmatrix} = \begin{bmatrix} x^0 & y^0 \end{bmatrix} \quad (3-4)$$

Her iki eksen (x ve y) üzerinde ölçekleme sonucunu üretir. Eğer $a \neq d$ ise ölçekleme oranları aynı değildir. Eğer $a = d > 1$ ise eşit ölçekli büyültme, eğer $0 < a = d < 1$ ise eşit ölçekli bir büzüşme meydana gelir. Bkz. Şekil 3-1b.

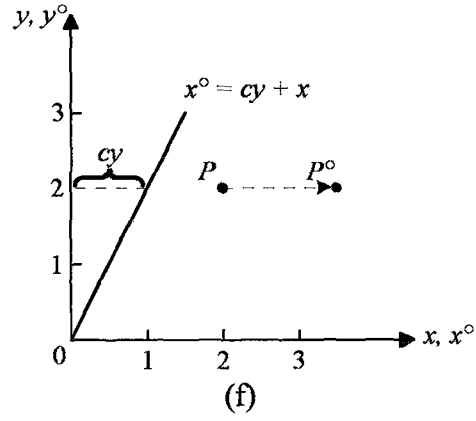
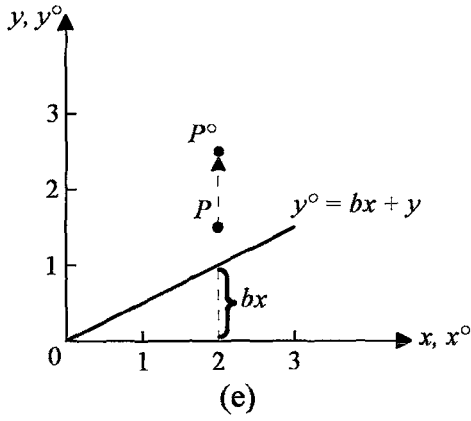
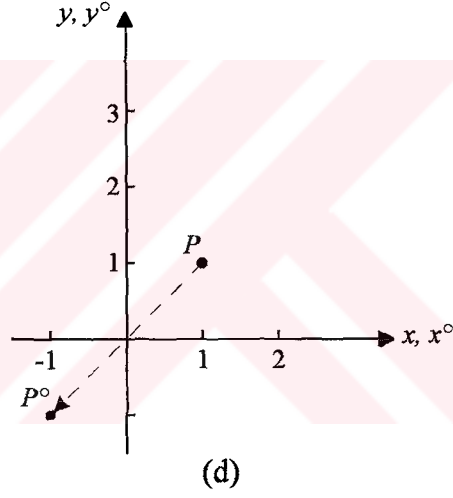
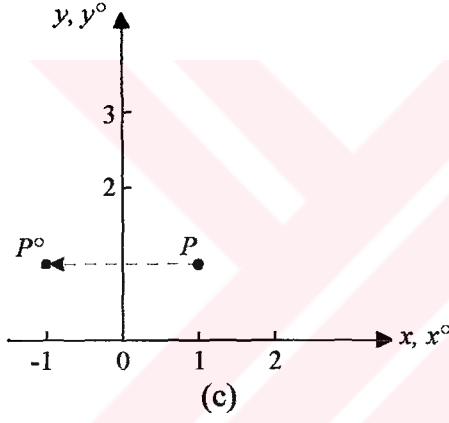
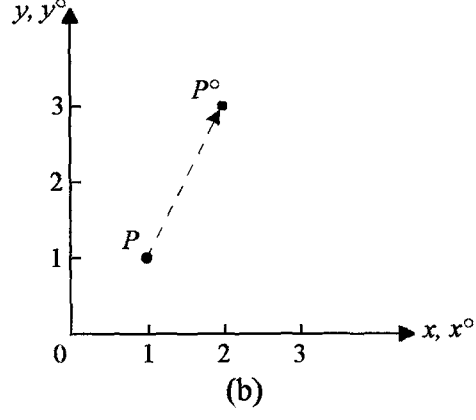
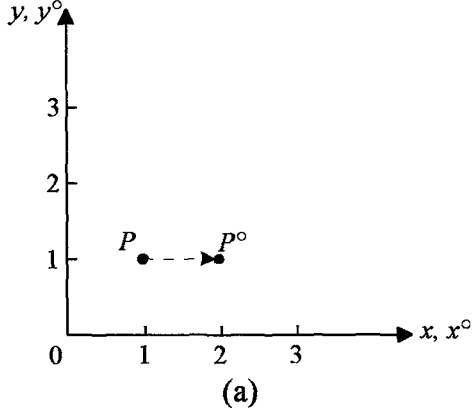
Eğer a ve/veya d negatif ise, eksen üzerinde yansıma değeri elde edilir. Bunu görmek için $b = c = 0$, $d = 1$ ve $a = -1$ kabul edelim;

$$[X][T] = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -x & y \end{bmatrix} = \begin{bmatrix} x^0 & y^0 \end{bmatrix} \quad (3-5)$$

Bu durumda Şekil 3-1c'de görüldüğü gibi y eksenine göre bir yansıma elde edilir. Eğer $d = -1$ olursa o zaman da x eksenine göre yansıma bulunur. Eğer $b = c = 0$ ve $a = d < 0$ ise orjine göre yansıma elde edilir. Bkz. Şekil 3-1d. Görüldüğü gibi ölçekleme ve yansıma işlemleri matrisin esas diyagonalı üzerinde yapılmaktadır.

Şimdi matrisin esas diyagonalindeki elemanları sabit tutarak, talî diyagonal üzerindeki elemanları değiştirelim. İlk önce $a = d = 1$ ve $c = 0$ olduğu durumda,

$$[X][T] = \begin{bmatrix} x & y \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} x & (bx + y) \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} x^0 & y^0 \end{bmatrix} \quad (3-6)$$



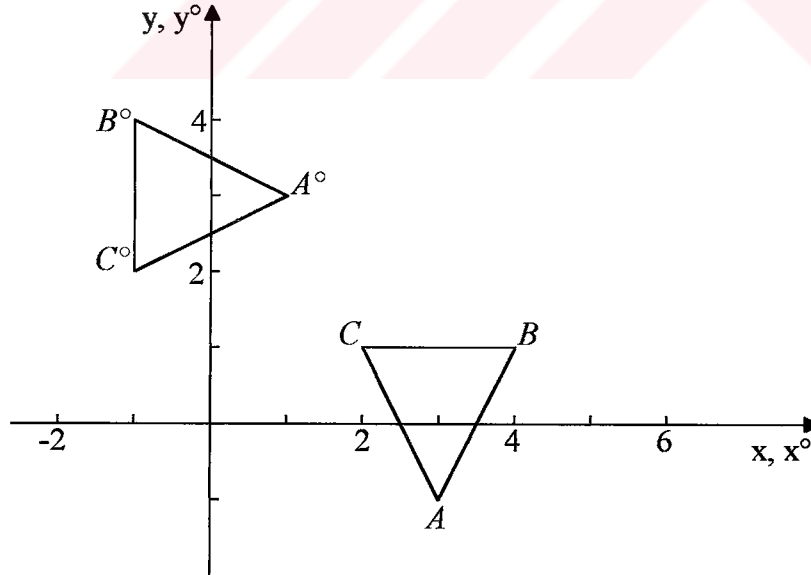
Şekil 3-1 Noktaların dönüşümü

Dikkat edilirse P noktasının x koordinatında bir deęişiklik olmadığı halde, y^0 noktası orjinal koordinatlara göre doğrusal olarak deęişmiştir. Bu etki büzme (shearing) olarak adlandırılır. Bkz. Şekil 3-1e. Benzer olarak $a = d = 1$ ve $b = 0$ iken y koordinatına baęlı olarak büzüşme işlemi gerçekleşir. Bkz. Şekil 3-1f. Yukarıdaki nedenden dolayı tali eksen büzüşme eksenini olarak da adlandırılır.

Son olarak 2×2 boyutundaki genel dönüşüm matrisi ile ilgili özel bir durum; orijin noktasının formül (2-1)'e uygulanmasıdır ki: bu durumda dönüşüm matrisinin her durumu için dönüşüm sonucu deęişmez olacaktır. Bu kısıtlama homojen koordinatların kullanılması ile giderilir.

3.3.1 Düz Çizgilerin Dönüşümü

Düz çizgi, başlangıç ve bitiş noktalarını gösteren iki pozisyon vektörü ile ifade edilebilir. Dönüşüm işlemleri bu iki nokta vektörüne uygulanır. Gerçek çizgi oluşturma işlemi ise kullanılan grafik cihazına göre farklılıklar gösterir.



Şekil 3-2 Döndürme

3.4 Döndürme

Şekil 3-2’de gösterilen ABC üçgenini orjin etrafında saat yönünün tersine 90 derece çevirmek için;

$$[T] = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

dönüşümünü kullanarak, üçgenin köşe noktalarının x ve y koordinatlarını bulunduran 3×2 boyutlu bir matris ile çarpıldığında

$$\begin{bmatrix} 3 & -1 \\ 4 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ -1 & 4 \\ -1 & 2 \end{bmatrix}$$

$A^\circ B^\circ C^\circ$ üçgeni elde edilir. Orjin etrafında 180° döndürme işlemi için dönüşüm matrisi;

$$[T] = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

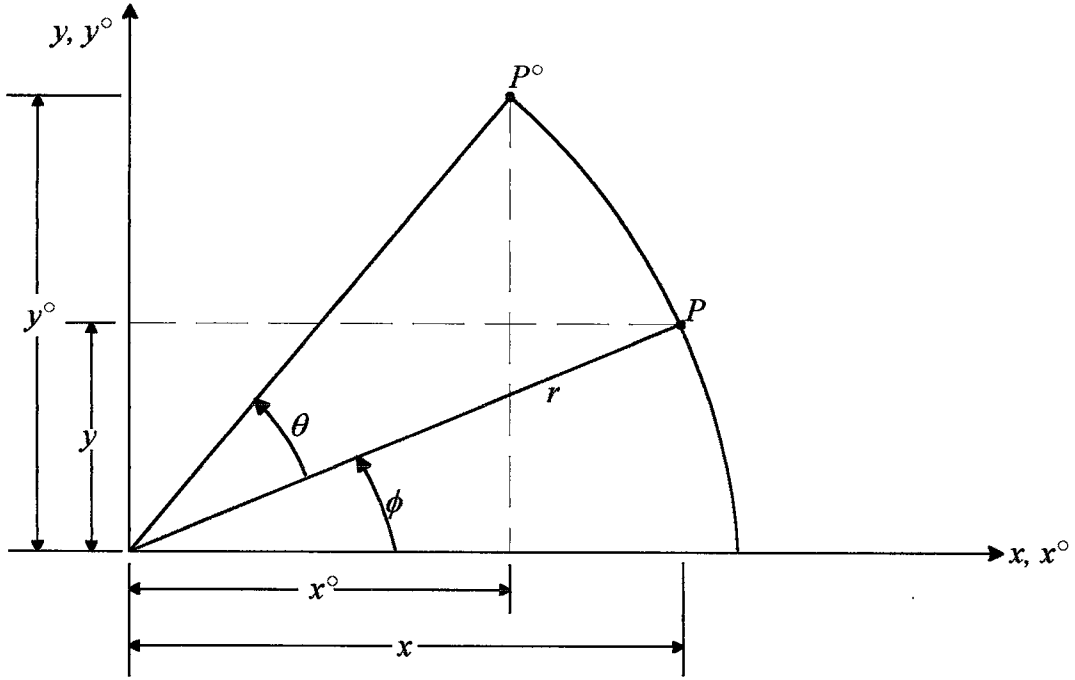
ve orjin etrafında 270° derece döndürme işlemi için;

$$[T] = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

dönüşüm matrisleri kullanılır. Orjin etrafında 0° veya 360° döndürmeye karşılık olan dönüşüm matrisi ise;

$$[T] = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

birim matristir.



Şekil 3-3 Durum vektörünün döndürülmesi

Döndürme işlemi bir noktayı bulunduğu yerden alarak, merkezi $[0, 0]$ noktası olan bir dairenin çevresi üzerinde dönüş açısına göre hareket ettirmekle elde edilebilir. Şekil 3-3’de gösterilen P noktasını orjin etrafında keyfi bir açıyla döndürmek için, r uzunluk ve x eksenine ϕ açısına sahip P durum vektörü θ açısıyla döndürülerek P^0 elde edilir.

P ve P^0 vektörlerini yazarsak;

$$P = \begin{bmatrix} x & y \end{bmatrix} = \begin{bmatrix} r \cos \phi & r \sin \phi \end{bmatrix}$$

ve

$$P^0 = \begin{bmatrix} x^0 & y^0 \end{bmatrix} = \begin{bmatrix} r \cos(\phi + \theta) & r \sin(\phi + \theta) \end{bmatrix}$$

Açı toplama formüllerini ¹ kullanarak P^0 ’ı yeniden yazarsak;

¹ $\cos(\phi \pm \theta) = \cos \phi \cos \theta \pm \sin \phi \sin \theta$
 $\sin(\phi \pm \theta) = \cos \phi \sin \theta \pm \sin \phi \cos \theta$

$$P^0 = \begin{bmatrix} x^0 & y^0 \end{bmatrix} = \begin{bmatrix} x \cos \theta - y \sin \theta & x \sin \theta + y \cos \theta \end{bmatrix}$$

Böylece dönüştürülen değerler;

$$x^0 = x \cos \theta - y \sin \theta \quad (3-7a)$$

$$y^0 = x \sin \theta + y \cos \theta \quad (3-7b)$$

Bunu matris formuna çevirecek olursak;

$$[T] = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$$

orjin etrafında keyfi bir θ açısıyla döndürme için kullanılabilecek dönüşüm matrisi

$$P^0 = \begin{bmatrix} x^0 & y^0 \end{bmatrix} = \begin{bmatrix} r (\cos \phi \cos \theta - \sin \phi \sin \theta) & r (\cos \phi \sin \theta + \sin \phi \cos \theta) \end{bmatrix}$$

elde edilir. Pozitif açılı döndürme işlemleri Şekil 3-3'de görüldüğü üzere saat yönünün tersine olur. P^0 noktasını eski konumuna döndürmek için

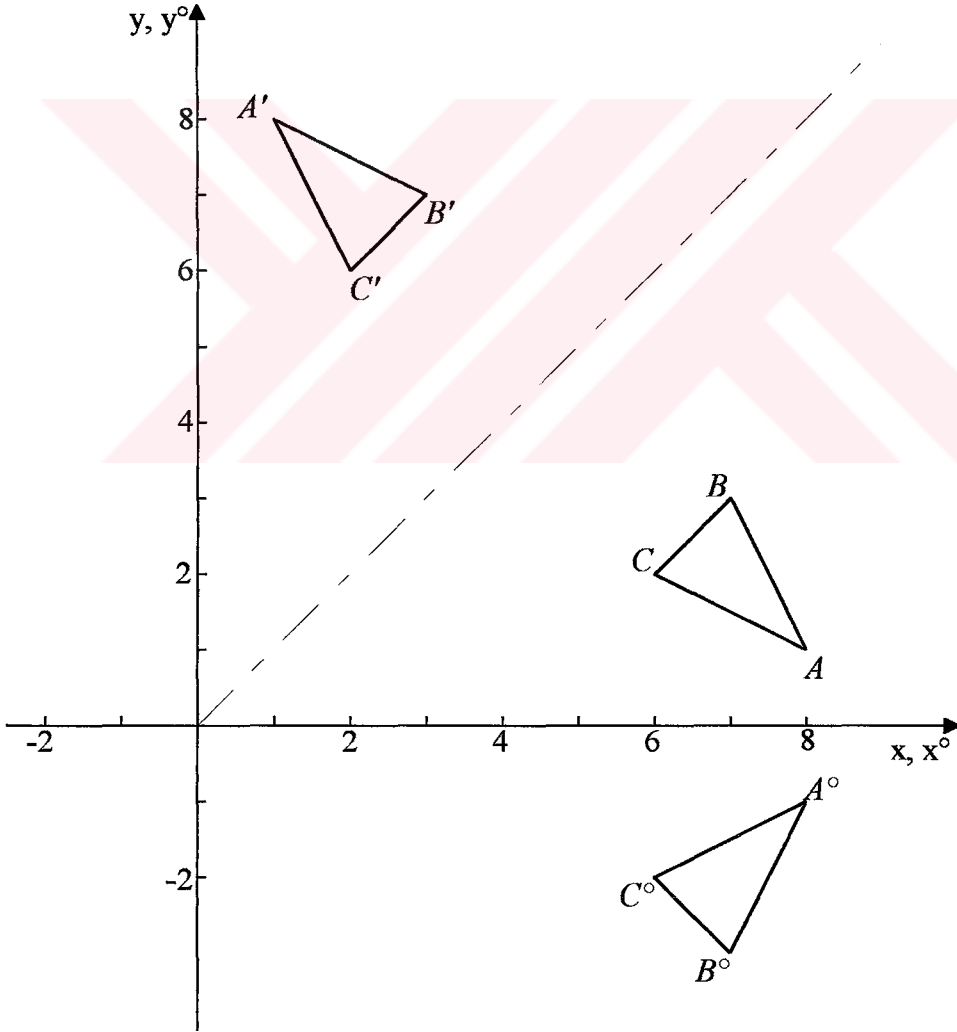
$$[T]^{-1} = \begin{bmatrix} \cos(-\theta) & \sin(-\theta) \\ -\sin(-\theta) & \cos(-\theta) \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (3-8)$$

$[T]$ matrisinin transpozesi olan $[T]^{-1}$ dönüşümü kullanılır ki bu aynı zamanda saat yönünde dönüşümü de ifade eder.

3.5 Yansıma

Yansıma işlemi temel olarak xy düzleminin üzerinde bulunan üçüncü bir düzlem üzerinde 180° döndürme ve sonucu tekrar xy düzlemi üzerine aktarma işlemidir. Şekil 3-4’de DEF üçgeninin iki yansıması görülmektedir. $y=0$ olduğunda x eksenini üzerindeki yansıma için:

$$[T] = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (3-9)$$



Şekil 3-4 Yansıma

dönüşümü uygulandığında $D^\circ E^\circ F^\circ$ noktalarının yeni durumu

$$\begin{bmatrix} 8 & 1 \\ 7 & 3 \\ 6 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 8 & -1 \\ 7 & -3 \\ 6 & -2 \end{bmatrix}$$

$x=0$ olduğunda y eksenini üzerindeki yansıma için uygulanacak dönüşüm

$$[T] = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad (3-10)$$

$y=x$ doğrusu üzerinde yansıma için uygulanacak dönüşüm

$$[T] = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (3-11)$$

kullanılarak D'E'F' noktalarının yeni durumu

$$\begin{bmatrix} 8 & 1 \\ 7 & 3 \\ 6 & 2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 8 \\ 3 & 7 \\ 2 & 6 \end{bmatrix}$$

olur. Aynı şekilde $y=-x$ doğrusu üzerinde yansıma için

$$[T] = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \quad (3-12)$$

dönüşüm matrisi kullanılır uygulanır. Bu yansıma matrislerinin determinanı -1'e eşittir. Dönüşüm matrisinin determinanı -1'e eşitse dönüşüm sonucu tam bir yansıma elde edilir.

3.6 Ölçekleme

Ölçekleme matrisin aslı köşegeni üzerindeki her iki terimin büyüklüğü ile kontrol edilir. Örneğin;

$$[T] = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

operatörü kullanılarak bir üçgen orjine göre 2 defa büyür ve düzgün ölçekleme (uniform scaling) elde edilir. Köşegen üzerindeki terimlerin eşit olmaması durumunda ise bozulma elde edilir. Dönüşüm matrisinde;

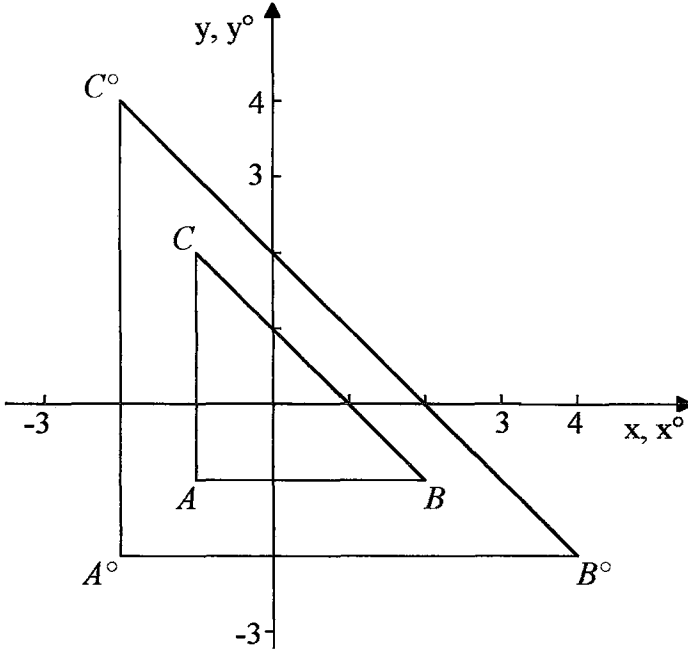
$$[T] = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$a=d$, $b=c=0$ sonucu düzgün ölçekleme elde edilir. $a \neq d$, $b=c=0$ ise düzgün olmayan ölçekleme elde edilir. $a=d>1$ ise düzgün büyüme elde edilir ve şekil daha büyür. Düzgün olmayan ölçekleme ise a veya d 'nin birbirinden farklı ve 1'den küçük veya büyük olmasıyla elde edilir.

Görüntüde yer değiştirmeksizin, tam bir ölçekleme için şeklin orijine merkezlenmiş olması gerekir. Bunun etkisi Şekil 3-5'de gösterilebilir. Şekilde görülen ölçekleme işlemi için;

$$[X^0] = [X][T] = \begin{bmatrix} -1 & -1 \\ 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} -2 & -2 \\ 4 & -2 \\ -2 & 4 \end{bmatrix}$$

dönüşümü uygulanmıştır ve yukarıdaki sonuç matrisi bulunmuştur.



Şekil 3-5 Homojen (düzgün) ölçekleme

3.7 Birleşik Dönüşümler

Köşe noktaları tanımlayan pozisyon vektörleri üzerine, önceki bölümlerde tartışılan dönüşüm işlemleri uygulanarak yüzeyin pozisyonu ve şekli kontrol edilir. İstenilen bir yönelim birden fazla dönüşüm gerektirebilir. Örnek olarak bir yüzeyin önce büyütülmesi (ölçeklenmesi), daha sonra yansıtılması ve son olarak da döndürülmesi gibi. Burada ardışık olarak üç dönüşüm işleminin uygulanması gerekir. Matris çarpımı değişebilir (komütatif) olmadığı için dönüşüm uygulamasının sırası önemlidir.

Matris çarpımının değişmeme özelliğinin etkisini göstermek için $[x \ y]$ durum vektörü üzerinde döndürme ve yansıtma işlemi yapıldığını düşünelim: $[T_1]$ ile 90° döndürme ve bunu takiben $y = -x$ doğrusu boyunca yansıma işlemlerini ardışık olarak yaparsak;

$$[x^*] = [x'] [T_2] = [-y \ x] \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} = [-x \ y]$$

ve daha sonra

$$[x'] = [x][T_1] = [x \ y] \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = [-y \ x]$$

Diğer taraftan yansıtmayı döndürme takip ederse

$$[x'] = [x][T_2] = [x \ y] \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} = [-y \ -x]$$

ve

$$[x^*] = [x][T_1] = [-y \ -x] \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = [x \ -y]$$

elde edilir. Sonuçlar farklıdır. Matris dönüşümlerinin uygulama sırasını belirlemek önemlidir. Diğer önemli bir nokta yukarıdaki sonuç ve aşağıda verilen örnek ile gösterilir.

3.8 Dönüşümler ve Homojen Koordinatlar

Bir çok dönüşümde önceki bölümlerde anlatılan 2x2 boyutlu dönüşüm matrisi kullanılır. Dikkat edilirse bütün bu dönüşümlerde koordinat sisteminin orjini değişmez. Bununla birlikte 2 boyutlu düzlemde her noktayı dönüştürebilmek için orjinin konumunu değiştirmek (ötelemek, taşımak) gerekir. Bu iki boyutlu düzlemde herhangi bir noktayı orjine ötelemekle mümkündür. Örnek olarak;

$$x^o = ax + cy + m$$

$$y^o = bx + dy + n$$

Faka 2x2 boyutlu dönüşüm matrisini kullanarak m , n taşıma sabitlerini kullanabilmek mümkün değildir.

Bu zorluk, homojen koordinatların tanıtılması ile giderilir. Homojen olmayan $[x \ y]$ durum vektörlerinin homojen koordinatları $[x' \ y' \ h]$ durum vektörüyle ifade edilir ve $x=x'/h$ ve $y=y'/h$, h ise gerçel bir sayıdır. $h=0$ özel bir durum olduğuna dikkat ediniz. İlerleyen bölümlerde xy düzleminde bulunan bir durum vektörünü, homojen koordinatlar ile ifadesinde $[x \ y \ 1]$ yolu seçilecektir. Bununla birlikte tek bir homojen koordinat sistemi olmayıp, genel homojen koordinat sistemi $[hx \ hy \ h]$ şeklindedir. Örnek olarak $[8 \ 4 \ 2]$, $[4 \ 2 \ 1]$ ve $[12 \ 6 \ 3]$ koordinatlarının hepsi $x=4$ ve $y=2$ noktasını tanımlar.

Genel 3x3 boyutlu dönüşüm matrisi;

$$[T] = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ m & n & 1 \end{bmatrix} \quad (3-13)$$

Üst sol 2x2 alt-matrisin elemanları a, b, c, d daha önceki bölümlerde anlatılan aynı etkiye sahiptir. m, n sırasıyla x, y yönlerinde öteleme faktörleridir.

$$[x^* \ y^* \ 1] = [x \ y \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix} = [x+m \ y+n \ 1] \quad (3-14)$$

Formülde de görüldüğü gibi iki boyuttaki, orjin dahil her nokta, bu matris kullanılarak ötelenebilir.

3.9 Keyfi Bir Nokta Etrafında Döndürme

Önceki bölümlerde, orjin etrafında dönmeleri inceledik. Homojen koordinatlar, orjin dışındaki noktalar etrafında döndürme işlemini sağlayan bir mekanizma sağlamıştır. Keyfi bir nokta etrafında döndürme işlemi, ilk olarak noktayı orijine

öteleme, daha sonra gereken döndürme işlemini gerçekleştirme ve son olarak da noktanın eski pozisyonuna ötelenmesidir.

$$\begin{bmatrix} x^o & y^o & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -m & -n & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix} \quad (3-15)$$

Matris çarpımları hesaplandığında;

$$\begin{bmatrix} x^o & y^o & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ \left\{ \begin{matrix} -m (\cos \theta - 1) \\ +n \sin \theta \end{matrix} \right\} & \left\{ \begin{matrix} -n (\cos \theta - 1) \\ -m \sin \theta \end{matrix} \right\} & 1 \end{bmatrix} \quad (3-16)$$

Bir örnekle açıklayacak olursak: merkezi [4 3] olan bir yüzeyin saat yönünde 90° döndürülmek istendiğini varsayalım. Döndürme için gerekli dönüşüm matrisi;

$$\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

ile orjin etrafında döndürme işlemi gerçekleştirilir. Döndürmenin, yüzeyin merkezine göre olmasını sağlamak amacıyla ilk önce;

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -4 & -3 & 1 \end{bmatrix}$$

dönüşüm matrisi kullanılarak, yüzeyin merkez noktasının orjine ötelenmesi sağlanacak, daha sonra döndürme işlemi yapıldıktan sonra ters öteleme matrisi kullanılarak merkez noktası eski pozisyonuna ötelenecektir. Bu işlemleri yazarsak;

$$\begin{bmatrix} x^o & y^o & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -4 & -3 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 4 & 3 & 1 \end{bmatrix}$$

elde edilir. Dönüşüm matrislerinin çarpımı hesaplandığında;

$$\begin{bmatrix} x^o & y^o & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 7 & -1 & 1 \end{bmatrix}$$

birleşmiş dönüşüm matrisi bulunur.

Objenin noktaları üzerine uygulanabilecek geometrik işlemlerin ardından, verilerin istatistiksel değerlendirmesi bölümünde, günlük hayatta sıkça kullanılan ve uygulama programının bir bölümünü işgal edecek olan, bir veri kümesinin karakteristik değerlerinin incelenmesi üzerinde kısaca durulacaktır.

BÖLÜM 4. VERİLERİN İSTATİSTİKSEL DEĞERLENDİRİLMESİ

Bu bölümde, uygulamada kullanılan ortalama, dağılma ve istatistiksel ilişkiler konularıyla ilgili ayrıntılı anlatım yapılacaktır.

4.1 Ortalamalar

İstatistik açıdan bir seri veya grubu temsil etmeye yarayan en önemli araçlardan biri ortalamalardır. [4] Ortalama, bir serinin en küçük değeri ve en büyük değerleri arasında bir değere sahiptir. Serinin en küçük değerini x_i , en büyük değerini x_n ve ortalamayı M ile ifade edecek olursak ortalama

$$x_i < M < x_n$$

eşitsizliğini sağlar. Serinin bütün terimleri birbirine eşit olduğunda $x_i = M = x_n$ olur. $M(r)$ ile gösterilen ortalamaların genel formülü

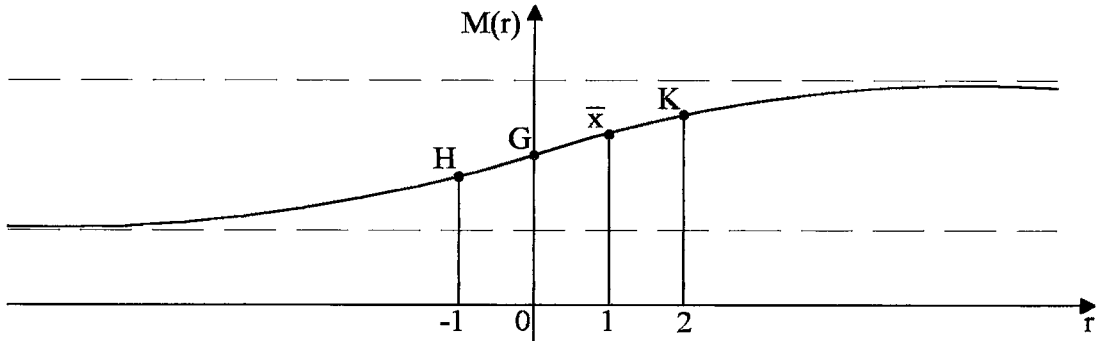
$$M(r) = \left(\frac{\sum_{i=1}^n x_i^r}{n} \right)^{\frac{1}{r}} \quad (4-1)$$

r 'nin değişim aralığı $(-\infty, +\infty)$ arasındadır. r 'nin $-\infty$ değeri alması durumunda ortalama x_i değerine eşit olacaktır. Aynı şekilde r 'nin $+\infty$ değeri alması durumunda ortalama x_n değerine eşit olacaktır. Bkz. Şekil 4-1.

Yukarıda açıklandığı gibi r 'nin değeri $(-\infty, +\infty)$ arasında olmakla beraber uygulamada en çok kullanılan ortalamalar r 'nin birkaç değerine (Bkz. Tablo 4-1) karşılık olan ortalamalardır ve her birinin farklı isimlerle bilinir.

Tablo 4-1 Çok kullanılan ortalamalar

r değeri	Ortalama adı	Formül
-1	Harmonik ortalama	$H = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}}$
0	Geometrik ortalama	$G = \left(\prod_{i=1}^n x_i \right)^{\frac{1}{n}}$
1	Aritmetik ortalama	$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$
2	Kareli ortalama	$K = \left(\frac{\sum_{i=1}^n x_i^2}{n} \right)^{\frac{1}{2}}$



Şekil 4-1 Duyarlı ortalamalar

4.1.1 Aritmetik Ortalama

Gözlenen sayısal değerler toplamının, gözlem sayısına bölünmesi ile elde edilen değere aritmetik ortalama denilir. Bkz. Formül (4-2).

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (4-2)$$

Aritmetik ortalamanın hesaplanmasında kullanılabilecek bir yöntem itibari ortalama yoluyla hesaplamadır. Bu yöntemde, seri toplamını hesaplanmak yerine, bir değeri (Örnek olarak: seri içinden alınacak bir değer; ilk değer, son değer veya en küçük ve en büyük değerlerin ortalaması vb.) ortalama olarak kabul edip, terimlerin bu değere göre farklarının toplamını, gözlem sayısına bölmek ve ortaya çıkan sonucu ilk değer ile toplamaktır. Bu şekilde yapılan hesaplama yöntemine itibari ortalama yoluyla aritmetik ortalama hesabı adı verilir. Bu yöntem gözlem sayısının fazla olduğu durumlarda hesaplama kolaylığı sağlar. (Bilgisayar uygulamalarında sayı büyümesinden doğan taşma hatalarını ve duyarlık kayıplarını büyük ölçüde telafi eder.) Bkz. Formül (4-3).

$$\bar{x} - \bar{x}' = \frac{\sum_{i=1}^n (x_i - \bar{x}')}{n} \quad (4-3)$$

4.1.2 Geometrik Ortalama

Pozitif verilerden meydana gelen bir serinin terimlerini birbirleriyle çarptıktan sonra, çarpımın terim derecesinden kökü alınarak bulunan ortalamaya geometrik ortalama denir. [5] Geometrik ortalaması alınacak seride sıfır veya negatif yüklü bir sayının bulunmaması gerekir.

$$\bar{G} = \sqrt[n]{\prod_{i=1}^n x_i} \quad (4-4)$$

Geometrik ortalama hesaplanırken, çarpım sonucu büyük değerler alabileceği için, değerlerin logaritmalarının aritmetik ortalamasının ters logaritması alınarak yapılan hesaplama yöntemi de aynı sonucu sağlar. Bir serinin geometrik ortalamasının logaritması, seri terimlerinin logaritmalarının aritmetik ortalamasına eşittir.

$$\log G = \frac{1}{n} (\log x_1 + \log x_2 + \dots + \log x_n) \quad (4-5)$$

Geometrik ortalama, olayların oransal değişimleri gözlemlenmek istendiğinde tercih edilen bir ortalama çeşididir.

4.1.3 Harmonik Ortalama

Harmonik ortalama her bir terimin tersinin aritmetik ortalamasıdır. Seri terimleri arasında farklı işaretler veya sıfır değeri varsa işlem sonucu hatalı olur ve bir anlam taşımaz. Bkz. Formül (4-6).

$$\bar{H} = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}} = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}} \quad (4-6)$$

Harmonik ortalama, özellikle zamana veya belirli bir birime oranlanan olayların ortalamasını bulmak için kullanılır.

4.1.4 Kareli Ortalama

Kareli ortalama terimlerin karelerinin toplamını, terim sayısına bölmek ve sonucun kare kökünü almak yoluyla hesaplanır. “Kuadratik Ortalama” adıyla da bilinir.

$$\overline{K} = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_n^2}{n}} = \sqrt{\frac{\sum_{i=1}^n x_i^2}{n}} \quad (4-7)$$

Uygulama sahası diğer analitik ortalamalara göre daha kısıtlı olan kareli ortalama, diğer ortalamaların kullanılmadığı durumlarda tercih edilir. Ayrıca ileride görüleceği gibi standart sapmanın hesaplanmasında da kullanılır.

4.2 Dağılım

Ortalamalar, serilerin temsilinde ve analizinde faydalı olmakla birlikte, yeterli olmamaktadır. Bunun nedeni aynı ortalamayı veren iki serinin yapıları, farklılık gösterebilmektedir. Bu bakımdan serilerin ortalamalarından başka, terimlerin dağılımlarının da dikkate alınması gerekir. Dağılımın amacı, terimlerin ortalamaya olan uzaklıklarının ölçülmesidir. Bir başka deyişle, ortalama etrafındaki terimlerin dağılımını gözlemektir.

Dağılım ölçüsünün hesaplanması, serinin bazı terimleri kullanılarak yapılabilir. Serinin en büyük ve en küçük değeri arasındaki farkın bulunduğu, ‘dağılım aralığı’ bu ölçülerden biridir. Ancak daha yaygın olarak kullanılan dağılım ölçüleri serideki bütün terimlerini dikkate alan dağılım ölçüleridir.

Bu bölümde bazı dağılım ölçüleri anlatılacaktır.

4.2.1 Ortalama Sapma

Ortalama sapma, serideki terimlerin aritmetik ortalamadan mutlak sapmalarının ortalamasıdır.

$$O.S. = \frac{\sum |x_i - \bar{x}|}{n} \quad (4-8)$$

Ortalama sapmanın hesaplanmasında, aritmetik ortalama dışındaki ortalamaların kullanıldığı durumlar da vardır.

4.2.2 Standart Sapma

Standart sapma seri terimlerinin ortalamadan ne kadar saptıklarını göstermeye yarayan bir ölçüdür. Standart sapma hesaplanırken, terimlerin aritmetik ortalamadan sapmalarının kareleri toplamını bulunarak, terim sayısına bölünür. Bulunan bu ölçüye varyans adı verilir. Varyansın karekökü alındığında, “ σ ” işareti ile gösterilen standart sapma bulunur. Bkz. Formül (4-9).

$$S.S. = \sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}} \quad (4-9)$$

Standart sapmanın hesaplamasında kullanılan bir başka yöntem de kareli ortalama yoluyla yapılan standart sapma hesabıdır. Formül (4-9)’un açılımıyla elde edilen yeni formülde, kareli ortalamanın karesi ile aritmetik ortalamanın karesi arasındaki fark varyansı ve karekökü alındığında standart sapmayı verir. Bkz Formül (4-10).

$$\sigma = \sqrt{KO^2 - \bar{x}^2} \quad (4-10)$$

4.2.3 Değişim Katsayısı

Buraya kadar açıklanan dağılma ölçülerinin hepsi mutlak değerler üzerinden hesaplanır ve incelenen olayla ilgili birim cinsinden ifade edilirler. Bu bakımdan farklı cinsten hesaplanan mutlak dağılma ölçülerini birbirleriyle doğrudan karşılaştırmak mümkün olmaz. Ayrıca her olayın yapısına göre kullanılan ölçülerdeki rakamların büyüklüğü de mutlak dağılma ölçülerinin büyüklüğü üzerinde önemli bir rol oynar ve karşılaştırılmalarını güçleştirir. İşte bu sakıncaları ortadan kaldırmak amacıyla sıkça kullanılan bir ölçü olan değişim katsayısı kullanılır. Değişim katsayısı bağıl bir dağılma ölçüsüdür ve standart sapmanın aritmetik ortalamaya oranı ile hesap edilerek bulunur ve yüzde olarak tanımlanır. Bkz. Formül (4-11).

$$D = \frac{\sigma}{x} \cdot 100 \quad (4-11)$$

Bu tür hesaplama sayesinde hem olayda belirtilen cins ölçüsü ortadan kalkar, hem de ölçülere ait rakam büyüklüğünün yarattığı gerçek olmayan ölçü büyüklüğü de giderilmiş olur.

4.3 İstatistik İlişkiler

Verilerin istatistiksel değerlendirmesi konusunda şimdiye kadar bir olayın tanımlanması ile ilgili yöntemler tartışılmıştır. Bu bölümde farklı olaylar arasındaki ilişkileri incelemeye yarayan yöntemlerin bazıları incelenecektir.

4.3.1 Bağımlılık İndeksi

İki seri arasında ilişki olup olmadığını anlamak için, serilerin terimden terime değişmelerine bakılır. Terimlerdeki artış aynı yönde ise doğrusal, değil ise ters bir ilişki bulunduğu söylenir. Bir örnek ile açıklarsak:

Tablo 4-2 Bağımlılık indeksinin hesaplanması

x	y	Δx	Δy	$\Delta x.\Delta y$
288	425			
263	292	-	-	+
353	294	+	+	+
357	165	+	-	-
494	182	+	+	+
304	215	-	+	-
425	167	+	-	-
422	150	-	-	+
302	89	-	-	+

Burada “+” işareti bir önceki terime göre artışı, “-” işareti ise azalışı ifade etmektedir. Daha sonra bu işaretler çarpılarak ($\Delta x.\Delta y$) işaretler arasında uyumluluk olup olmadığı gözlenmiştir. Çarpım sonucu “+” ise uygunluk, “-” ise uyumsuzluktan bahsedilir. Bunların arasındaki farkın toplam işarete oranı ise bağımlılık indeksini verir. a “+” işaretlerinin toplam sayısı ve b “-” işaretlerin toplamını gösterdiğinde Bkz. Formül (4-12).

$$\text{Bağımlılık İndeksi} = \frac{a - b}{a + b} \quad (4-12)$$

Örneğin bağımlılık indeksi $5-3/5+3=2/8=0,25$ ’tir.

Bağımlılık indeksinin sakıncalarından biri değişimin şiddetini ifade edememesidir. Diğer bir sakıncada terimler arasında değişme olmadığında (0 olduğunda) ne yapılacaktır.

4.3.2 Bağımlılık Katsayısı

Bağımlılık indeksinin, değişimin şiddetini göstermediği dikkate alınarak, bu sakıncayı ortadan kaldırmak amacıyla değişimin miktarları da gözönüne alınarak

bulunan değere bağımlılık katsayısı denir.

Tablo 4-3 Bağımlılık katsayısının hesaplanması

x	y	Δx	Δy	$\Delta x.\Delta y$
288	425			
263	392	-25	-133	+3325
353	394	+90	+2	+180
418	165	+4	-129	-516
494	182	+137	+17	+2329
304	215	-190	+33	-6270
475	167	+121	-48	-5808
422	150	-3	-17	+51
302	89	-120	-61	+7320

Buna göre bağımlılık katsayısı için bkz. Formül (4-13).

$$\text{Bağımlılık Katsayısı} = \frac{\sum \Delta x.\Delta y}{\sum |\Delta x.\Delta y|} \quad (4-13)$$

Örnek için bağımlılık katsayısı $13204-12594/13204+12594=611/25799=0,023$ 'tür.

4.3.3 Korelasyon Katsayısı

Seri terimleri arasındaki değişimleri ölçmek için en çok kullanılan yöntemlerden biridir. Bu ölçüde terimler arası farklar değil, bunların aritmetik ortalamadan sapmaları esas alınır. Korelasyon katsayısı çeşitli yollardan hesaplanabilmekle beraber bu kısımda, aritmetik ortalamalardan sapmalar kullanılarak hesaplama verilecektir. x bağımlı değişkeni, y bağımsız değişkeni ve r korelasyon katsayısını gösterdiğinde korelasyon katsayısı aşağıdaki gibi hesaplanır;

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}} \quad (4-14)$$

Korelasyon katsayısının (r 'nin) değeri -1 ile +1 arasındadır. Pozitif değerler ilişkinin doğrusal, negatif değerler ise ilişkinin ters oranlı olduğunu gösterir. Ayrıca 1'e yakın değerler ilişkinin kuvvetli, 0'a yakın değerler ise ilişkinin zayıf olduğunu gösterir.



BÖLÜM 5. EĞRİ VE OBJE ÇİZİMİ

Çizim aşamasına gelen bir grafiği mantıksal olarak iki grupta inceleyebiliriz. Bunlardan birincisi istatistiksel amaçlı grafiklerdir: bu tür grafiklerde verilen bir veya daha fazla serinin nokta kümesinin çeşitli şekillerde ifade edilmesi beklenir. Çizgiler, çubuklar, alan grafikleri, pay grafikleri gibi. Bu grafiklerin çizilmesi ile beklenen sonuç, bir serinin dağılımını veya birden fazla serinin arasındaki ilişkiyi göz ile izlemektir. Bu grafikler türlere göre farklı özellikler gösterirler. Örneğin pay grafikleri; bir serinin toplamı gözönüne alındığında, serinin herbir elemanının bütün içindeki payını (yüzdesini) simgeler. Bu nedenle pay grafiklerde ikinci bir seri kullanılması söz konusu değildir, logaritmik ölçekli ve pay grafiklerde negatif değerli eleman bulunmaması gerekir gibi. İlerleyen bölümde grafikler ayrıntılarıyla tartışılacaktır.

İkinci bir tür grafik çizimi ise matematiksel olarak tanımlanmış bir nesnenin çizilmesi ve modellemesine yöneliktir. Burada izlenen yöntem nesnenin üzerindeki herhangi bir noktanın tanımlanması, nesnenin o noktadaki özelliklerinin tanımlanması; nesnenin geçirgenlik ve yansıtıcılık durumu ile nesnenin rengi (kırmızı, mavi ve yeşil renklerden ne kadarının geçirildiği ve ne kadarının yansıtıldığı), ışık kaynağının rengi ve uzaklığı ile gözün (veya kameranın) nesneye olan uzaklıklarının belirtilmesi ile noktanın uygun renge boyanmasıdır. Ancak bir nesne üzerindeki tüm noktaları yukarıda anılan özellikleriyle tanımlamak ve üzerinde işlem yaparak uygun renge boyanması zaman (işlem süresi) ve maliyet (gerekli bellek miktarı) sorunlarını da beraberinde getirmektedir. Bu nedenle genellikle tercih edilen yöntem nesneyi noktalar yerine yüzeyler ile tanımlamak ve yukarıda anılan işlemleri bu yüzeyler üzerinde gerçekleştirmektir.

İlerleyen bölümlerde grafik çeşitleri sırasıyla incelenecektir. Ancak daha önce bu ve bundan sonra gelen bölümde kullanılmak üzere nesneye yönelik programlama ve Visual Basic ile ilgili kısa bilgiler verilecektir.

5.1 Nesneye Yönelik Programlama (Object Oriented Programming)

Bu bölümde daha sonraki bölümlerin daha iyi anlaşılabilmesi için nesneye yönelik programlama (NYP) ile ilgili bazı temel kavramlar, uygulamada kullanılacak grafik tabanlı bir işletim sistemi olan MS Windows ve bir programlama dili olan Visual Basic'den örnekler verilerek anlatılacaktır.

5.1.1 Nesne Nedir ?

Bilgisayar programlarında kullanılan bazı veri tipleri mevcuttur: tamsayılar, kayan noktalı gerçel sayılar, karakter katarları, mantıksal değişkenler, göstergeler (pointers). Bunlar programlamanın temel yapı taşlarıdır.

Bu yapı taşlarının bir araya gelerek kullanıldığı yeni durumlara veri yapıları “data structures” adı verilir: basit veri tiplerinin birleşerek bir araya geldiği yapılar (struct), (pek çok programlama ortamında bulunan) veri tipinin bir sürüsünü bir arada gösteren diziler, yığın (stack) veri yapıları, öncelik ve/veya sonralık ilişkilerini kurmaya yarayan liste veri yapıları “lists veya linked lists”, tek yönlü listeler, çift yönlü listeler, döngülü listeler vs, hiyerarşik sıralamayı gösteren ağaç veri yapıları; ikili ağaç yapıları, çoklu ağaç yapıları, düğümler (node) arası karmaşık ilişkileri gösteren çizge (graph) veya ağ (network) yapıları.

Yukarıdaki veri yapılarında, verilerin bir kümesi (set) diğer bir küme ile benzer özellik taşır. Böyle bir durumda, ancak bir tür nesneyi tanımlamak ve özelliklerini belleklemek mümkündür.

Bilgisayar da kullanılan bir başka veri yapısı ise çerçevelerdir (frame); burada anlatılan, ve bir veri yapısı olarak incelenen çerçeve'nin önceki bölümlerde ki kullanımdan farklı bir anlam taşımaktadır. Çerçeve sistemlerinin üç temel ögesi vardır. Bunlar; [6]

Sınıflar (classes) : Bir sistemde bulunan ve ortak yapıları (türleri) bünyesinde barındıran yapılardır. Her sınıfın tek bir ismi vardır. Sınıfların her biri, bir nesneyi tanımlamaya yarar. Sistemde her sınıfın birden fazla görünüşü bulunabilir. Bir başka deyişle, aynı türe ait birden fazla nesne bulunabilir.

Özellikler (slots, properties) : Sınıfın ortak özelliklerini belirten yapılardır. Aynı türden nesnelerin özellikleri aynı olmakla beraber, bu özelliklerin aldığı değerler farklı olabilirler. Özelliklerden önemli biri de sınıfın hangi üst sınıfa (ebeveyn, parent) ait olduğunu gösteren bilgidir.

İlave özellikler ve/veya alt sınıflar (expansions, subclasses) : Bir türün sadece belirli bir görünüşüne ait özellikler ile o sınıfa bağlı sınıfların isimlerini içeren kısımlardır.

Yukarıda anlatılan yapıda sınıflar bir hiyerarşi ilişkisi içindedir. Üst sınıfa ebeveyn (parent), hiyerarşik olarak onun altında bulunanan sebebiye ise çocuk (child) denir. Bir çocuk sınıfta bulunmayan özellik ebeveyn sınıftan alınarak kullanılabilir. Buna miras alma özelliği denir.

5.1.2 Nesneye Dayalı Programlamadaki Farklar

Yukarıda anlatılan özellikler, nesneye dayalı olmayan herhangi bir programlama dilinde de meydana getirilerek kullanılabilir. Ancak nesnenin özelliklerinden birinin değişmesi durumunda veya nesne üzerinde gerçekleşek bir işlemde, yapılacak kontrol ve icra faaliyetleri bir bütün halinde ardışık olarak çalışan program tarafından gerçekleştirilmelidir.

Nesneye yönelik programlama dillerindeki ayrım da buradan kaynaklanmaktadır. Bu tür dillerde, nesnenin özelliklerindeki bir değişimin yaratacağı değişiklikleri belirlemek ve ona göre davranmak keyfiyeti nesnenin bizi kendisi tarafından (sınıfsal bir davranış şekli olarak) bilinir ve gerçekleştirilir. Aynı zamanda bir nesneye gelen uyarı, gene nesne tarafından algılanır ve bu uyarı karşısında ne

yapacağı programlandığından, bunları icra etmek işi de gene nesne tarafından yapılır.

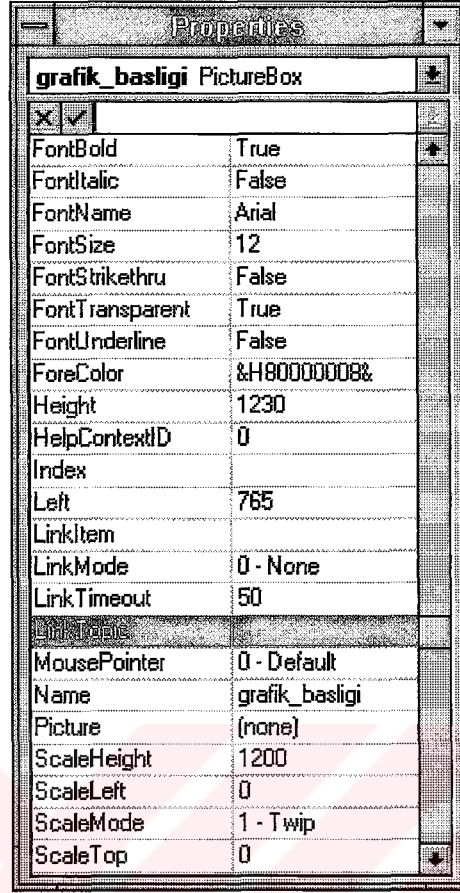
İlerleyen bölümlerde nesneye yönelik bir programlama dili olarak Visual Basic incelenecektir.

5.1.3 Visual Basic ve NYP

Nesneye yönelik programlama dili olarak Visual Basic dilinde kullanılabilen çok sayıda nesne (dildeki adıyla “controls”) bulunmaktadır. Bunların bazıları aşağıdaki gibidir:

Kontrol adı	Türü/Görevi
Form	Bir Windows penceresi açar ve diğer kotrolleri üzerinde barındırır.
PictureBox	Üzerine çizilenleri saklar.
Label	Sabit bir metin görüntüler.
TextBox	Bir metnin ekrandan kendi üzerine girişini sağlar.
CommandButton	Üzerinde iken basıldığında bir program parçasını icra eder.
ListBox	Bir liste içinden seçim yapılmasını sağlar.
ComboBox	Açılır liste içinden seçim yapılmasını sağlar.
Timer	Belirli zaman aralıklarında bir program parçasının tekrarlanmasını sağlar.
CheckBox	Evet/Hayır türü bilgi alınmasını sağlar.
HScrollBar	Yatay kaydırma çubuğu ile belirli değerler arasında bilgi alınmasını sağlar.
VScrollBar	Dikey kaydırma çubuğu ile belirli değerler arasında bilgi alınmasını sağlar.

Bir nesneye üç kısma ayırarak incelemek mümkündür. Bunlar nesnenin özellikleri (properties), nesneye uygulanabilecek olaylar (events), nesne üzerinde kullanılacak yöntemlerdir (methods). Şimdi bu özellikleri sırasıyla inceleyeceğiz.



Şekil 5-1 Resim kutusunun (PictureBox) özellikleri

Özellikler: Önceki bölümde de anlatıldığı gibi nesnenin özelliklerini barındıran bir veri yapısıdır. Şekil 5-1’de PictureBox nesnesinin özelliklerinin bir kısmı, program yazımı (design time) sırasında gösterilmiştir.

Bir nesnenin özelliklerini değiştirme iki zamanda yapılabilir. Bunlar yazılım zamanı (design time) ve çalışma anıdır (run time). Özelliklerin bazıları yazılım zamanında değiştirilebilir ve fakat çalışma anında sadece okunabilir, değiştirilemez. Bazı özellikler hem yazılım ve hem de çalışma sırasında okunabilir ve değiştirilebilir. Bazı özellikler ise yazılım sırasında değiştirilemez (ve hatta şekilde gösterilen özellikler ekranına dahi dahil edilmez), fakat çalışma sırasında okunabilir ve değiştirilebilir.

Her nesnenin farklı özellikler taşıyabileceği daha önceki bölümlerde tartışılmakla beraber, bu her nesnenin farklı özellikler taşıması gereğini ortaya koymamaktadır. Daha açık ifadeyle bir nesnede bulunan bir özellik adı, diğer

nesnelerde de bulunabilir.

Olaylar : Bir nesne, herhangi bir yol ile uyarıldığında çalışacak program parçasının bulunduğu kısımdır. Nesnenin uyarılma yolları, benzer nesneler için aynıdır. Örnek olarak bir TextBox nesnesinin Olayları; Change, Click, DblClick, DragDrop, DragOver, GotFocus, KeyDown, KeyPress, KeyUp, LinkClose, LinkError, LinkNotify, LinkOpen, LostFocus, MouseDown, MouseMove, MouseUp dır. Bu olaylardan bazılarını kısaca açıklayacak olursak, Change olayı TextBox alanına bir bilgi girişi olduğunda, GotFocus nesne seçildiğinde, KeyPress nesne üzerinde iken klavye tuşlarından birine basıldığında, MouseMove nesne üzerinde farenin dolanması sırasında gerçekleşen olaylardır.

Olayların programlanması ile ilgili basit bir örnek konunun daha iyi anlaşılmasını sağlayacaktır; Adı text1 olan bir metin alanı/nesnesi bulunduğunu, bu nesne seçildiğinde içinde göstermekte olduğu bilginin kalın olarak, nesneden uzaklaşıldığında ise metin ince yazıyla gösterilmek istendiğinde, bu iki farklı olayın programlanması şu şekilde olacaktır.

```
Sub text1_GotFocus()  
    text1.FontBold = True  
End Sub
```

```
Sub text1_LostFocus()  
    text1.FontBold = False  
End Sub
```

Yukarıdaki örnekte iki önemli husus vardır. Birincisi nesne seçildiğinde, GotFocus olayı, nesneden uzaklaşıldığında LostFocus olayı, text1 isimli nesne tarafından algılanmakta ve ilgili program parçacıkları nesne tarafından çağrılmaktadır. İkinci dikkat edilmesi gerek husus ise nesnenin özelliklerinin (text1 alanına ait FontBold özelliğinin belirtilerek) değiştirilebilmesidir.

Olaylar bölümünde de her nesne için önceden tanımlanmış olaylar mevcuttur. Aynı tür nesnelerin, olayları da aynı sayıda ve isimdedir. Benzer olaylar farklı nesneler için mevcut olabilir. Örnek olarak Click olayı pek çok nesnede vardır.

Yöntemler : Genel bir amaca yönelik olarak hazırlanmış bir program parçasının, belirtilen bir nesne için çalıştırılmasıdır. Her nesneye uygulanacak yöntemler önceden bellidir ve nesne tarafından isimleri bilinir. Ancak nesne üzerinde gerçekleştirilecek olan işlem, kullanılan yöntem (yöntem kelimesi burada belirli bir program parçasını ifade eder) tarafından bilinir ve icra edilir. Örnek grafik_basligi isimli bir PictureBox nesnesine “İ.T.Ü.” metnini yazdırmak için;

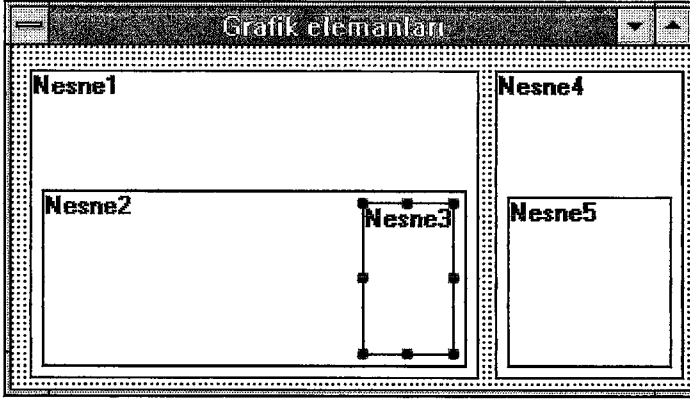
grafik_basligi.print “İ.T.Ü.”

kodu kullanılır. Burada grafik_basligi nesnenin adı, print yöntem ve önünde bulunan metin ise yöntemin argümanıdır. İşlem sırası incelendiğinde grafik_basligi isimli nesne kendine uygulanabilecek yöntemler arasında bir arama işlemi yapar ve print yöntemini argümanı ile birlikte çağırır (call). Print yöntemi nesnenin renk, yazıtipi, boyutu, bulunduğu koordinatlar vs. özelliklerine bakarak verilen metnin nesne üzerinde görünmesini sağlar.

5.1.4 Hiyerarşi İlişkisi

Nesnenin özelliklerinden birinin de hiyerarşi (ebeveynlik) ilişkisinin kurulabilmesi olduğu önceki bölümlerde tartışılmıştır. Visual Basic dilinde hiyerarşik yapılanma gerek yazılım ve gerekse çalışma esnasında sağlanabilir.

Bunu daha iyi anlamak için, yazılım sırasında yapılanmış bir hiyerarşi ilişkisine bakmakta fayda vardır. Şekil 5-2’de grafik elemanları isimli bir form alanı üzerinde beş farklı nesne hiyerarşik olarak tanımlanmıştır. Ancak bu tanımlamada formun doğrudan tanımlı iki çocuğu vardır (Nesne1 ve Nesne4), Nesne1, Nesne2’nin ebeveyni ve Nesne3’de Nesne2’nin çocuğudur. Nesne4 ise Nesne5’in ebeveynidir.



Şekil 5-2 Hiyerarşi (ebeveyn-çocuk) ilişkisinin sağlanması

Visual Basic dilinde, hiyerarşi ilişkisi kurulurken, nesneler üzerine bazı kısıtlamalar da konulmuştur. Örnek olarak, nesneler bir form nesnesi üzerinde bulunmak zorundadır. Bir nesnenin görünebilirliği form alanı ile sınırlıdır bu alana taşamaz, aynı şekilde yukarıdaki örnekte, Nesne3'ün görünürlüğü Nesne2'nin ve Nesne2'nin görünürlüğü Nesne1'in içinde bulunduğu alan ile sınırlıdır.

Hiyerarşiye ilişkisinin getirdiği bir özellik de, üst sınıfa uygulanan bir işlemin, alt sınıfları da etkilemesidir. Tekrar aynı örneğe dönülecek olursa; forma görünmez özelliği verildiğinde, tüm alt sınıflar da görünmez olacaktır, aynı şekilde Nesne4 silindiğinde, onun bir alt sınıfı olan Nesne5'te kendiliğinden (Nesne5 tarafından) silinecektir.

İlerleyen bölümlerde grafik çizimleri ve uygulamada verilen örnekler, bu yapılar üzerine kurulacaktır. Bir sonraki bölümde iki boyutlu bir eğri grafiğinin fonksiyonel olarak çeşitli nesnelere ayrılması ve bu nesnelerin özelliklerinin organizasyonu ayrıntılı şekilde incelenecektir.

5.2 İki Boyutlu Eğri Çizimi

Bir bilgi kümesinin matematiksel anlamda değer kazanabilmesi için bağımsız bir değişken yardımıyla ölçülmesi gerekir. Ölçüm sonucu sayısal bir değere sahiptir ve

her bağımsız değişkene karşılık elde edilen sonuç $[(X_i)_k, f(X_i)_k]$ olarak ifade edilir. Burada i bağımsız değişken sayısını, k gözlem sayısını temsil etmektedir. Gözlem çeşitli şekillerde yapılabilir; x bağımsız değişkeninin belirli bir değer aralığı için bir fonksiyonun aldığı değerler, bir ölçüm cihazıyla tespit edilen ve sayısallaştırılan değerler kümesi vs.

Elde edilen bilgi kümesinin daha iyi anlaşılmasını sağlayan yollardan biri kümenin grafiksel gösterimidir. Grafik çiziminin bilgisayar yardımıyla yapılması işleminden önce, grafik elemanlarının ve özelliklerinin incelenmesi gerekir. Bu bölümde önce grafik elemanları tek tek ele alınacak, daha sonra da birden fazla sayıdaki grafiğin ekran üzerinde tanımlanması nesneye yönelik programlama çerçevesinde incelenecektir.

5.3 Grafik Elemanları

İki boyutlu grafik çiziminde 5 ana eleman mevcuttur. Bunlar grafik başlığı, dikey eksen başlığı, yatay eksen başlığı, seri başlıkları ve grafik çizim alanlarıdır. Bkz. Şekil 5-3.

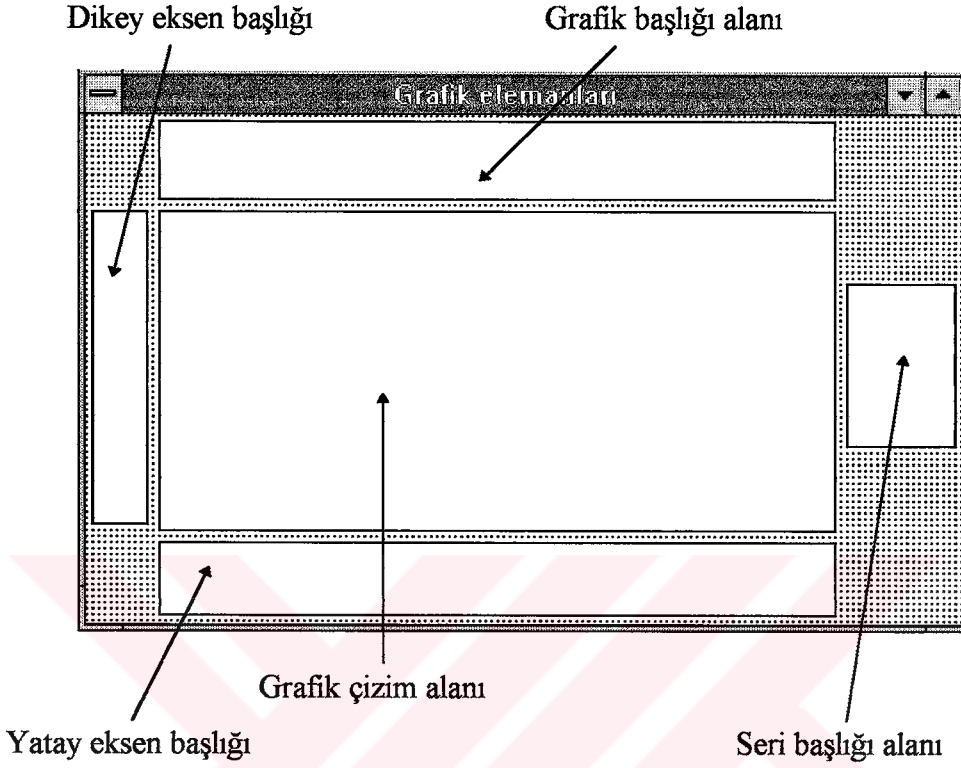
Bu kısımda yukarıda adı geçen elemanların fonksiyonları, boyutlarının hesaplanması ve bir grafik penceresi üzerindeki konumlarına karar verilmesi tartışılacaktır.

5.3.1 Grafik Başlığı

Grafiği tanıtmaya yarayan bir metin bilgisidir. Yalın bir alan olan grafik başlığında hesaplanması gereken iki kısım mevcuttur. Bunlar grafik başlığının yüksekliği ve genişliğidir.

Bir metnin ekran üzerinde kapladığı alan (genişliği ve yüksekliği) içinde bulunan karakterlere ve kullanılan yazıtipine (font) ve boyutuna (punto) göre

farklılıklar göstermektedir. Buna göre başlık alanının genişliğine ve yüksekliğine aşağıdaki gibi karar verilmektedir. Burada dikkat edilmesi gereken bir husus grafik



Şekil 5-3 Grafik alanlarının pencere üzerinde gösterimi

tabanlı işletim sistemlerinde her karakterin genişliği birbirinden farklıdır. Örnek olarak "W" harfi geniş yer kaplarken "ı" harfi daha az yer tutmaktadır. Karakterlerin genişliği ise boyutuna göre önceden belirli bir değere sahiptir. Bir karakterin yüksekliği ise kullanılan boyuta göre belirlenir ve satırların aynı hizada olmasını sağlaması bakımından aynı değere sahiptir. Özetlersek; bir harfin genişliği, harfin kullanılan yazıtipindeki genişliğine ve boyutuna, harfin yüksekliği ise (hangi harf olduğu önem taşımaksızın) sadece kullanılan yazıtipinin boyutuna göre değişiklik gösterir. Bir harfin (örnek olarak "W" harfinin) genişliği ve yüksekliği, Visual Basic dilinde aşağıdaki şekilde bulunabilir.

```
x = grafik_basligi.TextWidth("W")
```

```
y = grafik_basligi.TextHeight("W")
```

Yukarıdaki formülde, grafik_basligi bir “picture box” nesnesi (Visual Basic ve Windows’daki ismiyle “Control”) olup, özellikleri arasında “FontName” ve “FontSize” önceden bilinen değerlere sahiptir. Örnek olarak FontName=“Arial” ve FontSize=12. “TextWidth” fonksiyonu, grafik_basligi nesnesinde kullanılan “FontName” ve “FontSize” değerlerini bularak karakterin genişliğini fonksiyon sonucu olarak döndürür. Aynı şekilde “TextHeight” fonksiyonu da, karakterin yüksekliğini bulur.

Son olarak fonksiyonların döndürdüğü değer ile ilgili bazı hususlara dikkat edilmesi gereklidir. Grafik ile ilgili tüm fonksiyonların giriş ve çıkış değerlerine, nesnenin bir başka özelliğine “ScaleMode” bakılarak karar verilir. “ScaleMode” özelliği, farklı gösterim ortamlarındaki farklı çözünürlüklerde, hesaplamalar sırasında ölçekli görüntüler yaratabilmek için kullanılan bir özelliktir. Aşağıdaki tabloda “ScaleMode”un aldığı değerlerin ne anlama geldiğini görmek mümkündür. Bkz. Tablo 5-1.

Tablo 5-1 Ekran veya yazıcı ölçekleme türleri

ScaleMode	Ölçek adı
0	Özel ölçekleme
1	Twip
2	Point (punto)
3	Pixel (benek)
4	Karakter
5	İnç
6	Milimetre
7	Santimetre

Tablodan da anlaşılacağı gibi farklı ölçek modlarında, hesaplamalar bu modlar üzerinden yapılmaktadır. Bu ölçülerin birbirine karşılık gelen çevrim değerleri: 1440 twip = 1 mantıksal inç, veya 567 twip bir mantıksal santimetre, 72 points = 1 mantıksal inç, 1 karakter = (yatayda 120 twip, dikeyde 240 twip) şeklindedir.

Yukarıda verilen çevrim değerleri hesaplanırken ekranın veya yazıcının birim değeri “pixel” kabul edilir ve örnek olarak ekranın veya yazıcının o andaki çözünürlüğüne göre bir “pixel”in kaç “twip”e denk geldiği işletim sistemi tarafından da bilinen bir değere sahiptir. Bu değer ekran için;

$$x = \text{Screen.TwipsPerPixelX}$$

$$y = \text{Screen.TwipsPerPixelY}$$

yazıcı için ise;

$$x = \text{Printer.TwipsPerPixelX}$$

$$y = \text{Printer.TwipsPerPixelY}$$

fonksiyonları kullanılarak öğrenilebilir. Ancak verilen herhangi bir ölçek modundaki çizimin, “pixel” moduna çevrilerek basılması işletim sistemi tarafından yapıldığından, ve giriş ölçeği ile çıkış ölçekleri aynı türden kabul edileceğinden, ilerleyen bölümlerde kullanılacak formüllerde bir değişikliğe gerek kalmayacağı açıktır.

- i. Grafik başlığının genişliği : Grafik başlığının birkaç satır olarak alındığında, metin alanının bir satırında bulunan karakterlerin uzunluklarının toplanması ve en uzun satırın belirlenmesi gerekir. m satır sayısını, $n_1, n_2, \dots, n_q$ her satırdaki karakter sayısını ve x_{ij} karakterin genişliğini gösterdiği durumda hesaplama;

$$\max \left\{ \sum_{j=1}^{n_1} x_{1j}, \sum_{j=1}^{n_2} x_{2j}, \dots, \sum_{j=1}^{n_q} x_{mj} \right\} \quad (5-1)$$

şeklinde olacaktır.

- ii. Grafik başlığının yüksekliği : Grafik başlığının yüksekliğine, başlıkta bulunan satır sayısına bakılarak karar verilir. m satır sayısını, ve y herhangi bir harfin yüksekliğini gösterdiği durumda grafik başlığının yüksekliği; $m \cdot y$ kadar olacaktır.

5.3.2 Yatay Eksen Başlığı

Yatay eksen başlığı, yatayda gösterilen ekseni (genellikle yatayda x ekseni bulunmakla birlikte, mesela “bar” grafiklerde yatayda y ekseni bulunur) açıklamakta kullanılır. Burada da grafik başlığında olduğu gibi eksen başlığının, yüksekliğinin ve genişliğinin bulunmasında kullanılan hesaplamalar uygulanır. Bu alanın bulunmadığı durumlarda (pay grafiklerde olduğu gibi) yükseklik ve genişlik değerleri 0 kabul edilir.

5.3.3 Dikey Eksen Başlığı

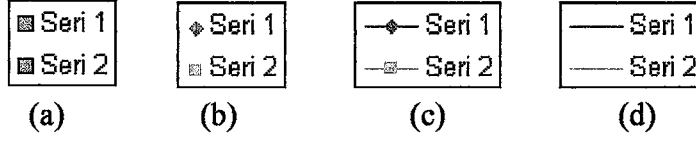
Dikey eksen başlığı temel özellikleriyle grafik başlığı ve yatay eksen başlıkları ile aynı olmakla beraber, önemli bir farkı vardır. Bu fark yatay düzlemde grafik çizim alanına daha çok yer ayrılmasını sağlamak amacıyla başlığın aşağıdan yukarıya doğru dikey olarak yazılması gereğidir. Bu zorunluluktan dolayı, hesaplama işlemlerinde yükseklik ve genişlik hesaplarının yatay ve dikeyler için ters çevrilerek yapılması ve yazılan metnin 3. bölümdeki döndürme işlemi kullanılarak saat yönünün tersinde 90° çevrilmesi gerekir.

5.3.4 Seri Başlıkları

Seri başlıkları, grafik üzerinde gösterilmekte olan çeşitli sayıdaki işaret ve renk gruplarına karşılık gelen serilerin herbirini açıklamaya yarayan bir alandır. Bu alan iki kısımdan oluşur. Bunlar her serinin çizimi sırasında kullanılan renk ve işaretleri gösteren grafik grubu ile bu işaretleri açıklamayı sağlayan seri isimlerini açıklayan metin alanlarıdır. Bkz. Şekil 5-4.

Seri başlıklarında kullanılan renkli işaretler, alınan grafik türüne göre değişir. İçi boyanmak suretiyle yapılan grafiklerde (pay, alan, çubuk) Şekil 5-4a’da görülen içi renkli boyanmış kareler, seriyi ifade eden değerlerin sadece işaretlerle gösterildiği grafiklerde Şekil 5-4b de görülen önceden belirlenmiş şekiller ile, değerlerin düz

izgilerle birleřtirildiđi durumlarda řekil 5-4c ve řekil 5-4d’de grlen izimler kullanılır.



řekil 5-4 Seri bařlıđı eřitleri

Seri bařlıđı alanının yksekliđine karar verilirken, diđer bařlık alanlarında kullanılan hesaplar burada da kullanılır. Seri bařlıđı alanındaki satır sayısı, seri adedi kadar kabul edilir.

Seri bařlıđının geniřliđine karar verilmesi ařamasında da diđer bařlık alanlarında kullanılan hesaplamalarda olduđu gibi en uzun satırın bulunması gerekir; ancak bulunan bu deđere řekilsel aıklama iin ayrılan sabit bir miktarın ilave edilmesi gerekir.

5.3.5 Grafik izim Alanı

Bu alan iki boyutlu eđrinin izileceđi tm alanı ifade eder. Bu alan incelendiđinde  temel blm gze arpar. x ekseni alanı, y ekseni alanı ve serilerin izileceđi seri izim alanı.

5.3.5.1 X Ekseni Alanının leklenmesi ve Etiketlenmesi

Bu alan kullanılan grafik trne gre yatayda veya dikeyde bulunabilir. X ekseninin yatayda bulunduđu durumlarda alan geniřliđi grafik izim alanının geniřliđi kadardır. Yksekliđi ise etiket alanında kullanılan yazıtıpinin boyutuna gre hesaplanır. Bazı grafik trlerinde ise x ekseni dikeyde bulunur. Bu durumda alanın yksekliđi bilinir (grafik izim alanının yksekliđi kadar) ve geniřliđinin hesaplanması gerekir.

X ekseninin bir başka özelliği ölçekleme sırasında ortaya çıkar. Eğri çizimlerinde x ekseninin kaç parçaya ayrılacağı önemli bir sorundur. Burada ilk önce $X_{\min}$ ve $X_{\max}$ değerlerinin bilinmesi gerekir. [7], [8] Bu değerler ekseninin en sağında ve en solunda yer alacak değerlerdir. İstendiği taktirde $X_{\min}$ değerinin altında ve $X_{\max}$ değerinin üstünde bir miktar alan bırakılabilir. Örnek olarak;

$$X_{\min} - X_{\text{ortalama}} \text{ ve } X_{\max} + X_{\text{ortalama}}$$

veya

$$X_{\min} - X_{\sigma} \text{ ve } X_{\max} + X_{\sigma}$$

hatta bazen $X_{\min}$ ve $X_{\max}$ değerlerinin yuvarlatılması istenebilir. $X_{\min}=3$, $X_{\max}=91$ ise sol tarafın 0 sağ tarafın 100 değerlerine tamamlanması gibi.

X ekseninin tespit edilen alt ve üst değerleri (iki değer) arasında en uzun olanı bulunarak, buna eklenecek keyfi bir boşluk payı ile birlikte bir etiket biriminin en fazla ne kadar yer işgal ettiği hesaplanır. Bu değer iki katı alınarak (alt ve üst değer etiketlerinin yazılabilmesi için) x ekseninin en az ne kadar genişlikte olması gerektiği hesaplanır. Eğer x ekseninin belirlenen uzunluğu bu değer tam katlarını sağlıyorsa etiketleme işlemi bu kata göre yeniden düzenlenir.

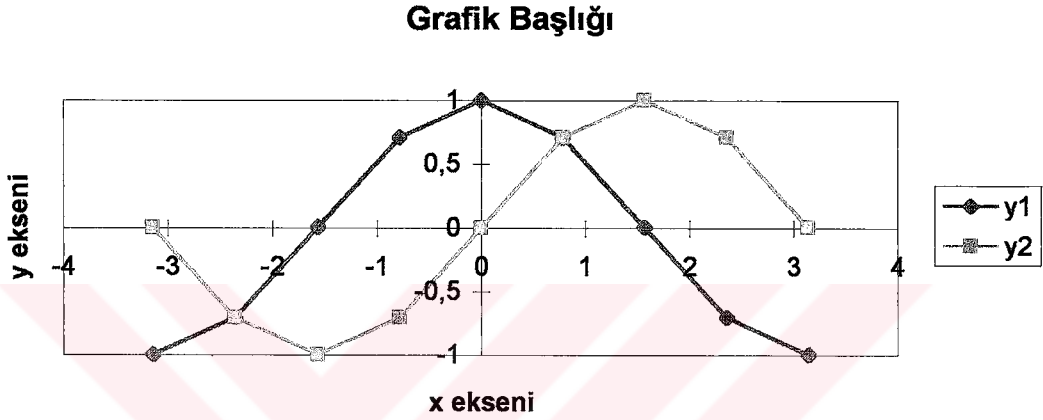
5.3.5.2 Y Ekseni Alanının Ölçeklenmesi ve Etiketlenmesi

Y ekseninin etiketlenmesi ve ölçülmesi x eksenine göre daha basittir (belirli bir yazıtipinde her karakterin aynı yükseklikte olduğu hatırlayınız). Alana sığacak etiket sayısı, alanın yüksekliğini karakter boyuna bölmekle bulunur.

Bu ekseninde $Y_{\min}$ ve $Y_{\max}$ testlerinin verilen tüm seriler için yapılması ve bu seriler içinde bulunan en küçük değer $Y_{\min}$ ve en büyük değer $Y_{\max}$ tespit edilmesi gerekir. Burdan çıkan değer için x ekseninde anlatılan tamsayıya vs. Yuvarlama işlemleri bu eksen için de yapılabilir.

5.3.6 XY Dağılım Grafikleri (XY Scatter)

X eksenı deęerlerinin ölçeklenebilir olduęu durumlarda kullanılır. x ve y deęerleri bilinen bir noktanın grafik üzerindeki konumuna eksen çizimlerinde hesaplanmış olan $[X_{\min}, X_{\max}]$ ile $[Y_{\min}, Y_{\max}]$ deęerlerine bakılarak karar verilir. x deęeri $X_{\max} - X_{\min}$ deęerine ölçeklenerek yatayda geleceęi pixel, y deęeri $Y_{\max} - Y_{\min}$



Şekil 5-5 XY Dağılım grafik örneęi

deęerine ölçeklenerek dikeyde geleceęi pixel bulunur ve nokta yerine konulur. Eęer noktalar arasının çizgi ile birleřtirilmesi isteniyorsa, en son konan nokta ile yeni bulunan nokta arası “Line” komutu kullanılarak birleřtirilir. Örnek bir XY scatter grafięi için Bkz. Şekil 5-5.

Bu tür grafiklerde bir x eksenı serisine karřılık, y ekseninde bir veya daha fazla seri bulunabilir. X ve y serilerinde negatif deęerler bulunabilir.

Alt türleri incelendięinde üç ayrı şekilde tanımlanması mümkündür.

- Seri deęerlerinin sadece bir iřaret ile gösterildięi grafikler. Bu grafiklerde amaç serinin sıklık (frekans) dağılımını grafik üzerinde göstermektir. Bu grafik türünde görme karmařıklıęından kaçınmak amacıyla y kategorisinde bir seri kullanılması

tercih edilir. Eğer birden fazla serinin dağılımı gözlenmek isteniyorsa, grafiği farklı tablolar halinde almak tercih edilir.

- ii. Serideki değerlerin serideki sıra içinde düz çizgilerle birleştirilmesi. Serideki değerlerin, belirli bir eğimi yansıttığı gösterilmek istendiğinde kullanılır. Aynı zamanda birden fazla serinin eğimini de gözlemek mümkündür.
- iii. Hem işaret ve hem de düz çizgilerin kullanıldığı grafikler. İlk iki grafiğin birleşimi olan bu grafik yukarıda anılan iki temel amaç (sıklık dağılımı ve eğim gösterme) için kullanılabildiği gibi aynı zamanda seri üzerindeki dönüm noktalarını belirtmek amacıyla da kullanılır.
- iv. Eksenlerden birindeki veya her ikisindeki değerlerin dağılımı şiddetli olduğunda bir başka deyişle standard sapmanın, aritmetik ortalamadan büyük olduğunda durumlarda ilgili eksen için tercih edilirler. Logaritma işlemi için serideki değerler üzerinde değişiklik yapılmadan sadece grafiğin çizim aşamasında uygulanır.
- v. Logaritmik ölçeğin uygulanacağı eksene karşılık gelen seri veya serilerdeki tüm değerler içinde (logaritmanın tabiatı gereği) negatif değer bulunmaması gerekir.

5.3.7 Pay (Pasta) Grafikler

Pay grafiklerde tek bir serinin çizimine izin verilir. Bu grafik türünde eksen etiketleri ve başlıkları bulunmaz. Pay grafiklerde seri elemanlarının seri içindeki önemlerine (büyüklüklerine) göre bir pay alırlar. Bunu hesaplamak için serideki elemanlar toplanarak, her birinin toplam içindeki payları yüzdeleri bulunur. Daha sonra 360° nin $100'$ e bölümü olan 3.6 sayısı ile bulunan yüzdeler çarpılır. Böylece her bir eleman için bulunan değerler elemanın pay içinde alacağı θ açısını verir. Bu θ açısına bakılarak her bir elemanın pay içindeki yerleri (daire çizildikten sonra $\theta = 1.360$, $x=r.\sin\theta$ ve $y=r.\cos\theta$) kümülatif yüzdeler ile dairenin merkezine çizilen bir "Line" komutu ile belirlenir. Örnek olarak Line (xm,ym)-(r*sin(ky), r*cos(ky)).

Burada x_m ve y_m dairenin merkez noktasını, k_y ise serideki bir elemanın kümülatif yüzdesini göstermektedir.

Diğer grafiklerin nasıl oluşturulduğu uygulama bölümüne bakılarak daha iyi anlaşılabilir.



BÖLÜM 6. UYGULAMA

Bu kısımda geliştirilen uygulama anlatılacak kullanımı ile ilgili kısa bilgiler verilecektir.

Uygulama iki esasa dayanmaktadır. Bunlar;

- Serilerin oluşturulması,
- Bu serilerden yararlanılarak grafiksel çizimler elde etmek.

Seri oluşturma işleminde bir seride bulunabilecek en fazla eleman sayısı 2^{32} ile sınırlıdır. Bir seriye, doğrudan bilgi girişi yapılabileceği gibi, girilen bir fonksiyonun üreteceği sonuçlar da seriyi oluşturabilir. Verilerin farklı zamanlarda kullanılabilmesini sağlamak amacıyla dosyala işlemleri uygulamaya eklenmiştir.

Girilen her seriye bir isim verildikten sonra, bazı serilerin grafiğinin alınması bir menu işlemi ile sağlanır. Grafik alındıktan sonra grafiğin özellikleri değiştirilebilir. Aynı anda farklı grafikleri oturumları açılarak hepsinin birarada gözlenebilme imkânı vardır.

6.1 Menüler ve Dosyalamalar

Program pek çok parçadan oluşmakla birlikte, kullanıcı arayüzünde sürekli kullanılan üç farklı pencere mevcuttur.

Bunlardan ilki diğer pencereleri içinde barındıran İstagraf adlı bir penceredir. Bkz. Şekil 6-1.

Adsız:2	
	Yıllar
1	1991
2	1992
3	1993
4	1994
5	1995

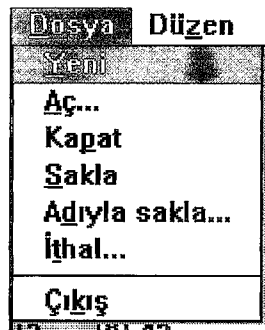
Satışlar	
1	2.683,97
2	3.285,27
3	2.720,07
4	4.137,06
5	409,47

Adsız:		
	x	Seril
1	70,55	266,71
2	57,95	144,78
3	30,19	387,37
4	1,4	380,36
5	81,45	354,52
6	4,54	207,02
7	86,26	395,24
8	37,35	480,98

Şekil 6-1 Uygulama ekranı ve giriş formu

6.2 Menülerin tanıtımı

Bu kısımda bulunan menüler (Bkz Şekil 6-2) ve fonksiyonları aşağıda kısaca anlatılacaktır.



Şekil 6-2 Dosya menüsü

Dosya Menüsü: Adından da anlaşılacağı gibi dosyalama işlemleri ile ilgili menü komutları bu menüde bulunur. Bunların fonksiyonları aşağıda verildiği gibidir:

Yeni... : İki tür pencereden (seri giriş ve grafik) birinin açılmasını sağlar,

Aç... : Daha önceden kaydedilmiş seri bilgileri dosyasının açılması,

Kapat : Aktif olan pencereyi kapatır,

Sakla, Adıyla sakla

: Girilen bilgilerin bir dosyaya yazdırılması,

İthal... : Bir metin dosyasının giriş ekranı içine dahil edilmesini,

Çıkış : Programın sona ermesini sağlar.

Düzen menüsünde (Bkz. Şekil 6-3) giriş formundaki alanlar üzerinde işlemler yapan bölümler bulunur. Bunların fonksiyonları da aşağıda verildiği gibidir:

Kes : Seçilen bir alanın hafızaya alındıktan sonra silinmesini,

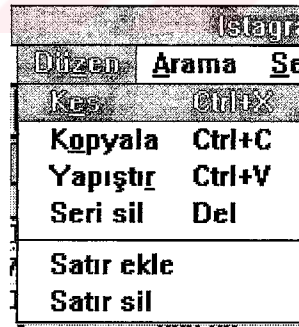
Kopyala : Seçilen bir alanın hafızaya alınmasını,

Yapıştır : Hafızada bulunan bir bilginin giriş formuna konulmasını,

Sil : Bir serinin tamamen silinmesini,

Satır ekle : Serilere birer satır ilave edilmesini,

Satır sil : Serilerden bir satır silinmesini sağlar.



İstekte	
Düzen	Arama Se
Kes	Ctrl+X
Kopyala	Ctrl+C
Yapıştır	Ctrl+V
Seri sil	Del
Satır ekle	
Satır sil	

Şekil 6-3 Düzen menüsü

Silme işlemleri sırasında, bu silmeden etkilene satırlardaki değerler yeniden hesaplanmaz. Tekrar hesaplama yaptırmak için seçenekler menüsünden “seri özellikleri” ne girerek “Şimdi hesapla” butonunun seçilmesi gerekir.

Diğer menü işlemleri pek çok uygulamada kullanılan genel işlemleri içermektedir. Farklı olan bir menü, yukarıda da bahsedilen “Seçenekler”

menüsündeki, “Seri özellikleri” dir. Bu kullanıcıya serini görünüşü ve hesaplanışı ile ilgili seçenekler verir. Seri sağa dayalı, sola dayalı veya ortalanmış mı olsun, ondalık hanelerin ne kadarı görünsün, hangi yazıtipi ve renginde olsun vs.

Ancak özellikler içinde en önemlisi “Formül” alanıdır. Bu alan, bilgi giriş formu kullanılarak oluşturulan serilerde, bir değere sahip değildir. Ancak bir serinin değeri diğer seriler kullanılarak yapılmak istendiğinde, bu “formül” alanına bir fonksiyonun girişi ile olur. Formül girildikten sonra serideki her değer (gerektiğinde diğer serilere bakılarak) hesaplanır ve yerine konur.

Formül girilirken dikkat edilmesi gereken bir husus, diğer serileri formül içinde kullanmak gereğidir. Formül alanı içinde bir başka seri kullanılmadığında, formülül bulunduğu serideki tüm değerler birbirinin aynı olacaktır. Örnek olarak bu alana ‘ $1/\sin(\pi/3)$ ’ yazıldığında, serideki tüm değerler 1.5407 değerine eşit olacaktır. Seri3 alanına girilen formül ‘ $\text{seri1} * \log(\text{seri2})$ ’ benzeri bir değer aldığında Seri3 sütunundaki bütün değerler (ilgili Seri1 ve Seri2 satırlarındaki değerlere bağlı olarak) değişecektir.

6.3 İstagraf Fonksiyonları

Bu bölümde, uygulamada kullanılan fonksiyonların tanıtımı, matematiksel ve istatistiksel olmak üzere iki grupta yapılacaktır. Bu uygulamada kullanılan fonksiyonların parçalanarak yorumlanması için Ek A bölümünde bulunan ‘PARSER.BAS’ program parçası kullanılmıştır.

6.3.1 Matematiksel Fonksiyonlar

ABS Fonksiyonu

Yazılımı : ABS(sayı)

Açıklama : sayı’nın mutlak değerini bulur.

ATN Fonksiyonu

Yazılımı : ATN(sayı)

Açıklama : sayı'nın mutlak değerini bulur.

-pi/2 ile pi/2 arasında radyan cinsinden sonuç verir.

Dereceyi radyana çevirmek için dereceyi pi/180 ile çarpmak, radyanı dereceye çevirmek için ise radyanı 180/pi ile çarpmak gerekir.

COS Fonksiyonu

Yazılımı : COS(açı)

Açıklama : açı'nın radyan cinsinden kosinüsünü bulur.

Açı radyan cinsindendir. Sonuç -1 ile 1 arasında bir değerdir.

Dereceyi radyana çevirmek için dereceyi pi/180 ile çarpmak, radyanı dereceye çevirmek için ise radyanı 180/pi ile çarpmak gerekir.

EXP Fonksiyonu

Yazılımı : EXP(sayı)

Açıklama : e sayısının sayı. kuvvetini bulur.

Sayı 709.782712893 değerini geçerse hata verir.

FIX ve INT Fonksiyonu

Yazılımı : FIX(sayı) veya INT(sayı)

Açıklama : her ikisinde sayının tamsayı kısmını bulur. Farkları, sayı negatif olduğunda INT ilk küçük tamsayısı, FIX ise ilk büyük tamsayıyı bulur

LOG Fonksiyonu

Yazılımı : LOG(sayı)

Açıklama : sayının tabi logaritmasını bulur.

SIN Fonksiyonu

Yazılımı : SIN(açı)

Açıklama : açı'nın radyan cinsinden sinüsünü bulur.

Açı radyan cinsindendir. Sonuç -1 ile 1 arasında bir değerdir.

Dereceyi radyana çevirmek için dereceyi $\pi/180$ ile çarpmak, radyanı dereceye çevirmek için ise radyanı $180/\pi$ ile çarpmak gerekir.

SQR Fonksiyonu

Yazılımı : SQR(sayı)

Açıklama : sayının karekökünü bulur.

TAN Fonksiyonu

Yazılımı : TAN(sayı)

Açıklama : açı'nın radyan cinsinden tanjantını bulur.

Açı radyan cinsindendir. Sonuç -1 ile 1 arasında bir değerdir.

Dereceyi radyana çevirmek için dereceyi $\pi/180$ ile çarpmak, radyanı dereceye çevirmek için ise radyanı $180/\pi$ ile çarpmak gerekir.

PI, PI2 ve PI4 Sabitleri

Yazılımı : pi, pi2, pi4

Açıklama : $\pi = 3.14159265358979$,
 $\pi^2 = 3.14159265358979 / 2$,
 $\pi^4 = 3.14159265358979 / 4$ sonuçlarını üretir.

6.3.2 İstatistik Fonksiyonları

ORTALAMA Fonksiyonu

Yazılımı : ORTALAMA(seri)

Açıklama : serinin aritmetik ortalamasını bulur.

HORTALAMA Fonksiyonu

Yazılımı : HORTALAMA(seri)

Açıklama : serinin harmonik ortalamasını bulur.

GORTALAMA Fonksiyonu

Yazılımı : GORTALAMA(seri)

Açıklama : serinin geometrik ortalamasını bulur.

Seride negatif veya 0 değerinin olmaması gerekir.

KORTALAMA Fonksiyonu

Yazılımı : KORTALAMA(seri)

Açıklama : serinin kareli ortalamasını bulur.

ORTSAPMA Fonksiyonu

Yazılımı : ORTSAPMA(seri)

Açıklama : serinin ortalama sapmasını bulur.

STDSAPMA Fonksiyonu

Yazılımı : STDSAPMA(seri)

Açıklama : serinin standart sapmasını bulur.

KOREL Fonksiyonu

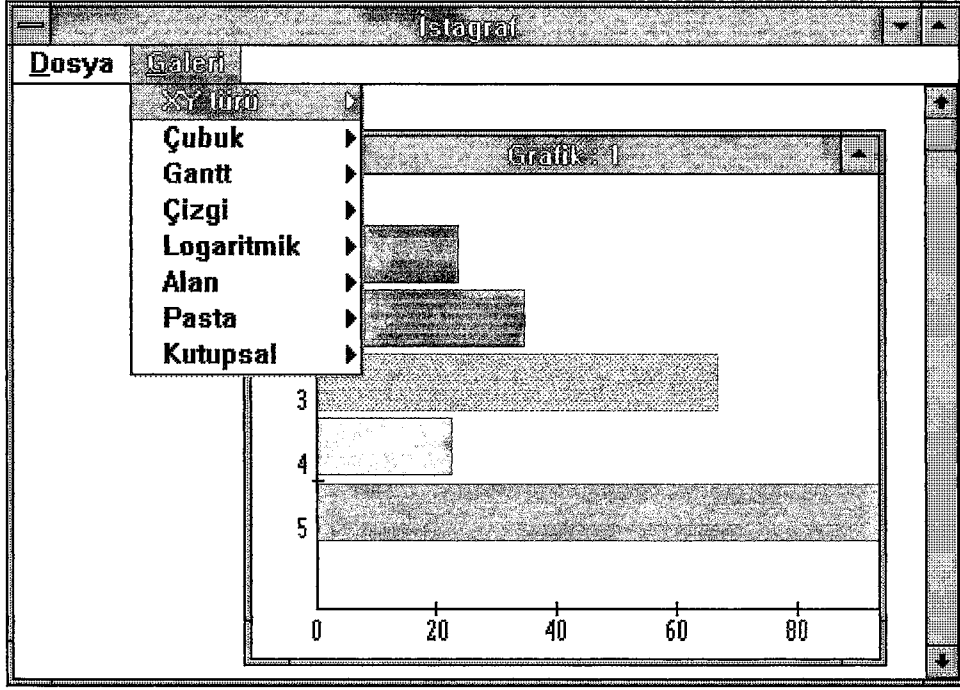
Yazılımı : KOREL(seri1,seri2)

Açıklama : İki serinin korelasyon katsayısını bulur.

6.4 Grafik Penceresinin Kullanımı

Grafik penceresinde bir grafik görüntüsü alındıktan sonra buradaki menüler yardımıyla grafiğin özellikleri değiştirilebilir. Bkz. Şekil 6-4.

Menüden XY, çubuk, gantt, çizgi, logaritmik, alan, pasta ve kutupsal grafikler alınabilir. Aynı zamanda dosya menüsü kullanılarak grafik görüntünün yazıcıdan çıkışı sağlanabilir.



Şekil 6-4 Grafik Penceresi

SONUÇ ve ÖNERİLER

Bu çalışmada grafiksel eğrilerin ve nesnelerin çizimi ve bu eğrilerin oluşturulmasında kullanılan seriler üzerinde istatistiksel analizler yapılmaya çalışılmıştır. Serilerdeki eleman sayıları üzerine büyük kısıtlamalar getirilmemiş, aksine analizlerde faydalı olması bakımından seri giriş ekranında $2^{32}-1$ elemanlık bir serbesti getirilmiştir.

Çalışma sırasında nesneye dayalı programlama kullanılmış ve bununla ilgili bazı ipuçları benzer bir sistemde çalışmak isteyenlere başlangıç bilgisi olarak yeterli olacaktır..

Ancak bu çalışmaya ilave edilecek çeşitli istatistiksel analiz yöntemleri ve üç boyutlu grafiksel yüzey algoritmalarının eklenmesiyle çalışma daha etkin olarak kullanılabilir.

KAYNAKLAR

- [1] ROGERS, D. F., ADAMS, J. A., "Mathematical Elements for Computer Graphics", McGraw-Hill Inc., Singapore, 1990
- [2] SCOTT, J. E., "Introduction to Interactive Computer Graphics", Willey-Interscience Publication, New York, 1982
- [3] WATT, A. H. "Fundamentals of Three-Dimensional Computer Graphics", Addison-Wesley Inc., England, 1989
- [4] CILLOV, C., "İstatistik Tekniđi ve Uygulaması", İstanbul Üniversitesi Yayınları, İstanbul, 1976
- [5] URAL, K., "İstatistik Yöntemleri ve Uygulamaları", İstanbul Üniversitesi Yayınları, İstanbul, 1979
- [6] RICH, E., KNIGHT, K., "Artificial Intelligence", McGraw-Hill Inc., Singapore, 1991
- [7] TUNCER, C., "Bilgisayarda Eğri Çizimi", İstanbul Teknik Üniversitesi, Cilt 50, Sayı 4, 1992
- [8] TUNCER, C., "Grafik Modelleme", İstanbul Teknik Üniversitesi, Cilt 51, Sayı 3, 1993

EK A. UYGULAMA

Bu uygulama kullanılan programlama dili Visual Basic 4.0'dır. Uygulama programının çalıştırılabilmesi için 'ISTAGRAF.VBP' isimli dosyada bulunan '.VBX' (Visual Basic Extender) ve '.OCX' (Object Control Extender) kontrol programlarının sistemde bulunması gerekir.

Programın çalıştırılabilmesi için Visual Basic çalışma ortamına girildikten sonra 'ISTAGRAF.VBP' isimli dosyanın proje dosyası yüklenmesi ve çalıştırılması gerekir. Programın çalışan ilk satırı ana modülde (ISTAGRAF.FRM) bulunan MDIForm_Load alt yordamının ilk satırıdır.

```
*****
ISTAGRAF.VBP
*****

Module=Module1; PARSER.BAS
Module=Module2; GENEL.BAS
Module=Module3; DOSYAC.BAS
Module=Module4; DİGER.BAS
Form=ISTAGRAF.FRM
Form=DBDUZEN.FRM
Form=ÖZELLİK.FRM
Form=FRGEGRF.FRM
Form=GRAFIK.FRM
VBX=E:\WINVB16\SYSTEM\TRUEGRID.VBX
Object={F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.0#0; COMDLG16.OCX
Reference=*\G{00025E01-0000-0000-C000-
000000000046}#2.5#0#E:\WINVB16\SYSTEM\DAO2516.DLL#Microsoft DAO 2.5
Object Library
Object={BDC217C8-ED16-11CD-956C-0000C04E4C0A}#1.0#0; TABCTL16.OCX
Object={0842D103-1E19-101B-9AAF-1A1626551E7C}#1.0#0; GRAPH16.OCX
ProjWinSize=75,467,170,340
ProjWinShow=2
IconForm="frmISTAGRAF"
HelpFile=""
Title="İstagraf 1.0"
ExeName32="ISTAGRAF.EXE"
ExeName="ISTAGRAF.EXE"
Command32=""
Name="IstaGraf"
StartMode=0
VersionCompatible32="0"
VersionCompatible="0"
MajorVer=1
MinorVer=0
```



```
*****
PARSER.BAS
*****
```

```
Attribute VB_Name = "Module1"
Option Explicit
Global input_line As String, character As String, result As String, c As String
Global exp_value As Double
Global scan_index As Integer
Global place As Integer, character_class As Integer

Global dbformül As Database, tbgenel As Table, tbgenel1 As Table
Global tbözel As Table
```

```
Function evaluate(ByVal expr_line As String) As Double
input_line = UCase$(expr_line)
scan_index = 0
nextchr
exp_value = 0
expr exp_value
If character_class = 10 Then
    evaluate = exp_value
Else
    evaluate = 0
End If
End Function
```

```
Sub expr(exp_answer As Double)
Dim temp As Double

If Not (character_class = 1 Or character_class = 2) Then
    scan_index = scan_index - 1
    term exp_answer
End If
Do
    temp = 0
    Select Case character_class
    Case 1 ' Toplama işlemi
        term temp
        exp_answer = exp_answer + temp
    Case 2 ' Çıkarma işlemi
        term temp
        exp_answer = exp_answer - temp
    Case Else
        Exit Sub
    End Select
Loop
End Sub
```

```
Sub exps(exps_answer As Double)
Dim temp As Double
factor exps_answer
Do
    temp = 0
    Select Case character_class
    Case 5 ' Üs işlemi
        factor temp
        exps_answer = exps_answer ^ temp
    Case Else
        Exit Sub
    End Select
Loop
End Sub
```

```
Sub factor(factor_answer As Double)
```

```
Dim temp As Double

nextchr
Select Case character_class
Case 6 'sol parantez
    nextchr
    expr factor_answer
    If character_class = 7 Then 'sağ parantez
        nextchr
    Else
        character_class = 0 'geçersiz karakter
    End If
Exit Sub
Case 8
    parse_numeric factor_answer
Case 9 'fonksiyon veya değişken
    parse_function factor_answer
Case Else
    character_class = 0
End Select

End Sub

Sub nextchr()
Do
    scan_index = scan_index + 1
    If scan_index > Len(input_line) Then
        character_class = 10
        Exit Sub
    End If
    character = Mid$(input_line, scan_index, 1)
Loop While (character = " ")
Select Case (character)
Case "+"
    character_class = 1
Case "-"
    character_class = 2
Case "*"
    character_class = 3
Case "/"
    character_class = 4
Case "^"
    character_class = 5
Case "("
    character_class = 6
Case ")"
    character_class = 7
Case "0" To "9", "."
    If Not character_class = 9 Then character_class = 8
Case "a" To "z", "A" To "Z", "0" To "9"
    character_class = 9
Case Else
    character_class = 0
End Select
End Sub

Sub parse_function(parse_answer As Double)
Dim temp As Double
Dim x As String
x = ""
Do While character_class = 9
    x = x & character
    nextchr
Loop
Select Case character_class
Case 6
```

```
x = Trim$(x)
Select Case x
Case "ORTALAMA"
    parse_answer = ıstatistik(1)
Case "HORTALAMA"
    parse_answer = ıstatistik(2)
Case "GORTALAMA"
    parse_answer = ıstatistik(3)
Case "KORTALAMA"
    parse_answer = ıstatistik(4)
Case "ORTSAPMA"
    parse_answer = ıstatistik(5)
Case "STDSAPMA"
    parse_answer = ıstatistik(6)
Case "KOREL"
    parse_answer = ıstatistik(7)
Case Else
    scan_index = scan_index - 1
    factor temp

    Select Case x
    Case "ABS"
        parse_answer = Abs(temp)
    Case "ATN"
        parse_answer = Atn(temp)
    Case "COS"
        parse_answer = Cos(temp)
    Case "EXP"
        parse_answer = Exp(temp)
    Case "FIX"
        parse_answer = Fix(temp)
    Case "INT"
        parse_answer = Int(temp)
    Case "LOG"
        parse_answer = Log(temp)
    Case "RND"
        parse_answer = Rnd(temp)
    Case "SGN"
        parse_answer = Sgn(temp)
    Case "SIN"
        parse_answer = Sin(temp)
    Case "SQR"
        parse_answer = Sqr(temp)
    Case "TAN"
        parse_answer = Tan(temp)
    Case Else
        character_class = 0
    End Select
End Select
Case Else
    x = Trim$(x)
    'scan_index = scan_index - 1
    Select Case x
    Case "PI"
        parse_answer = 3.14159265358979
    Case "PI2"
        parse_answer = 3.14159265358979 / 2
    Case "PI4"
        parse_answer = 3.14159265358979 / 4
    Case "INF"
        parse_answer = 1.79766931348623E+308
    Case "RND"
        parse_answer = Rnd
    Case Else
        On Error GoTo hatadüz
        parse_answer = tbgenel(x)
```

```
        On Error GoTo 0
    End Select
End Select
Exit Sub
hatadüz:
character_class = 0
Resume
End Sub
```

```
Sub parse_numeric(parse_answer As Double)
Static point_pos As Integer, temp As Double
point_pos = -1
Do While character_class = 8
    If character = "." Then
        If Not point_pos = -1 Then
            Exit Sub
        Else
            point_pos = 0
            nextchr
        End If
    End If

    temp = Asc(character) - Asc("0")
    If point_pos = -1 Then
        parse_answer = parse_answer * 10 + temp
    Else
        point_pos = point_pos + 1
        parse_answer = parse_answer + (temp / 10 ^ point_pos)
    End If
    nextchr
Loop
End Sub
```

```
Sub term(term_answer As Double)
Dim temp As Double
exps term_answer
Do
    temp = 0
    Select Case character_class
        Case 3      ' Çarpma işlemi
            exps temp
            term_answer = term_answer * temp
        Case 4      ' Bölme işlemi
            exps temp
            If temp <> 0 Then term_answer = term_answer / temp
        Case Else
            Exit Sub
    End Select
Loop
End Sub
```

```
Function İstatistik(Çeşit As Integer) As Double
Dim i As Long, hes1 As Double, sa As String, yer As Integer
Dim sal As String
```

```
    yer = InStr(scan_index + 1, input_line, ")")
    sa = Mid$(input_line, scan_index + 1, yer - scan_index - 1)
    seriayır sa, sal
    scan_index = yer
    tbözel.Seek "=", LCASE(sa)
```

```
    Select Case Çeşit
        Case 1 ' Aritmetik ortalama
            If IsNull(tbözel("Ortalama")) Or tbözel("Ortalama") = 0 Then
                Ortalama sa
            End If
```

```
    İstatistik = tbözel("Ortalama")
Case 2 ' Harmonik ortalama
    If IsNull(tbözel("Hortalama")) Or tbözel("Hortalama") = 0 Then
        Hortalama sa
    End If
    İstatistik = tbözel("Hortalama")
Case 3 ' Geometrik ortalama
    If IsNull(tbözel("Gortalama")) Or tbözel("Gortalama") = 0 Then
        Gortalama sa
    End If
    İstatistik = tbözel("Gortalama")
Case 4 ' Kareli ortalama
    If IsNull(tbözel("Kortalama")) Or tbözel("Kortalama") = 0 Then
        Kortalama sa
    End If
    İstatistik = tbözel("Kortalama")
Case 5 ' Ortalama Sapma
    If IsNull(tbözel("OrtSapma")) Or tbözel("OrtSapma") = 0 Then
        OrtSapma sa
    End If
    İstatistik = tbözel("OrtSapma")
Case 6 ' Standart Sapma
    If IsNull(tbözel("StdSapma")) Or tbözel("StdSapma") = 0 Then
        StdSapma sa
    End If
    İstatistik = tbözel("StdSapma")
Case 7 ' Standart Sapma
    If IsNull(tbözel("Korel")) Or tbözel("Korel") = 0 Or tbözel("Korel Y
Serisi") <> sal Then
        Korel sa, sal
    End If
    İstatistik = tbözel("Korel")

End Select
nextchr
End Function
Sub Ortalama(seriadı As String)
Dim i As Long, hes1 As Double
    tbgenell1.MoveFirst
    i = 0
    hes1 = 0
    Do
        If tbgenell1(seriadı) <> "" Then
            i = i + 1
            hes1 = hes1 + Val(tbgenell1(seriadı))
        End If
        tbgenell1.MoveNext
    Loop Until tbgenell1.EOF
    tbözel.Edit
    If i > 0 Then
        tbözel("Ortalama") = hes1 / i
    Else
        tbözel("Ortalama") = 0
    End If
    tbözel.Update
End Sub
Sub Hortalama(seriadı As String)
Dim i As Long, hes1 As Double
    tbgenell1.MoveFirst
    i = 0
    hes1 = 0
    Do
        If tbgenell1(seriadı) <> "" Then
            If Val(tbgenell1(seriadı)) <> 0 Then
                i = i + 1
                hes1 = hes1 + 1 / Val(tbgenell1(seriadı))
            End If
        End If
    Loop Until tbgenell1.EOF
    tbözel.Edit
    If i > 0 Then
        tbözel("Hortalama") = hes1
    Else
        tbözel("Hortalama") = 0
    End If
    tbözel.Update
End Sub
```

```
        End If
    End If
    tbgenell.MoveNext
    Loop Until tbgenell.EOF
    tbözel.Edit
    If i > 0 Then
        tbözel("Hortalama") = i / hes1
    Else
        tbözel("Hortalama") = 0
    End If
    tbözel.Update
End Sub

Sub Gortalama(seriadı As String)
Dim i As Long, hes1 As Double
    tbgenell.MoveFirst
    i = 0
    hes1 = 0
    Do
        If tbgenell(seriadı) <> "" Then
            i = i + 1
            hes1 = hes1 + Log(Val(tbgenell(seriadı)))
        End If
        tbgenell.MoveNext
    Loop Until tbgenell.EOF
    tbözel.Edit
    If i > 0 Then
        tbözel("Gortalama") = Exp(1) ^ (hes1 / i)
    Else
        tbözel("Gortalama") = 0
    End If
    tbözel.Update
End Sub

Sub Kortalama(seriadı As String)
Dim i As Long, hes1 As Double
    tbgenell.MoveFirst
    i = 0
    hes1 = 0
    Do
        If tbgenell(seriadı) <> "" Then
            If Val(tbgenell(seriadı)) <> 0 Then
                i = i + 1
                hes1 = hes1 + Val(tbgenell(seriadı)) ^ 2
            End If
        End If
        tbgenell.MoveNext
    Loop Until tbgenell.EOF
    tbözel.Edit
    If i > 0 Then
        tbözel("Kortalama") = Sqr(hes1 / i)
    Else
        tbözel("Kortalama") = 0
    End If
    tbözel.Update
End Sub

Sub OrtSapma(seriadı As String)
Dim i As Long, hes1 As Double, hes2 As Double, hes3 As Double
    If IsNull(tbözel("Ortalama")) Or tbözel("Ortalama") = 0 Then
        Ortalama seriadı
    End If
    hes2 = tbözel("Ortalama")
    tbgenell.MoveFirst
    i = 0
    hes1 = 0
    Do
```

```
i = i + 1
hes3 = Val(tbgenel1(seriad1)) - hes2
If hes3 < 0 Then hes3 = -1 * hes3
hes1 = hes1 + hes3
tbgenel1.MoveNext
Loop Until tbgenel1.EOF
tbözel.Edit
If i > 0 Then
    tbözel("OrtSapma") = hes1 / i
Else
    tbözel("OrtSapma") = 0
End If
tbözel.Update
End Sub

Sub StdSapma(seriad1 As String)
Dim i As Long, hes1 As Double, hes2 As Double, hes3 As Double
If IsNull(tbözel("Ortalama")) Or tbözel("Ortalama") = 0 Then
    Ortalama seriad1
End If
hes2 = tbözel("Ortalama")
tbgenel1.MoveFirst
i = 0
hes1 = 0
Do
    i = i + 1
    hes3 = Val(tbgenel1(seriad1)) - hes2
    hes1 = hes1 + hes3 ^ 2
    tbgenel1.MoveNext
Loop Until tbgenel1.EOF
tbözel.Edit
If i > 0 Then
    tbözel("StdSapma") = Sqr(hes1 / i)
Else
    tbözel("StdSapma") = 0
End If
tbözel.Update
End Sub

Sub Korel(seriad1 As String, seriad11 As String)
Dim i As Long, hes1 As Double, hes2 As Double, hes3 As Double
Dim xort As Double, yort As Double
Dim xstd As Double, ystd As Double

If IsNull(tbözel("Ortalama")) Or tbözel("Ortalama") = 0 Then
    Ortalama seriad1
End If
xort = tbözel("Ortalama")
If IsNull(tbözel("StdSapma")) Or tbözel("StdSapma") = 0 Then
    StdSapma seriad1
End If
xstd = tbözel("StdSapma")

tbözel.Seek "=", LCASE(seriad11)
If IsNull(tbözel("Ortalama")) Or tbözel("Ortalama") = 0 Then
    Ortalama seriad11
End If
yort = tbözel("Ortalama")
If IsNull(tbözel("StdSapma")) Or tbözel("StdSapma") = 0 Then
    StdSapma seriad11
End If
ystd = tbözel("StdSapma")

tbgenel1.MoveFirst
i = 0
hes1 = 0
Do
```

```
        i = i + 1
        hes1 = hes1 + (tbgenell1(seriad1) - xort) * (tbgenell1(seriad11) -
yort)
        tbgenell1.MoveNext
    Loop Until tbgenell1.EOF

    tbözel.Seek "=", LCASE(seriad1)
    tbözel.Edit
    If i > 0 Then
        tbözel("Korel") = (hes1 / i) / (xstd * ystd)
    Else
        tbözel("Korel") = 0
    End If
    tbözel("Korel Y Serisi") = seriad1
    tbözel.Update
```

End Sub

```
Sub seriayır(bir As String, iki As String)
Dim i As Integer
i = InStr(bir, ",")
If i <> 0 Then
    iki = Mid$(bir, i + 1)
    bir = Left$(bir, i - 1)
End If
End Sub
```

```
*****
GENELBAS
*****
```

```
Attribute VB_Name = "Module2"
Option Explicit
Global seriformat1 As String
Sub AlanYarat(tbd As TableDef, Adı As String, Tipi As Integer, Özellik As
Integer)
Static fld As Field
If Tipi = 10 Then
    Set fld = tbd.CreateField(Adı, Tipi, 64)
Else
    Set fld = tbd.CreateField(Adı, Tipi)
End If
fld.Attributes = Özellik
On Error Resume Next
fld.AllowZeroLength = True
On Error GoTo 0
tbd.Fields.Append fld
End Sub
```

```
Sub YeniVTYarat(Dosyadı As String)
Dim db As Database, tb As Table, tbd As TableDef
Dim fld As Field, Indeks As Index, i As Integer
```

```
'Veri tabanı yaratılıyor...
Set db = CreateDatabase(Dosyadı, dbLangTurkish)
```

```
'Tablo 1 Yaratılıyor...
Set tbd = db.CreateTableDef("Tablo 1")
```

AlanYarat tbd, "i", dbLong, dbFixedField

```
For i = 1 To 10
    AlanYarat tbd, "Seri" & i, dbMemo, dbVariableField
Next i
db.TableDefs.Append tbd
```



```
Set Indeks = tbd.CreateIndex("i")
Set fld = Indeks.CreateField("i")
Indeks.Primary = True
Indeks.Required = True
Indeks.Fields.Append fld
tbd.Indexes.Append Indeks

'Özellikler tablosu (tüm tablolar için) Yaratılıyor...
Set tbd = db.CreateTableDef("Özellikler")

AlanYarat tbd, "Seri adı", dbText, dbVariableField
AlanYarat tbd, "Formatı", dbText, dbVariableField
AlanYarat tbd, "Yaslama", dbByte, dbVariableField
AlanYarat tbd, "Formül", dbText, dbVariableField

AlanYarat tbd, "Ortalama", dbDouble, dbFixedField
AlanYarat tbd, "HOrtalama", dbDouble, dbFixedField
AlanYarat tbd, "GOrtalama", dbDouble, dbFixedField
AlanYarat tbd, "KOrtalama", dbDouble, dbFixedField
AlanYarat tbd, "OrtSapma", dbDouble, dbFixedField
AlanYarat tbd, "StdSapma", dbDouble, dbFixedField
AlanYarat tbd, "Korel", dbDouble, dbFixedField
AlanYarat tbd, "Korel Y Serisi", dbText, dbFixedField

db.TableDefs.Append tbd
' Alan adını indeksle
Set Indeks = tbd.CreateIndex("Seri adı")
Set fld = Indeks.CreateField("Seri adı")
Indeks.Primary = True
Indeks.Required = True
Indeks.Fields.Append fld
tbd.Indexes.Append Indeks

Set tb = db.OpenTable("Tablo 1")
For i = 1 To 40
    tb.AddNew
    tb("i") = i
    tb.Update
Next i
tb.Close

Set tb = db.OpenTable("Özellikler")

tb.AddNew
tb("Seri adı") = "i"
tb.Update
For i = 1 To 10
    tb.AddNew
    tb("Seri adı") = "Seri" & i
    tb.Update
Next i
tb.Close

db.Close
End Sub

Function AdVer() As String

Dim b As String, fn As String
Dim snum As Double, c As Double, d As Integer

b = Format(Now, "ssnnhhddmmyy")

snum = Mid(b, 1, 2)
```

```
snum = snum * 60 + Mid(b, 3, 2)
snum = snum * 24 + Mid(b, 5, 2)
snum = snum * 31 + Mid(b, 7, 2)
snum = snum * 12 + Mid(b, 9, 2)
snum = snum * 100 + Mid(b, 11, 2)

fn = ""
Do Until snum = 0
    c = Int(snum / 36)
    d = snum - c * 36
    snum = c
    Select Case d
        Case 0 To 9
            fn = Chr$(Asc("0") + d) & fn
        Case 10 To 35
            fn = Chr$(Asc("A") + d - 10) & fn
    End Select
Loop
AdVer = "~" & fn & ".MDB"
End Function
Function numcpy(num As Variant) As Variant
If IsNull(num) Then
    numcpy = 0
Else
    numcpy = num
End If
End Function
Function strcpy(str As Variant) As Variant
If IsNull(str) Then
    strcpy = ""
Else
    strcpy = str
End If
End Function
```

```
*****
DOSYAC.BAS
*****
```

```
Attribute VB_Name = "Module3"
```

```
Sub FOpenProc()
    DimRetVal
    On Error Resume Next
    Dim OpenFileName As String
    frmIstagraf.CMDialog1.FileName = ""
    frmIstagraf.CMDialog1.ShowOpen
    If Err <> 32755 Then ' İptal seçildi.
        OpenFileName = frmIstagraf.CMDialog1.FileName

        OpenFile (OpenFileName)
        UpdateFileMenu (OpenFileName)
        If gToolsHidden Then
            frmIstagraf!imgcutbutton.Visible = True
            frmIstagraf!imgcopybutton.Visible = True
            frmIstagraf!imgPasteButton.Visible = True
            gToolsHidden = False
        End If
    End If
End Sub
```

```
Function GetFileName(FileName As Variant)
    On Error Resume Next
    frmIstagraf.CMDialog1.FileName = FileName
    frmIstagraf.CMDialog1.ShowSave
```

```
If Err <> 32755 Then
    GetFileName = frmIstagraf.CMDialog1.FileName
Else
    GetFileName = ""
End If
End Function

Function OnRecentFilesList(FileName) As Integer
    Dim i

    For i = 1 To 4
        If frmIstagraf.mnuRecentFile(i).Caption = FileName Then
            OnRecentFilesList = True
            Exit Function
        End If
    Next i
    OnRecentFilesList = False
End Function

Sub OpenFile(FileName)
    Dim fIndex As Integer

    NL = Chr(13) + Chr(10)

    On Error Resume Next
    Screen.MousePointer = 11

    fIndex = FindFreeIndex()
    Document(fIndex).db.DatabaseName = FileName
    Document(fIndex).db.Refresh

    Document(fIndex).Tag = fIndex
    Document(fIndex).Caption = LCase(FileName)
    FDurum(fIndex).Degisti = False
    Document(fIndex).Show
    Close #1
    Screen.MousePointer = 0
End Sub

Sub SaveFileAs(OldFileName, FileName)
    On Error Resume Next
    Dim Contents As String

    Name OldFileName As FileName
    Screen.MousePointer = 0
    If Err Then
        MsgBox Error, 48, App.Title
    Else
        frmIstagraf.ActiveForm.Caption = LCase(FileName)
        FDurum(frmIstagraf.ActiveForm.Tag).Degisti = False
    End If
End Sub

Sub UpdateFileMenu(FileName)
    Dim RetVal
    RetVal = OnRecentFilesList(FileName)
    If Not RetVal Then
        WriteRecentFiles (FileName)
    End If
    GetRecentFiles
End Sub
```

```
*****
DIGER.BAS
*****

Attribute VB_Name = "Module4"
Option Explicit

Type FormState
    Silindi As Integer
    Degisti As Integer
    Color As Long
End Type

Public FDurum() As FormState
Public Document() As New frmddb

Public GDurum() As FormState
Public grafik() As New frmgraph

Public gFindString, gFindCase As Integer, gFindDirection As Integer
Public gCurPos As Integer, gFirstTime As Integer
Public ArrayNum As Integer
Public gToolsHidden As Boolean
Public Const ThisApp = "İstaGraf"
Public Const ThisKey = "Recent Files"

Function AnyPadsLeft() As Integer
    Dim i As Integer

    For i = 1 To UBound(Document)
        If Not FDurum(i).Silindi Then
            AnyPadsLeft = True
            Exit Function
        End If
    Next
End Function

Sub EditCopyProc()
    Clipboard.SetText frmIstagraf.ActiveForm.ActiveControl.SelText
End Sub

Sub EditCutProc()
    Clipboard.SetText frmIstagraf.ActiveForm.ActiveControl.SelText
    frmIstagraf.ActiveForm.ActiveControl.SelText = ""
End Sub

Sub EditPasteProc()
    frmIstagraf.ActiveForm.ActiveControl.SelText = Clipboard.GetText()
End Sub

Sub YeniDosya()
    Dim fIndex As Integer, a As String

    fIndex = FindFreeIndex()

    a = AdVer
    YeniVTYarat a
    'Buraya dosya oluşturma eklenecek

    Document(fIndex).Tag = fIndex
    Document(fIndex).db.DatabaseName = a
    Document(fIndex).db.Refresh
    Document(fIndex).Caption = "Adsız:" & fIndex
    Document(fIndex).Show
End Sub
```

```
frmIstagraf!imgcutbutton.Visible = True
frmIstagraf!imgcopybutton.Visible = True
frmIstagraf!imgPasteButton.Visible = True
End Sub

Function FindFreeIndex() As Integer
    Dim i As Integer
    Dim ArrayCount As Integer

    ArrayCount = UBound(Document)

    For i = 1 To ArrayCount
        If FDurum(i).Silindi Then
            FindFreeIndex = i
            FDurum(i).Silindi = False
            Exit Function
        End If
    Next

    ReDim Preserve Document(ArrayCount + 1)
    ReDim Preserve FDurum(ArrayCount + 1)
    FindFreeIndex = UBound(Document)
End Function

Sub FindIt()
    Dim start, pos, findstring, sourcestring, Msg, Response, Offset

    If (gCurPos = frmIstagraf.ActiveForm.ActiveControl.SelStart) Then
        Offset = 1
    Else
        Offset = 0
    End If

    If gFirstTime Then Offset = 0
    start = frmIstagraf.ActiveForm.ActiveControl.SelStart + Offset

    If gFindCase Then
        findstring = gFindString
        sourcestring = frmIstagraf.ActiveForm.ActiveControl.Text
    Else
        findstring = UCase(gFindString)
        sourcestring = UCase(frmIstagraf.ActiveForm.ActiveControl.Text)
    End If

    If gFindDirection = 1 Then
        pos = InStr(start + 1, sourcestring, findstring)
    Else
        For pos = start - 1 To 0 Step -1
            If pos = 0 Then Exit For
            If Mid(sourcestring, pos, Len(findstring)) = findstring Then
Exit For
                Next
            End If
        End If

        If pos Then
            frmIstagraf.ActiveForm.ActiveControl.SelStart = pos - 1
            frmIstagraf.ActiveForm.ActiveControl.SelLength = Len(findstring)
        Else
            Msg = "Cannot find " & Chr(34) & gFindString & Chr(34)
            Response = MsgBox(Msg, 0, App.Title)
        End If

        gCurPos = frmIstagraf.ActiveForm.ActiveControl.SelStart
        gFirstTime = False
    End Sub
```

```
Sub GetRecentFiles()
    Dim i, j As Integer
    Dim varFiles As Variant

    If GetSetting(ThisApp, ThisKey, "RecentFile1") = Empty Then Exit Sub

    varFiles = GetAllSettings(ThisApp, ThisKey)

    For i = 0 To UBound(varFiles, 1)

        frmIstagraf!mnuRecentFile(0).Visible = True
        frmIstagraf!mnuRecentFile(i).Caption = varFiles(i, 1)
        frmIstagraf!mnuRecentFile(i).Visible = True

        For j = 1 To UBound(Document)
            If Not FDurum(j).Silindi Then
                Document(j).mnuRecentFile(0).Visible = True
                Document(j).mnuRecentFile(i + 1).Caption = varFiles(i, 1)
                Document(j).mnuRecentFile(i + 1).Visible = True
            End If
        Next j
    Next i

End Sub

Sub OptionsToolbarProc(CurrentForm As Form)
    CurrentForm.mnuOToolbar.Checked = Not CurrentForm.mnuOToolbar.Checked
    If TypeOf CurrentForm Is MDIForm Then
        Else
            frmIstagraf.mnuSArac.Checked = CurrentForm.mnuOToolbar.Checked
        End If
    If CurrentForm.mnuOToolbar.Checked Then
        frmIstagraf.picToolbar.Visible = True
    Else
        frmIstagraf.picToolbar.Visible = False
    End If
End Sub

Sub WriteRecentFiles(OpenFileName)
    Dim i, j As Integer
    Dim strFile, key As String

    For i = 3 To 1 Step -1
        key = "RecentFile" & i
        strFile = GetSetting(ThisApp, ThisKey, key)
        If strFile <> "" Then
            key = "RecentFile" & (i + 1)
            SaveSetting ThisApp, ThisKey, key, strFile
        End If
    Next i

    SaveSetting ThisApp, ThisKey, "RecentFile1", OpenFileName
End Sub
```

```
*****
ISTAGRAF.FRM
*****
```

```
VERSION 4.00
Begin VB.MDIForm frmIstagraf
    BackColor      = &H8000000C&
    Caption        = "İstaGraf 1.0"
    ClientHeight   = 6525
    ClientLeft     = 45
```

```
ClientTop      = 630
ClientWidth    = 9495
Height         = 7215
Icon           = "ISTAGRAF.frx":0000
Left           = -15
LinkTopic      = ""
Top            = 0
Width          = 9615
Begin VB.PictureBox picToolbar
    Align        = 1 'Align Top
    BeginProperty Font
        name      = "MS Sans Serif"
        charset    = 1
        weight     = 700
        size       = 8.25
        underline  = 0 'False
        italic     = 0 'False
        strikethrough = 0 'False
    EndProperty
    Height       = 405
    Left         = 0
    ScaleHeight  = 375
    ScaleWidth   = 9465
    TabIndex     = 0
    Top          = 0
    Width        = 9495
    Begin MSComDlg.CommonDialog CMDialog1
        Left      = 1995
        Top       = 0
        _version   = 65536
        _extentx   = 847
        _extenty   = 847
        _stockprops = 0
        cancelerror = -1 'True
        defaulttext = "MDB"
        filter     = "İstagraf dosyası (*.mdb)|*.mdb|Tüm dosyalar (*.*)|*.*"
    End
    Begin VB.Image imgPasteButtonUp
        Height      = 330
        Left        = 5280
        Picture      = "ISTAGRAF.frx":0442
        Top         = 0
        Visible     = 0 'False
        Width       = 375
    End
    Begin VB.Image imgPasteButtonDn
        Height      = 330
        Left        = 4920
        Picture      = "ISTAGRAF.frx":0624
        Top         = 0
        Visible     = 0 'False
        Width       = 375
    End
    Begin VB.Image imgCopyButtonDn
        Height      = 330
        Left        = 4200
        Picture      = "ISTAGRAF.frx":0806
        Top         = 0
        Visible     = 0 'False
        Width       = 375
    End
    Begin VB.Image imgCopyButtonUp
        Height      = 330
        Left        = 4560
        Picture      = "ISTAGRAF.frx":09E8
```

```
Top           = 0
Visible       = 0 'False
Width        = 375
End
Begin VB.Image imgCutButtonDn
    Height     = 330
    Left       = 3840
    Picture    = "ISTAGRAF.frx":0BCA
    Top        = 0
    Visible    = 0 'False
    Width      = 375
End
Begin VB.Image imgCutButtonUp
    Height     = 330
    Left       = 3480
    Picture    = "ISTAGRAF.frx":0DAC
    Top        = 0
    Visible    = 0 'False
    Width      = 375
End
Begin VB.Image imgFileOpenButtonDn
    Height     = 330
    Left       = 2760
    Picture    = "ISTAGRAF.frx":0F8E
    Top        = 0
    Visible    = 0 'False
    Width      = 360
End
Begin VB.Image imgFileOpenButtonUp
    Height     = 330
    Left       = 3120
    Picture    = "ISTAGRAF.frx":1118
    Top        = 0
    Visible    = 0 'False
    Width      = 360
End
Begin VB.Image imgFileNewButtonUp
    Height     = 330
    Left       = 2400
    Picture    = "ISTAGRAF.frx":12A2
    Top        = 0
    Visible    = 0 'False
    Width      = 360
End
Begin VB.Image imgFileNewButtonDn
    Height     = 330
    Left       = 2040
    Picture    = "ISTAGRAF.frx":142C
    Top        = 0
    Visible    = 0 'False
    Width      = 375
End
Begin VB.Image imgPasteButton
    Height     = 330
    Left       = 1560
    Picture    = "ISTAGRAF.frx":160E
    Top        = 0
    Width      = 375
End
Begin VB.Image imgCopyButton
    Height     = 330
    Left       = 1200
    Picture    = "ISTAGRAF.frx":17F0
    Top        = 0
    Width      = 375
End
```



```
Begin VB.Image imgCutButton
    Height      = 330
    Left        = 840
    Picture     = "ISTAGRAF.frx":19D2
    Top         = 0
    Width       = 375
End
Begin VB.Image imgFileOpenButton
    Height      = 330
    Left        = 360
    Picture     = "ISTAGRAF.frx":1BB4
    Top         = 0
    Width       = 360
End
Begin VB.Image imgFileNewButton
    Height      = 330
    Left        = 0
    Picture     = "ISTAGRAF.frx":1D3E
    Top         = 0
    Width       = 360
End
End
Begin VB.Menu mnuFile
    Caption     = "&Dosya"
    Begin VB.Menu mnuDYeni
        Caption  = "&Yeni"
    End
    Begin VB.Menu mnuDac
        Caption  = "&Aç"
    End
    Begin VB.Menu mnuSeparator
        Caption  = "-"
    End
    Begin VB.Menu mnuDÇıkış
        Caption  = "Çı&kış"
    End
    Begin VB.Menu mnuRecentFile
        Caption  = "-"
        Index    = 0
        Visible  = 0 'False
    End
    Begin VB.Menu mnuRecentFile
        Caption  = "RecentFile1"
        Index    = 1
        Visible  = 0 'False
    End
    Begin VB.Menu mnuRecentFile
        Caption  = "RecentFile2"
        Index    = 2
        Visible  = 0 'False
    End
    Begin VB.Menu mnuRecentFile
        Caption  = "RecentFile3"
        Index    = 3
        Visible  = 0 'False
    End
    Begin VB.Menu mnuRecentFile
        Caption  = "RecentFile4"
        Index    = 4
        Visible  = 0 'False
    End
End
Begin VB.Menu mnuSeçenekler
    Caption     = "&Seçenekler"
    Begin VB.Menu mnuSArac
        Caption  = "&Araç çubuğu"
```

```
End
End
End
Attribute VB_Name = "frmIstagraf"
Attribute VB_Creatable = False
Attribute VB_Exposed = False

Private Sub imgCopyButton_Click()
    imgcopybutton.Refresh
    EditCopyProc
End Sub

Private Sub imgCopyButton_MouseDown(Button As Integer, Shift As Integer, x
As Single, Y As Single)
    imgcopybutton.Picture = imgCopyButtonDn.Picture
End Sub

Private Sub imgCopyButton_MouseMove(Button As Integer, Shift As Integer, x
As Single, Y As Single)
    Select Case Button
        Case 1
            If x <= 0 Or x > imgcopybutton.Width Or Y < 0 Or Y >
imgcopybutton.Height Then
                imgcopybutton.Picture = imgCopyButtonUp.Picture
            Else
                imgcopybutton.Picture = imgCopyButtonDn.Picture
            End If
        End Select
    End Sub

Private Sub imgCopyButton_MouseUp(Button As Integer, Shift As Integer, x
As Single, Y As Single)
    imgcopybutton.Picture = imgCopyButtonUp.Picture
End Sub

Private Sub imgCutButton_Click()
    imgcutbutton.Refresh
    EditCutProc
End Sub

Private Sub imgCutButton_MouseDown(Button As Integer, Shift As Integer, x
As Single, Y As Single)
    imgcutbutton.Picture = imgCutButtonDn.Picture
End Sub

Private Sub imgCutButton_MouseMove(Button As Integer, Shift As Integer, x
As Single, Y As Single)
    Select Case Button
        Case 1
            If x <= 0 Or x > imgcutbutton.Width Or Y < 0 Or Y >
imgcutbutton.Height Then
                imgcutbutton.Picture = imgCutButtonUp.Picture
            Else
                imgcutbutton.Picture = imgCutButtonDn.Picture
            End If
        End Select
    End Sub

Private Sub imgCutButton_MouseUp(Button As Integer, Shift As Integer, x As
Single, Y As Single)
    imgcutbutton.Picture = imgCutButtonUp.Picture
End Sub

Private Sub imgFileNewButton_Click()
    imgFileNewButton.Refresh
    YeniDosya
```

```
End
End
End
Attribute VB_Name = "frmIstagraf"
Attribute VB_Creatable = False
Attribute VB_Exposed = False

Private Sub imgCopyButton_Click()
    imgcopybutton.Refresh
    EditCopyProc
End Sub

Private Sub imgCopyButton_MouseDown(Button As Integer, Shift As Integer, x As Single, Y As Single)
    imgcopybutton.Picture = imgCopyButtonDn.Picture
End Sub

Private Sub imgCopyButton_MouseMove(Button As Integer, Shift As Integer, x As Single, Y As Single)
    Select Case Button
        Case 1
            If x <= 0 Or x > imgcopybutton.Width Or Y < 0 Or Y > imgcopybutton.Height Then
                imgcopybutton.Picture = imgCopyButtonUp.Picture
            Else
                imgcopybutton.Picture = imgCopyButtonDn.Picture
            End If
        End Select
    End Sub

Private Sub imgCopyButton_MouseUp(Button As Integer, Shift As Integer, x As Single, Y As Single)
    imgcopybutton.Picture = imgCopyButtonUp.Picture
End Sub

Private Sub imgCutButton_Click()
    imgcutbutton.Refresh
    EditCutProc
End Sub

Private Sub imgCutButton_MouseDown(Button As Integer, Shift As Integer, x As Single, Y As Single)
    imgcutbutton.Picture = imgCutButtonDn.Picture
End Sub

Private Sub imgCutButton_MouseMove(Button As Integer, Shift As Integer, x As Single, Y As Single)
    Select Case Button
        Case 1
            If x <= 0 Or x > imgcutbutton.Width Or Y < 0 Or Y > imgcutbutton.Height Then
                imgcutbutton.Picture = imgCutButtonUp.Picture
            Else
                imgcutbutton.Picture = imgCutButtonDn.Picture
            End If
        End Select
    End Sub

Private Sub imgCutButton_MouseUp(Button As Integer, Shift As Integer, x As Single, Y As Single)
    imgcutbutton.Picture = imgCutButtonUp.Picture
End Sub

Private Sub imgFileNewButton_Click()
    imgFileNewButton.Refresh
    YeniDosya
End Sub
```

End Sub

```
Private Sub imgFileNewButton_MouseDown(Button As Integer, Shift As Integer, x As Single, Y As Single)
    imgFileNewButton.Picture = imgFileNewButtonDn.Picture
End Sub
```

```
Private Sub imgFileNewButton_MouseMove(Button As Integer, Shift As Integer, x As Single, Y As Single)
    Select Case Button
    Case 1
        If x <= 0 Or x > imgFileNewButton.Width Or Y < 0 Or Y > imgFileNewButton.Height Then
            imgFileNewButton.Picture = imgFileNewButtonUp.Picture
        Else
            imgFileNewButton.Picture = imgFileNewButtonDn.Picture
        End If
    End Select
End Sub
```

```
Private Sub imgFileNewButton_MouseUp(Button As Integer, Shift As Integer, x As Single, Y As Single)
    imgFileNewButton.Picture = imgFileNewButtonUp.Picture
End Sub
```

```
Private Sub imgFileOpenButton_Click()
    imgFileOpenButton.Refresh
    FOpenProc
End Sub
```

```
Private Sub imgFileOpenButton_MouseDown(Button As Integer, Shift As Integer, x As Single, Y As Single)
    imgFileOpenButton.Picture = imgFileOpenButtonDn.Picture
End Sub
```

```
Private Sub imgFileOpenButton_MouseMove(Button As Integer, Shift As Integer, x As Single, Y As Single)
    Select Case Button
    Case 1
        If x <= 0 Or x > imgFileOpenButton.Width Or Y < 0 Or Y > imgFileOpenButton.Height Then
            imgFileOpenButton.Picture = imgFileOpenButtonUp.Picture
        Else
            imgFileOpenButton.Picture = imgFileOpenButtonDn.Picture
        End If
    End Select
End Sub
```

```
Private Sub imgFileOpenButton_MouseUp(Button As Integer, Shift As Integer, x As Single, Y As Single)
    imgFileOpenButton.Picture = imgFileOpenButtonUp.Picture
End Sub
```

```
Private Sub imgPasteButton_Click()
    imgPasteButton.Refresh
    EditPasteProc
End Sub
```

```
Private Sub imgPasteButton_MouseDown(Button As Integer, Shift As Integer, x As Single, Y As Single)
    imgPasteButton.Picture = imgPasteButtonDn.Picture
End Sub
```

```
Private Sub imgPasteButton_MouseMove(Button As Integer, Shift As Integer, x As Single, Y As Single)
    Select Case Button
```

```
Case 1
    If x <= 0 Or x > imgPasteButton.Width Or Y < 0 Or Y >
imgPasteButton.Height Then
        imgPasteButton.Picture = imgPasteButtonUp.Picture
    Else
        imgPasteButton.Picture = imgPasteButtonDn.Picture
    End If
End Select
End Sub

Private Sub imgPasteButton_MouseUp(Button As Integer, Shift As Integer, x
As Single, Y As Single)
    imgPasteButton.Picture = imgPasteButtonUp.Picture
End Sub

Private Sub MDIForm_Load()
Dim a As String
    ' Uygulamanın başlangıç satırı burasıdır.
    Show
    ChDir App.Path
    a = AdVer
    YeniVTYarat a
    ReDim Document(1)
    ReDim FDurum(1)

    ReDim grafik(1)
    ReDim GDurum(1)
    GDurum(1).Silindi = True
    GDurum(1).Degisti = False

    'Buraya dosya oluşturma eklenecek
    Document(1).Tag = 1
    Document(1).db.DatabaseName = a
    Document(1).db.Refresh
    FDurum(1).Degisti = False
    GetRecentFiles
    gFindDirection = 1
End Sub

Private Sub MDIForm_Unload(Cancel As Integer)
    If Not AnyPadsLeft() Then
        End
    End If
End Sub

Private Sub mnuDac_Click()
    FOpenProc
End Sub

Private Sub mnuDÇıkış_Click()
    End
End Sub

Private Sub mnuDYeni_Click()
    YeniDosya
End Sub

Private Sub mnuRecentFile_Click(Index As Integer)
    OpenFile (mnuRecentFile(Index).Caption)
    GetRecentFiles
End Sub

Private Sub mnuSArac_Click()
    OptionsToolbarProc Me
End Sub
```

```
Private Sub mnuSeçenekler_Click()  
    mnuSArac.Checked = frmIstagraf!picToolbar.Visible  
End Sub
```

```
*****  
DBDUZEN.FRM  
*****
```

VERSION 4.00

Begin VB.Form frmddb

```
    Caption           = "Adsız"  
    ClientHeight      = 5190  
    ClientLeft        = 735  
    ClientTop         = 1335  
    ClientWidth       = 7800  
    BeginProperty Font  
        name          = "MS Sans Serif"  
        charset       = 1  
        weight        = 700  
        size          = 8.25  
        underline     = 0 'False  
        italic        = 0 'False  
        strikethrough  = 0 'False
```

EndProperty

```
    Height            = 5880  
    Left              = 675  
    LinkTopic         = "Form1"  
    MDIChild          = -1 'True  
    ScaleHeight       = 702.406  
    ScaleMode         = 0 'User  
    ScaleWidth        = 13.242  
    Top               = 705  
    Visible           = 0 'False  
    Width             = 7920
```

Begin VBX.TrueGrid Table1

```
    AllowArrows      = -1 'True  
    AllowTabs        = -1 'True  
    DataSource       = "db"  
    Editable         = -1 'True  
    EditDropDown     = -1 'True  
    ExposeCellMode   = 0 'Expose upon selection  
    FetchMode        = 0 'By cell  
    HeadingHeight    = 1  
    Height           = 6615  
    HorzLines        = 2 '3D  
    Layout           = "DBDUZEN.frx":0000  
    LayoutIndex      = 1  
    Left             = 0  
    LinesPerRow      = 1  
    MarqueeUnique    = -1 'True  
    SplitPropsGlobal = -1 'True  
    SplitTabMode     = 0 'Don't tab across splits  
    TabCapture       = 0 'False  
    TabIndex         = 0  
    Top              = 0  
    UseBookmarks     = -1 'True  
    Width            = 9615  
    WrapCellPointer  = 0 'False
```

End

Begin VB.Data db

```
    Connect          = "Access"  
    DatabaseName     = "NACI.MDB"  
    Exclusive        = 0 'False  
    Height           = 270
```

```
Left          = 0
Options       = 0
ReadOnly      = 0 'False
RecordsetType = 0 'Table
RecordSource  = "Tablo 1"
Top          = 0
Width        = 2490
End
Begin VB.Menu mnuDosya
    Caption      = "&Dosya"
    WindowList   = -1 'True
    Begin VB.Menu mnuFNew
        Caption    = "&Yeni"
    End
    Begin VB.Menu mnuFOpen
        Caption     = "&Aç..."
    End
    Begin VB.Menu mnuFClose
        Caption      = "&Kapa"
    End
    Begin VB.Menu mnuFSave
        Caption       = "&Sakla"
    End
    Begin VB.Menu mnuFSaveAs
        Caption        = "A&diyla sakla..."
    End
    Begin VB.Menu mnuithal
        Caption         = "İ&thal..."
    End
    Begin VB.Menu mnuFSep
        Caption          = "-"
    End
    Begin VB.Menu mnuFExit
        Caption           = "Çı&kış"
    End
    Begin VB.Menu mnuRecentFile
        Caption            = "-"
        Index              = 0
        Visible             = 0 'False
    End
    Begin VB.Menu mnuRecentFile
        Caption             = "RecentFile1"
        Index               = 1
        Visible              = 0 'False
    End
    Begin VB.Menu mnuRecentFile
        Caption             = "RecentFile2"
        Index               = 2
        Visible              = 0 'False
    End
    Begin VB.Menu mnuRecentFile
        Caption             = "RecentFile3"
        Index               = 3
        Visible              = 0 'False
    End
    Begin VB.Menu mnuRecentFile
        Caption             = "RecentFile4"
        Index               = 4
        Visible              = 0 'False
    End
End
Begin VB.Menu mnudüzen
    Caption      = "Dü&zen"
    Begin VB.Menu mnukes
        Caption     = "&Kes"
        Shortcut    = ^X
    End
End
```

```
End
Begin VB.Menu mnukopya
    Caption      = "K&opyala"
    Shortcut     = ^C
End
Begin VB.Menu mnuEPaste
    Caption      = "&Yapıştır"
    Shortcut     = ^V
End
Begin VB.Menu mnuESep1
    Caption      = "-"
End
Begin VB.Menu mnuSerisil
    Caption      = "S&eri sil"
End
Begin VB.Menu mnusatırsil
    Caption      = "S&atır sil"
End
Begin VB.Menu mnusatırekle
    Caption      = "&Satır ekle"
End
End
Begin VB.Menu mnuArama
    Caption      = "&Arama"
Begin VB.Menu mnuara
    Caption      = "&Ara"
End
Begin VB.Menu mnusonraki
    Caption      = "Sonrakini Bul"
    Shortcut     = {F3}
End
End
Begin VB.Menu mnusecenek
    Caption      = "&Seçenekler"
Begin VB.Menu mnuOToolbar
    Caption      = "&Araç çubuğu"
End
Begin VB.Menu mnuozellik
    Caption      = "Ö&zellikler"
End
End
Begin VB.Menu mnuWindow
    Caption      = "&Pencere"
Begin VB.Menu mnuPbasamak
    Caption      = "&Basamakla"
End
Begin VB.Menu mnuPdöşe
    Caption      = "&Döşe"
End
Begin VB.Menu mnupyerleş
    Caption      = "&Simgeleri yerleştir"
End
Begin VB.Menu asdqwe
    Caption      = "-"
End
Begin VB.Menu mnugrafik
    Caption      = "&Grafik ekle"
End
End
End
Attribute VB_Name = "frmdb"
Attribute VB_Creatable = False
Attribute VB_Exposed = False
Option Explicit

Dim dbadı As String, tbadı As String, sradı As String
```



```
Dim seridegisti As Integer

Dim dba As Database, tba As Table
Sub DosyaKapa()
    dbadi = db.DatabaseName
    tbadı = db.RecordSource
    db.DatabaseName = ""
    db.RecordSource = ""
    db.Refresh
    db.Enabled = False
End Sub
Sub DosyaAc()
    db.DatabaseName = dbadi
    db.RecordSource = tbadı
    db.Refresh
    db.Enabled = True
End Sub

Private Sub Form_Load()
Set dba = OpenDatabase(db.DatabaseName)
Set tba = dba.OpenTable("Özellikler")
tba.Index = "Seri adı"
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
Dim Msg, Filename, NL
Dim Response As Integer

    If FDurum(Me.Tag).Degisti Then
        Filename = Me.Caption
        NL = Chr(10) & Chr(13)
        Select Case Response
            Case 6      ' Evet seçildi.
            Case 7      ' Hayır seçildi.
                Cancel = False
            Case 2      ' İptal seçildi.
                Cancel = True
        End Select
    End If
End Sub

Private Sub Form_Resize()
    If WindowState <> 1 And ScaleHeight <> 0 Then
        Table1.Move 0, 0, ScaleWidth, ScaleHeight
    End If
End Sub

Private Sub Form_Unload(Cancel As Integer)
    FDurum(Me.Tag).Silindi = True
    tba.Close
    dba.Close
    ' Toolbar gizle.
    If Not AnyPadsLeft() Then
        frmIstagraf!imgcutbutton.Visible = False
        frmIstagraf!imgcopybutton.Visible = False
        frmIstagraf!imgPasteButton.Visible = False
        gToolsHidden = True
        GetRecentFiles
    End If
End Sub

Private Sub mnuECopy_Click()
    EditCopyProc
End Sub

Private Sub mnuECut_Click()
```

```
        EditCutProc
    End Sub

    Private Sub mnuEDelete_Click()
        If Screen.ActiveControl.SelStart <> Len(Screen.ActiveControl.Text) Then
            If Screen.ActiveControl.SelLength = 0 Then
                Screen.ActiveControl.SelLength = 1
                If Asc(Screen.ActiveControl.SelText) = 13 Then
                    Screen.ActiveControl.SelLength = 2
                End If
            End If
            Screen.ActiveControl.SelText = ""
        End If
    End Sub

    Private Sub mnuEPaste_Click()
        EditPasteProc
    End Sub

    Private Sub mnuFClose_Click()
        Unload Me
    End Sub

    Private Sub mnuFExit_Click()
        Unload frmIstagraf
    End Sub

    Private Sub mnuFNew_Click()
        YeniDosya
    End Sub

    Private Sub mnuFOpen_Click()
        FOpenProc
    End Sub

    Private Sub mnuFSave_Click()
        Dim Dosyadı As String

        If Left(Me.Caption, 5) = "Adsız" Then
            Dosyadı = GetFileName(Dosyadı)
        Else
            Dosyadı = Me.Caption
        End If
        If Dosyadı <> "" Then
            DosyaKapa
            SaveFileAs dbadı, Dosyadı
            dbadı = Dosyadı
            DosyaAc
        End If
    End Sub

    Private Sub mnuFSaveAs_Click()
        Dim SaveFileName As String
        Dim DefaultName As String
        DefaultName = Me.Caption
        If Left(Me.Caption, 8) = "Untitled" Then
            SaveFileName = GetFileName("adsiz.mdb")
            If SaveFileName <> "" Then SaveFileAs SaveFileName, SaveFileName
            UpdateFileMenu (SaveFileName)
        Else
            SaveFileName = GetFileName(DefaultName)
            If SaveFileName <> "" Then SaveFileAs SaveFileName, SaveFileName
            UpdateFileMenu (SaveFileName)
        End If
    End Sub
```

```
End Sub

Private Sub mnugrafik_Click()
Dim i As Integer, ff As Integer
Dim ArrayCount As Integer

    ArrayCount = UBound(Document)

    For i = 1 To ArrayCount
        If GDurum(i).Silindi Then
            ff = i
            GDurum(i).Silindi = False
            Exit For
        End If
    Next
    If i > ArrayCount Then
        ReDim Preserve grafik(ArrayCount + 1)
        ReDim Preserve GDurum(ArrayCount + 1)
        ff = UBound(grafik)
    End If
    grafik(ff).Tag = 1
    GDurum(ff).Degisti = False
    grafik(ff).txt(0) = db.DatabaseName
    grafik(ff).txt(1) = db.RecordSource
    grafik(ff).Caption = "Grafik:" & ff
    grafik(ff).lblcontrol = "*"
End Sub

Private Sub mnuOToolbar_Click()
    OptionsToolbarProc Me
End Sub

Private Sub mnuozellik_Click()
sradi = Table1.ColumnIndex
DosyaKapa
Form1.dbadi = dbadi
Form1.tbadi = tbadi
Form1.sradi = sradi
db.DatabaseName = ""
db.RecordSource = ""
db.Refresh
Table1.Refresh
Form1.Show 1
db.DatabaseName = dbadi
db.RecordSource = tbadi
db.Refresh
Table1.Refresh
Table1.EditMask(Table1.ColumnIndex) = seriformat1
End Sub

Private Sub mnuPbasamak_Click()
    frmIstagraf.Arrange vbCascade
End Sub

Private Sub mnuPdöşe_Click()
    frmIstagraf.Arrange vbTileHorizontal
End Sub

Private Sub mnupyerleş_Click()
    frmIstagraf.Arrange vbArrangeIcons
End Sub

Private Sub mnuRecentFile_Click(Index As Integer)
    OpenFile (mnuRecentFile(Index).Caption)
    GetRecentFiles
End Sub
```

```
Private Sub mnusatirekle_Click()  
Dim satirno As Long  
Dim db1 As Database  
db.Recordset.MoveLast  
satirno = db.Recordset("i")  
db.Recordset.AddNew  
db.Recordset("i") = satirno + 1  
db.Recordset.Update  
Table1.Refresh  
End Sub
```

```
Private Sub mnusatirsil_Click()  
Dim satirno As Long, i As Integer  
Dim db1 As Database  
satirno = db.Recordset("i")  
db.Recordset.Delete  
Set db1 = OpenDatabase("NACI.MDB")  
db1.Execute "UPDATE [Tablo 1] SET [i]= [i]-1 where [i]>" & satirno & ";"  
db1.Close  
Table1.Refresh  
End Sub
```

```
Private Sub mnusecenek_Click()  
mnuOToolbar.Checked = frmIstagraf!picToolbar.Visible  
End Sub
```

```
Private Sub mnuSeriSil_Click()  
Dim seriadi As String, i As Integer  
Dim db1 As Database, tdl As TableDef  
seriadi = Table1.ColumnNames(Table1.ColumnIndex)  
If seriadi <> "i" Then  
i = MsgBox(seriadi & " serisini silmek istediğinizden emin misiniz  
?", 257, "Dikkat !")  
If i = 1 Then  
DosyaKapa  
Set db1 = OpenDatabase(dbadi)  
Set tdl = db1.TableDefs(tbad1)  
tdl.Fields.Delete seriadi  
db1.Close  
DosyaAc  
End If  
End If  
End Sub
```

```
Private Sub mnuSFind_Click()  
If Me!Text1.SelText <> "" Then  
frmAra!Text1.Text = Me!Text1.SelText  
Else  
frmAra!Text1.Text = gFindString  
End If  
gFirstTime = True  
If (gFindCase) Then  
frmAra.chkCase = 1  
End If  
frmAra.Show  
End Sub
```

```
Private Sub mnuSFindNext_Click()  
If Len(gFindString) > 0 Then  
FindIt  
Else  
mnuSFind_Click  
End If  
End Sub
```

```
Private Sub Table1_Change()  
    FDurum(Me.Tag).Degisti = True  
End Sub
```

```
Private Sub Table1_ColumnChange()  
    If seridegisti Then  
        tba.Seek "=", LCASE(Table1.ColumnName(Table1.ColumnIndex))  
        If tba.NoMatch Then  
            tba.AddNew  
            tba("Seri adı") = Table1.ColumnName(Table1.ColumnIndex)  
        Else  
            tba.Edit  
        End If  
        tba("Ortalama") = 0  
        tba("GOrtalama") = 0  
        tba("HOrtalama") = 0  
        tba("KOrtalama") = 0  
        tba("OrtSapma") = 0  
        tba("StdSapma") = 0  
        tba("StdSapma") = 0  
        tba("Korel Y Serisi") = ""  
        tba("Korel") = 0  
        tba.Update  
        seridegisti = False  
    End If  
End Sub
```

```
Private Sub Table1_EnterEdit()  
    seridegisti = True  
End Sub
```

OZELLIK.FRM

VERSION 4.00

```
Begin VB.Form Form1  
    BorderStyle      = 3   'Fixed Dialog  
    Caption          = "Özellikler"  
    ClientHeight     = 4050  
    ClientLeft       = 1095  
    ClientTop        = 1485  
    ClientWidth      = 5055  
    Height           = 4455  
    Left             = 1035  
    LinkTopic        = "Form1"  
    MaxButton        = 0   'False  
    MinButton        = 0   'False  
    ScaleHeight      = 4050  
    ScaleWidth       = 5055  
    ShowInTaskbar    = 0   'False  
    Top              = 1140  
    Width            = 5175  
    Begin VB.CommandButton cmd  
        Caption      = "&Tamam"  
        Height       = 315  
        Index        = 0  
        Left         = 300  
        TabIndex     = 20  
        Top          = 3660  
        Width        = 1335  
    End  
    Begin VB.CommandButton cmd  
        Caption      = "&Uygula"  
        Height       = 315
```

```
Index          = 2
Left           = 3420
TabIndex       = 19
Top            = 3660
Width          = 1335
End
Begin VB.CommandButton cmdvazgeç
Caption        = "&Vazgeç"
Height        = 315
Left          = 1860
TabIndex       = 18
Top           = 3660
Width         = 1335
End
Begin TabDlg.SSTab SSTab1
Height        = 3435
Left          = 60
TabIndex       = 16
Top           = 120
Width         = 4935
_version      = 65536
_extentx      = 8705
_extenty      = 6059
_stockprops   = 15
caption       = "Görünüm"
BeginProperty font {FB8F0823-0164-101B-84ED-08002B2EC713}
name          = "Arial"
charset       = 1
weight        = 400
size          = 8.25
underline     = 0 'False
italic        = 0 'False
striketrough  = 0 'False
EndProperty
tabsparrow    = 3
tab           = 1
taborientation = 0
tabs          = 3
style         = 1
tabmaxwidth   = 0
tabheight     = 529
tabcaption(0) = "Genel"
tab(0).controlcount= 11
tab(0).controlenabled= 0 'False
tab(0).control(0)= "lbl(0)"
tab(0).control(1)= "lbl(1)"
tab(0).control(2)= "lbl(2)"
tab(0).control(3)= "lbl(3)"
tab(0).control(4)= "lbl(4)"
tab(0).control(5)= "txtname"
tab(0).control(6)= "txtformat"
tab(0).control(7)= "txtformül"
tab(0).control(8)= "cmbalantipi"
tab(0).control(9)= "txtsize"
tab(0).control(10)= "Frame1"
tabcaption(1) = "Görünüm"
tab(1).controlcount= 4
tab(1).controlenabled= -1 'True
tab(1).control(0)= "frmort(0)"
tab(1).control(1)= "frmort(1)"
tab(1).control(2)= "frmort(2)"
tab(1).control(3)= "Command1"
tabcaption(2) = "Yazıtipi"
tab(2).controlcount= 2
tab(2).controlenabled= 0 'False
tab(2).control(0)= "Label1"
```

```
tab(2).control(1)= "cmbyazıtipi"
Begin VB.Frame Frame1
    Caption        = "Yaslama"
    Height         = 1215
    Left           = -71880
    TabIndex       = 45
    Top            = 600
    Width          = 1575
    Begin VB.OptionButton Option1
        Caption      = "Ortalı"
        Height       = 255
        Index        = 2
        Left         = 120
        TabIndex     = 48
        Top          = 840
        Width        = 735
    End
    Begin VB.OptionButton Option1
        Caption      = "Sola"
        Height       = 255
        Index        = 1
        Left         = 120
        TabIndex     = 47
        Top          = 540
        Width        = 735
    End
    Begin VB.OptionButton Option1
        Caption      = "Sağa"
        Height       = 255
        Index        = 0
        Left         = 120
        TabIndex     = 46
        Top          = 240
        Width        = 735
    End
End
Begin VB.CommandButton Command1
    Caption        = "&Bul"
    Height         = 315
    Left           = 3780
    TabIndex       = 44
    Top            = 1860
    Width          = 975
End
Begin VB.Frame frmort
    Caption        = "İlişkiler"
    Height         = 915
    Index          = 2
    Left           = 120
    TabIndex       = 39
    Top            = 2400
    Width          = 4695
    Begin VB.TextBox ilişki
        Height       = 285
        Index        = 0
        Left         = 120
        TabIndex     = 12
        Top          = 480
        Width        = 1035
    End
    Begin VB.TextBox ilişki
        Height       = 285
        Index        = 3
        Left         = 3540
        TabIndex     = 15
        Top          = 480
    End
End
```

```
        Width          = 1035
    End
    Begin VB.TextBox ilişki
        Height          = 285
        Index           = 2
        Left            = 2400
        TabIndex        = 14
        Top             = 480
        Width           = 1035
    End
    Begin VB.TextBox ilişki
        Height          = 285
        Index           = 1
        Left            = 1260
        TabIndex        = 13
        Top             = 480
        Width           = 1035
    End
    Begin VB.Label lbl2
        Caption          = "Y Serisi"
        Height          = 195
        Index            = 7
        Left            = 120
        TabIndex        = 43
        Top             = 240
        Width           = 525
    End
    Begin VB.Label lbl2
        Caption          = "Korelasyon K."
        Height          = 195
        Index            = 10
        Left            = 3540
        TabIndex        = 42
        Top             = 240
        Width           = 975
    End
    Begin VB.Label lbl2
        Caption          = "B. Katsayısı"
        Height          = 195
        Index            = 9
        Left            = 2400
        TabIndex        = 41
        Top             = 240
        Width           = 810
    End
    Begin VB.Label lbl2
        Caption          = "B. İndeksi"
        Height          = 195
        Index            = 8
        Left            = 1260
        TabIndex        = 40
        Top             = 240
        Width           = 630
    End
End
Begin VB.Frame frmort
    Caption          = "Sapmalar"
    Height          = 915
    Index           = 1
    Left            = 120
    TabIndex        = 35
    Top             = 1380
    Width           = 3555
    Begin VB.TextBox sapma
        Height          = 285
        Index           = 0
```



```
        Left           = 120
        TabIndex       = 9
        Top            = 480
        Width          = 1035
    End
    Begin VB.TextBox sapma
        Height          = 285
        Index           = 1
        Left            = 1260
        TabIndex        = 10
        Top             = 480
        Width           = 1035
    End
    Begin VB.TextBox sapma
        Height          = 285
        Index           = 2
        Left            = 2400
        TabIndex        = 11
        Top             = 480
        Width           = 1035
    End
    Begin VB.Label lbl2
        Caption          = "Ortalama"
        Height           = 195
        Index            = 4
        Left             = 120
        TabIndex         = 38
        Top              = 240
        Width            = 630
    End
    Begin VB.Label lbl2
        Caption          = "Standart"
        Height           = 195
        Index            = 5
        Left             = 1260
        TabIndex         = 37
        Top              = 240
        Width            = 600
    End
    Begin VB.Label lbl2
        Caption          = "D. Katsayısı"
        Height           = 195
        Index            = 6
        Left             = 2400
        TabIndex         = 36
        Top              = 240
        Width            = 825
    End
End
Begin VB.Frame frmort
    Caption            = "Ortalamlar"
    Height             = 915
    Index              = 0
    Left               = 120
    TabIndex           = 30
    Top                = 420
    Width              = 4695
    Begin VB.TextBox ort
        Height          = 285
        Index           = 3
        Left            = 3540
        TabIndex        = 8
        Top             = 480
        Width           = 1035
    End
    Begin VB.TextBox ort
```

```

        Height      = 285
        Index       = 2
        Left        = 2400
        TabIndex    = 7
        Top         = 480
        Width       = 1035
    End
    Begin VB.TextBox ort
        Height      = 285
        Index       = 1
        Left        = 1260
        TabIndex    = 6
        Top         = 480
        Width       = 1035
    End
    Begin VB.TextBox ort
        Height      = 285
        Index       = 0
        Left        = 120
        TabIndex    = 5
        Top         = 480
        Width       = 1035
    End
    Begin VB.Label lbl2
        Caption      = "Kareli"
        Height       = 195
        Index        = 3
        Left         = 3540
        TabIndex     = 34
        Top          = 240
        Width        = 600
    End
    Begin VB.Label lbl2
        Caption      = "Harmonik"
        Height       = 195
        Index        = 2
        Left         = 2400
        TabIndex     = 33
        Top          = 240
        Width        = 675
    End
    Begin VB.Label lbl2
        Caption      = "Geometrik"
        Height       = 195
        Index        = 1
        Left         = 1260
        TabIndex     = 32
        Top          = 240
        Width        = 720
    End
    Begin VB.Label lbl2
        Caption      = "Aritmetik"
        Height       = 195
        Index        = 0
        Left         = 120
        TabIndex     = 31
        Top          = 240
        Width        = 600
    End
End
Begin VB.TextBox txtsize
    Enabled        = 0    'False
    Height         = 285
    Left           = -74160
    TabIndex       = 2
    Top            = 1500

```

```
Width          = 2115
End
Begin VB.ComboBox cmbalantipi
    Enabled      = 0    'False
    Height       = 300
    Left        = -74160
    Style       = 2    'Dropdown List
    TabIndex     = 1
    Top         = 1080
    Width       = 2115
End
Begin VB.TextBox txtformül
    Height       = 285
    Left        = -74160
    TabIndex     = 4
    Top         = 2760
    Width       = 3855
End
Begin VB.ComboBox cmbyaşıtipi
    Height       = 300
    Left        = -74160
    TabIndex     = 22
    Top         = 480
    Width       = 1875
End
Begin VB.TextBox txtformat
    Height       = 285
    Left        = -74160
    TabIndex     = 3
    Top         = 2340
    Width       = 3855
End
Begin VB.TextBox txtname
    Enabled      = 0    'False
    Height       = 285
    Left        = -74160
    TabIndex     = 0
    Top         = 660
    Width       = 2115
End
Begin VB.Label lbl
    Caption      = "Uzunluğu"
    Height       = 195
    Index       = 4
    Left        = -74880
    TabIndex     = 29
    Top         = 1560
    Width       = 675
End
Begin VB.Label lbl
    Caption      = "Tipi"
    Height       = 195
    Index       = 3
    Left        = -74880
    TabIndex     = 25
    Top         = 1140
    Width       = 615
End
Begin VB.Label lbl
    Caption      = "Formül"
    Height       = 195
    Index       = 2
    Left        = -74820
    TabIndex     = 24
    Top         = 2820
    Width       = 615
```

```
End
Begin VB.Label Label1
    Caption      = "Yazıtipi"
    Height       = 195
    Left         = -74820
    TabIndex     = 23
    Top          = 540
    Width        = 495
End
Begin VB.Label lbl
    Caption      = "Formatı"
    Height       = 195
    Index        = 1
    Left         = -74820
    TabIndex     = 21
    Top          = 2400
    Width        = 615
End
Begin VB.Label lbl
    Caption      = "Alan adı"
    Height       = 195
    Index        = 0
    Left         = -74880
    TabIndex     = 17
    Top          = 720
    Width        = 615
End
End
Begin VB.Label sradi
    Height       = 195
    Left         = 5400
    TabIndex     = 28
    Top          = 840
    Width        = 1155
End
Begin VB.Label tbadi
    Height       = 195
    Left         = 5400
    TabIndex     = 27
    Top          = 480
    Width        = 1155
End
Begin VB.Label dbadi
    Height       = 195
    Left         = 5400
    TabIndex     = 26
    Top          = 120
    Width        = 1155
End
End
Attribute VB_Name = "Form1"
Attribute VB_Creatable = False
Attribute VB_Exposed = False
Option Explicit

Dim db As Database, td As TableDef, fd As Field

Dim tböz As Table

Private Sub cmbalantipi_Click()
    cmbalantipi.Tag = "1"
End Sub

Private Sub cmd_Click(Index As Integer)
    Dim asd As Double
    If txtformül.Tag = "1" And Trim(txtformül) <> "" Then
```

```
Set dbformül = OpenDatabase(dbadı)
Set tbgenel = dbformül.OpenTable(tbadı)
Set tbgenel1 = dbformül.OpenTable(tbadı)
tbgenel.Index = "i"
tbgenel1.Index = "i"
Set tbözel = dbformül.OpenTable("Özellikler")
tbözel.Index = "Seri adı"
tbgenel.MoveFirst
'sradı = ""
Do
    tbgenel.Edit
    asd = evaluate(txtformül)
    tbgenel(sradı) = asd
    tbgenel.Update
    tbgenel.MoveNext
Loop Until tbgenel.EOF
End If

Set db = OpenDatabase(dbadı.Caption)
Set tböz = db.OpenTable("Özellikler")
tböz.Index = "Seri adı"
tböz.Seek "=", sradı
If tböz.NoMatch Then
    tböz.AddNew
    tböz("Seri adı") = sradı
Else
    tböz.Edit
End If

tböz("Ortalama") = ort(0)
tböz("GOrtalama") = ort(1)
tböz("HOrtalama") = ort(2)
tböz("KOrtalama") = ort(3)
tböz("OrtSapma") = sapma(0)
tböz("StdSapma") = sapma(1)
tböz("StdSapma") = sapma(2)
tböz("Korel Y Serisi") = ilişki(0)
tböz("Korel") = ilişki(3)

tböz("Formatı") = txtformat
tböz("Formül") = Trim(txtformül)

tböz.Update

tböz.Close
db.Close
seriformatı = txtformat
Unload Me
End Sub

Private Sub cmdvazgeç_Click()
Unload Me
End Sub

Private Sub Form_Activate()
Dim i As Integer
Set db = OpenDatabase(dbadı.Caption)
txtname = db.TableDefs(tbadı).Fields(sradı).Name
i = db.TableDefs(tbadı).Fields(sradı).Type
If i < 9 Then
    cmbalantipi.ListIndex = i - 1
Else
    cmbalantipi.ListIndex = i - 2
End If
txtsize = db.TableDefs(tbadı).Fields(sradı).Size
txtname.Tag = ""
```

```
cmbalantipi.Tag = ""
txtsize.Tag = ""

Set tböz = db.OpenTable("Özellikler")
tböz.Index = "Seri adı"
tböz.Seek "=", sradı
If tböz.NoMatch Then
    tböz.AddNew
    tböz("Seri adı") = sradı
    tböz.Update
End If
ort(0) = numcpy(tböz("Ortalama"))
ort(1) = numcpy(tböz("GOrtalama"))
ort(2) = numcpy(tböz("HOrtalama"))
ort(3) = numcpy(tböz("KOrtalama"))
sapma(0) = numcpy(tböz("OrtSapma"))
sapma(1) = numcpy(tböz("StdSapma"))
If numcpy(tböz("StdSapma")) > 0 Then sapma(2) = numcpy(tböz("OrtSapma")) /
numcpy(tböz("StdSapma")) * 100 Else sapma(2) = 0
ilişki(0) = strcpy(tböz("Korel Y Serisi"))
ilişki(3) = numcpy(tböz("Korel"))

txtformat = strcpy(tböz("Formatı"))
txtformül = strcpy(tböz("Formül"))

tböz.Close
db.Close
End Sub
Private Sub Form_Load()
Dim i As Integer

For i = 1 To Screen.FontCount - 1
    cmbyazıtipi.AddItem Screen.Fonts(i)
Next
cmbalantipi.AddItem "Mantıksal"
cmbalantipi.AddItem "Bayt"
cmbalantipi.AddItem "Tamsayı"
cmbalantipi.AddItem "Uzun Tamsayı"
cmbalantipi.AddItem "Parasal"
cmbalantipi.AddItem "Tek Duyarlıklılı"
cmbalantipi.AddItem "Çift Duyarlıklılı"
cmbalantipi.AddItem "Tarih/Saat"
cmbalantipi.AddItem "Uzun İkili"
cmbalantipi.AddItem "Metin"
cmbalantipi.AddItem "Uzun Metin"

End Sub

Private Sub txtformül_Change()
    txtformül.Tag = "1"
End Sub

Private Sub txtsize_Change()
    txtsize.Tag = "1"
End Sub
```

```
*****
FRGEGRF.FRM
*****
```

```
VERSION 4.00
Begin VB.Form Form2
    Caption       =   "Eğrisel Grafik Özellikleri"
    ClientHeight  =   4230
    ClientLeft    =   1065
```

```
ClientTop      = 1470
ClientWidth    = 6720
Height         = 4635
Left           = 1005
LinkTopic      = "Form2"
ScaleHeight    = 4230
ScaleWidth     = 6720
Top            = 1125
Width          = 6840
Begin VB.CommandButton cmd
    Caption      = "&Vazgeç"
    Height       = 315
    Index        = 1
    Left         = 1920
    TabIndex     = 2
    Top          = 3660
    Width        = 1335
End
Begin VB.CommandButton cmd
    Caption      = "&Uygula"
    Height       = 315
    Index        = 2
    Left         = 3480
    TabIndex     = 1
    Top          = 3660
    Width        = 1335
End
Begin VB.CommandButton cmd
    Caption      = "&Tamam"
    Height       = 315
    Index        = 0
    Left         = 360
    TabIndex     = 0
    Top          = 3660
    Width        = 1335
End
Begin TabDlg.SSTab SSTab1
    Height       = 3435
    Left        = 120
    TabIndex     = 3
    Top         = 120
    Width       = 4935
    _version    = 65536
    _extentx    = 8705
    _extenty    = 6059
    _stockprops = 15
    caption     = "Genel"
    BeginProperty font {FB8F0823-0164-101B-84ED-08002B2EC713}
        name      = "Arial"
        charset   = 1
        weight    = 400
        size      = 8.25
        underline  = 0 'False
        italic    = 0 'False
        strikethrough = 0 'False
    EndProperty
    tabsperrow  = 3
    tab         = 0
    taborientation = 0
    tabs        = 3
    style       = 1
    tabmaxwidth = 0
    tabheight   = 529
    tabcaption(0) = "Genel"
    tab(0).controlcount= 2
    tab(0).controlenabled= -1 'True
```

```
tab(0).control(0)= "lbl(3)"
tab(0).control(1)= "cmbgraftürü"
tabcaption(1) = "Görünüm"
tab(1).controlcount= 0
tab(1).controlenabled= 0 'False
tabcaption(2) = "Yazıtipi"
tab(2).controlcount= 2
tab(2).controlenabled= 0 'False
tab(2).control(0)= "cmbyazıtipi"
tab(2).control(1)= "Label1"
Begin VB.ComboBox cmbgraftürü
    Height      = 300
    Left        = 660
    Style       = 2 'Dropdown List
    TabIndex    = 5
    Top         = 720
    Width       = 1515
End
Begin VB.ComboBox cmbyazıtipi
    Height      = 300
    Left        = -74160
    TabIndex    = 4
    Top         = 480
    Width       = 1875
End
Begin VB.Label lbl
    Caption     = "Türü"
    Height      = 195
    Index       = 3
    Left        = 120
    TabIndex    = 7
    Top         = 780
    Width       = 330
End
Begin VB.Label Label1
    Caption     = "Yazıtipi"
    Height      = 195
    Left        = -74820
    TabIndex    = 6
    Top         = 540
    Width       = 495
End
End
End
Attribute VB_Name = "Form2"
Attribute VB_Creatable = False
Attribute VB_Exposed = False
Option Explicit

Private Sub Form_Load()

End Sub
```

```
*****
GRAFIK.FRM
*****
```

```
VERSION 4.00
Begin VB.Form frmgraph
    Appearance  = 0 'Flat
    BackColor   = &H00C0C0C0&
    ClientHeight = 4305
    ClientLeft  = 1815
    ClientTop   = 1035
    ClientWidth = 4560
```



```
BeginProperty Font
    name           = "MS Sans Serif"
    charset        = 1
    weight         = 700
    size           = 8.25
    underline      = 0 'False
    italic         = 0 'False
    strikethrough  = 0 'False
EndProperty
ForeColor        = &H80000008&
Height           = 4710
Left             = 1755
LinkTopic        = "frmgraph"
MDIChild         = -1 'True
MinButton        = 0 'False
ScaleHeight      = 4305
ScaleWidth       = 4560
Top              = 690
Width            = 4680
Begin VB.CommandButton cmdyazdir
    Caption        = "&Yazdır"
    Height         = 315
    Left           = 2580
    TabIndex       = 12
    Top            = 3660
    Width          = 1095
End
Begin VB.CommandButton cmd
    Caption        = "&Tamam"
    Height         = 315
    Index          = 0
    Left           = 180
    TabIndex       = 10
    Top            = 3660
    Width          = 1095
End
Begin VB.CommandButton cmd
    Caption        = "&Vazgeç"
    Height         = 315
    Index          = 1
    Left           = 1380
    TabIndex       = 11
    Top            = 3660
    Width          = 1095
End
Begin TabDlg.SSTab SSTab1
    Height         = 3435
    Left           = 120
    TabIndex       = 14
    Top            = 120
    Width          = 3615
    _version       = 65536
    _extentx       = 6376
    _extenty       = 6059
    _stockprops    = 15
    caption        = "Yazıtipi"
    forecolor      = -2147483630
    backcolor      = -2147483633
    BeginProperty font {FB8F0823-0164-101B-84ED-08002B2EC713}
        name           = "Arial"
        charset        = 1
        weight         = 400
        size           = 8.25
        underline      = 0 'False
        italic         = 0 'False
        strikethrough  = 0 'False
```

```
EndProperty
tabsperrrow      = 5
tab              = 4
taborientation  = 0
tabs            = 5
style           = 1
tabmaxwidth     = 0
tabheight       = 529
tabcaption(0)   = "Genel"
tab(0).controlcount= 9
tab(0).controlenabled= 0 'False
tab(0).control(0)= "cmbaluttürü"
tab(0).control(1)= "txt(1)"
tab(0).control(2)= "txt(0)"
tab(0).control(3)= "cmbgraftürü"
tab(0).control(4)= "Label2"
tab(0).control(5)= "lbl(9)"
tab(0).control(6)= "lbl(1)"
tab(0).control(7)= "lbl(0)"
tab(0).control(8)= "lbl(3)"
tabcaption(1)   = "Seriler"
tab(1).controlcount= 6
tab(1).controlenabled= 0 'False
tab(1).control(0)= "txt(2)"
tab(1).control(1)= "lst"
tab(1).control(2)= "cmbseriler"
tab(1).control(3)= "lbl(2)"
tab(1).control(4)= "lbl(5)"
tab(1).control(5)= "lbl(4)"
tabcaption(2)   = "Başlıklar"
tab(2).controlcount= 6
tab(2).controlenabled= 0 'False
tab(2).control(0)= "txt(5)"
tab(2).control(1)= "txt(4)"
tab(2).control(2)= "txt(3)"
tab(2).control(3)= "lbl(8)"
tab(2).control(4)= "lbl(7)"
tab(2).control(5)= "lbl(6)"
tabcaption(3)   = "İstatistik"
tab(3).controlcount= 2
tab(3).controlenabled= 0 'False
tab(3).control(0)= "ckh(1)"
tab(3).control(1)= "ckh(0)"
tabcaption(4)   = "Yazıtipi"
tab(4).controlcount= 2
tab(4).controlenabled= -1 'True
tab(4).control(0)= "Label1"
tab(4).control(1)= "cmbyazıtipi"
Begin VB.CheckBox ckh
    Caption      = "Standart sapma"
    Height       = 255
    Index        = 1
    Left         = -74640
    TabIndex     = 29
    Top          = 1260
    Width        = 1635
End
Begin VB.CheckBox ckh
    Caption      = "Ortalama"
    Height       = 255
    Index        = 0
    Left         = -74640
    TabIndex     = 28
    Top          = 840
    Width        = 1035
End
```

```
Begin VB.ComboBox cmbyazıtipi
    Height      = 300
    Left        = 840
    TabIndex    = 26
    Top         = 600
    Width       = 1875
End
Begin VB.ComboBox cmbalıtıurü
    Height      = 300
    Left        = -73920
    Style       = 2 'Dropdown List
    TabIndex    = 3
    Top         = 2580
    Width       = 2295
End
Begin VB.TextBox txt
    Height      = 285
    Index       = 5
    Left        = -73380
    TabIndex    = 9
    Top         = 1800
    Width       = 1815
End
Begin VB.TextBox txt
    Height      = 285
    Index       = 4
    Left        = -73380
    TabIndex    = 8
    Top         = 1320
    Width       = 1815
End
Begin VB.TextBox txt
    Height      = 285
    Index       = 3
    Left        = -73380
    TabIndex    = 7
    Top         = 840
    Width       = 1815
End
Begin VB.TextBox txt
    Height      = 285
    Index       = 2
    Left        = -74100
    TabIndex    = 5
    Top         = 1140
    Width       = 1815
End
Begin VB.ListBox lst
    Height      = 1395
    Left        = -74100
    TabIndex    = 6
    Top         = 1620
    Width       = 1815
End
Begin VB.ComboBox cmbseriler
    Height      = 300
    Left        = -74100
    TabIndex    = 4
    Top         = 600
    Width       = 1815
End
Begin VB.TextBox txt
    Height      = 285
    Index       = 1
    Left        = -73920
    TabIndex    = 1
```

```
Top           = 1260
Width         = 2295
End
Begin VB.TextBox txt
Height       = 285
Index       = 0
Left        = -73920
TabIndex    = 0
Top         = 840
Width       = 2295
End
Begin VB.ComboBox cmbgrafiürü
Height       = 300
Left        = -73920
Style       = 2 'Dropdown List
TabIndex    = 2
Top         = 2160
Width       = 2295
End
Begin VB.Label Label2
Caption      = "Grafiğin"
Height      = 195
Left        = -74820
TabIndex    = 30
Top         = 1920
Width       = 675
End
Begin VB.Label Label1
Caption      = "Yazıtipi"
Height      = 195
Left        = 120
TabIndex    = 27
Top         = 660
Width       = 615
End
Begin VB.Label lbl
Caption      = "Alt türü"
Height      = 195
Index       = 9
Left        = -74820
TabIndex    = 24
Top         = 2640
Width       = 630
End
Begin VB.Label lbl
Caption      = "Y eksen başlığı"
Height      = 195
Index       = 8
Left        = -74820
TabIndex    = 23
Top         = 1860
Width       = 1350
End
Begin VB.Label lbl
Caption      = "X eksen başlığı"
Height      = 195
Index       = 7
Left        = -74820
TabIndex    = 22
Top         = 1380
Width       = 1350
End
Begin VB.Label lbl
Caption      = "Grafik başlığı"
Height      = 195
Index       = 6
```

```
Left          = -74820
TabIndex      = 21
Top           = 900
Width         = 1125
End
Begin VB.Label lbl
Caption       = "Y serisi"
Height       = 195
Index        = 2
Left         = -74820
TabIndex     = 20
Top          = 1620
Width        = 630
End
Begin VB.Label lbl
Caption       = "X serisi"
Height       = 195
Index        = 5
Left         = -74820
TabIndex     = 19
Top          = 1200
Width        = 630
End
Begin VB.Label lbl
Caption       = "Seriler"
Height       = 195
Index        = 4
Left         = -74820
TabIndex     = 18
Top          = 660
Width        = 555
End
Begin VB.Label lbl
Caption       = "Tablo adı"
Height       = 195
Index        = 1
Left         = -74820
TabIndex     = 17
Top          = 1320
Width        = 810
End
Begin VB.Label lbl
Caption       = "Dosya adı"
Height       = 195
Index        = 0
Left         = -74820
TabIndex     = 16
Top          = 900
Width        = 855
End
Begin VB.Label lbl
Caption       = "Türü"
Height       = 195
Index        = 3
Left         = -74820
TabIndex     = 15
Top          = 2220
Width        = 390
End
End
Begin VB.Label lblcontrol
Height       = 255
Left         = 4140
TabIndex     = 25
Top          = 180
Width        = 375
```

```
End
Begin VB.Line lin
    Visible      = 0    'False
    X1           = 4020
    X2           = 4035
    Y1           = 4020
    Y2           = 4035
End
Begin GraphLib.Graph grafik
    Height       = 4155
    Left         = 60
    TabIndex     = 13
    Top          = 0
    Visible      = 0    'False
    Width        = 3975
    _version     = 65536
    _extentx     = 7011
    _extenty     = 7329
    _stockprops  = 96
    autoinc      = 0
    graphtype    = 9
    legendstyle  = 1
    linestats    = 5
    numpoints    = 2
    randomdata   = 0
    colordata    = 0
    extradata    = 0
    extradata[]  = 0
    fontfamily   = 4
    fontsize     = 4
    fontsize[0]  = 200
    fontsize[1]  = 150
    fontsize[2]  = 100
    fontsize[3]  = 100
    fontstyle    = 4
    graphdata    = 1
    graphdata[]  = 2
    graphdata[0,0] = 0
    graphdata[0,1] = 0
    labeltext    = 0
    legendtext   = 0
    patterndata  = 0
    symboldata   = 0
    xposdata    = 0
    xposdata[]  = 0
End
End
Attribute VB_Name = "frmgraph"
Attribute VB_Creatable = False
Attribute VB_Exposed = False
Option Explicit

Dim cizik As Integer

Dim db As Database, tb As Table

Private Sub cmbgrafter_Click()
    cmbalfter.Clear
    Select Case cmbgrafter.ListIndex
    Case 0, 1
        cmbalfter.AddItem "Boş", 0
        cmbalfter.AddItem "Etiketsiz", 1
        cmbalfter.AddItem "Renkli etiketli", 2
        cmbalfter.AddItem "Renkli etiketli ve çizgisiz", 3
        cmbalfter.AddItem "% etiketli", 4
        cmbalfter.AddItem "% etiketli ve çizgisiz", 5
    
```

```
cmbalttürü.AddItem "% Renkli etiketli", 6
cmbalttürü.AddItem "% Renkli etiketli ve çizgisiz", 7
Case 2
cmbalttürü.AddItem "Boş", 0
cmbalttürü.AddItem "Yatay", 1
cmbalttürü.AddItem "Yığın", 2
cmbalttürü.AddItem "Yatay yığın", 3
cmbalttürü.AddItem "Yığın %", 4
cmbalttürü.AddItem "Yatay yığın %", 5
Case 3
cmbalttürü.AddItem "Boş", 0
cmbalttürü.AddItem "Yatay", 1
cmbalttürü.AddItem "Yığın", 2
cmbalttürü.AddItem "Yatay yığın", 3
cmbalttürü.AddItem "Yığın %", 4
cmbalttürü.AddItem "Yatay yığın %", 5
cmbalttürü.AddItem "Z-Sıralı", 6
cmbalttürü.AddItem "Yatay Z-Sıralı", 7
Case 4
cmbalttürü.AddItem "Boş", 0
cmbalttürü.AddItem "Ara çubuklu", 1
Case 5, 6, 9
cmbalttürü.AddItem "Boş", 0
cmbalttürü.AddItem "Sembollü", 1
cmbalttürü.AddItem "Göstergeli", 2
cmbalttürü.AddItem "Göstergeli ve Sembollü", 3
cmbalttürü.AddItem "Çizgili", 4
cmbalttürü.AddItem "Çizgili ve Sembollü", 5
cmbalttürü.AddItem "Çizgili ve Sticks", 6
cmbalttürü.AddItem "Çizgili, Göstergeli ve Sembollü", 7
Case 7
cmbalttürü.AddItem "Boş", 0
cmbalttürü.AddItem "Mutlak", 1
cmbalttürü.AddItem "Yüzdeli", 2
Case 8
cmbalttürü.AddItem "Boş", 0
Case 10
cmbalttürü.AddItem "Boş", 0
cmbalttürü.AddItem "No Close Bar", 1
cmbalttürü.AddItem "No High-Low Bars", 2
cmbalttürü.AddItem "No Bars", 3
End Select
cmbalttürü.ListIndex = 0

End Sub

Private Sub cmbseriler_Click()
lst.AddItem cmbseriler.List(cmbseriler.ListIndex)
End Sub

Private Sub cmd_Click(Index As Integer)
Dim i As Integer, j As Integer
grafik.NumSets = lst.ListCount

tb.MoveLast
grafik.NumPoints = tb.RecordCount
For j = 0 To lst.ListCount - 1
    grafik.ThisPoint = j + 1

    grafik.ThisSet = j + 1
    grafik.LegendText = lst.List(j)
Next j

i = 0
tb.MoveFirst
Do
```

```
i = i + 1
grafik.ThisPoint = i
grafik.LabelText = tb(txt(2))
For j = 0 To lst.ListCount - 1
    grafik.ThisSet = j + 1
    grafik.GraphData = tb(lst.List(j))
Next j
tb.MoveNext
Loop Until tb.EOF

grafik.GraphTitle = txt(3)
grafik.BottomTitle = txt(4)
grafik.LeftTitle = txt(5)

grafik.GraphType = cmbgrafftürü.ListIndex + 1
grafik.GraphStyle = cmbalttürü.ListIndex

i = 0
If ckh(0) Then i = 1
If ckh(1) Then i = i + 4
grafik.LineStats = i
grafik.Visible = True
grafik.ZOrder 0
grafik.Width = grafik.Width + 100
grafik.Width = grafik.Width - 100
End Sub

Private Sub Form_Load()
    cizik = False
    cmbgrafftürü.Clear
    cmbgrafftürü.AddItem "2B Pay", 0
    cmbgrafftürü.AddItem "3B Pay", 1
    cmbgrafftürü.AddItem "2B Çubuk", 2
    cmbgrafftürü.AddItem "3B Çubuk", 3
    cmbgrafftürü.AddItem "Gantt", 4
    cmbgrafftürü.AddItem "Çizgi", 5
    cmbgrafftürü.AddItem "Logaritmik", 6
    cmbgrafftürü.AddItem "Alan", 7
    cmbgrafftürü.AddItem "XY türü", 8
    cmbgrafftürü.AddItem "Kutupsal", 9
End Sub

Private Sub Form_Resize()
    If Me.WindowState <> 1 Then
        grafik.Move 0, 0, Me.Width, Me.Height - 385
    End If
End Sub

Private Sub Form_Unload(Cancel As Integer)
    tb.Close
    db.Close
End Sub

Private Sub grafik_DblClick()
    grafik.Visible = False
    If Me.WindowState <> 2 Then
        Me.Width = lin.X1
        Me.Height = lin.Y1 + 500
    End If
End Sub

Private Sub lblcontrol_Change()
    Static i As Integer
    If lblcontrol = "*" Then
        grafik.Visible = False
    End If
End Sub
```



```
Me.Width = lin.X1
Me.Height = lin.Y1 + 500

If Not cizik Then
    Set db = OpenDatabase(txt(0))
    Set tb = db.OpenTable(txt(1))
    For i = 0 To tb.Fields.Count - 1
        cmbseriler.AddItem tb.Fields(i).Name
    Next i
    cizik = True
End If
End If
End Sub

Private Sub lst_DblClick()
lst.RemoveItem lst.ListIndex
End Sub

Private Sub mnucikis_Click()
Unload Me
End Sub

Private Sub mprint_click()
PrintForm
End Sub

Private Sub txt_DblClick(Index As Integer)
If Index = 2 Then txt(2) = cmbseriler.List(cmbseriler.ListIndex)
End Sub
```

ÖZGEÇMİŞ

Naci Yananlı aslen Kilisli olup 1964 yılında İstanbulda doğmuştur. Fenerbahçe Lisesini bitirdikten sonra 1981 yılında girdiği İstanbul Üniversitesi İktisat Fakültesinden 1986 yılında mezun olmuştur. Aynı yıl İstanbul Üniversitesi Bilgi İşlem Merkezinde çalışmaya başlamıştır. 1991 yılında İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Mühendislik Bilimleri Ana Bilim dalı Sistem Analizi Programında yüksek lisans yapmaya hak kazanmıştır. 1990 yılından bu yana İstanbul Üniversitesi Sosyal Bilimler Meslek Yüksek Okulu Bilgisayarlı Muhasebe ve Vergi Denetimi Programında araştırma görevlisi olarak çalışmaktadır.