T.C. MILLI EĞITIM BAKANLIĞI



GÜÇLENDİRİLMESİ PROJESİ)

ENDÜSTRİYEL OTOMASYON TEKNOLOJİLERİ

BİLGİSAYARLI KONTROL - 1

ANKARA 2008

Milli Eğitim Bakanlığı tarafından geliştirilen modüller;

- Talim ve Terbiye Kurulu Başkanlığının 02.06.2006 tarih ve 269 sayılı Kararı ile onaylanan, Mesleki ve Teknik Eğitim Okul ve Kurumlarında kademeli olarak yaygınlaştırılan 42 alan ve 192 dala ait çerçeve öğretim programlarında amaçlanan mesleki yeterlikleri kazandırmaya yönelik geliştirilmiş öğretim materyalleridir (Ders Notlarıdır).
- Modüller, bireylere mesleki yeterlik kazandırmak ve bireysel öğrenmeye rehberlik etmek amacıyla öğrenme materyali olarak hazırlanmış, denenmek ve geliştirilmek üzere Mesleki ve Teknik Eğitim Okul ve Kurumlarında uygulanmaya başlanmıştır.
- Modüller teknolojik gelişmelere paralel olarak, amaçlanan yeterliği kazandırmak koşulu ile eğitim öğretim sırasında geliştirilebilir ve yapılması önerilen değişiklikler Bakanlıkta ilgili birime bildirilir.
- Örgün ve yaygın eğitim kurumları, işletmeler ve kendi kendine mesleki yeterlik kazanmak isteyen bireyler modüllere internet üzerinden ulaşılabilirler.
- Basılmış modüller, eğitim kurumlarında öğrencilere ücretsiz olarak dağıtılır.
- Modüller hiçbir şekilde ticari amaçla kullanılamaz ve ücret karşılığında satılamaz.

İÇİNDEKİLER

AÇIKLAMALAR	iii
GÍRİŞ	1
ÖĞRENME FAALİYETİ – 1	3
1. GÖRSEL PROGRAMLAMA NEDİR	3
1.1. Görsel Programlama Dilinin Kurulumu	5
1.1.1. Kurulum	5
1.1.2. Sürümler	5
1.1.3. Kurulum Adımları	5
1.2. Yardım Menüleri ve Online Yardım	13
1.2.1. Yardım Menüleri	13
1.2.2. Yardım Menülerinin Kullanımı	14
1.2.3. Online Yardım	15
1.2.4. Programın Başlatılması	17
UYGULAMA FAALİYETİ	. 19
ÖLÇME VE DEĞERLENDİRME	20
ÖĞRENME FAALİYETİ–2	22
2. ARAYÜZ OLUŞTURMAK	22
2.1. Görsel Programlama Ortamı	23
2.1.2. Başlık Çubuğu	24
2.1.3. Menü Çubuğu	24
2.1.4. Araç Çubuğu	25
2.1.5. Araç Kutusu	27
2.1.6. Form Taşıyıcı Penceresi	27
2.1.7. Form Yüzeyi	28
2.1.8. Kod Penceresi	28
2.1.9. Proje Penceresi	29
2.1.10. Özellikler Penceresi	30
2.1.11. Form Yerleşim Penceresi	31
2.2. Görsel Programlama Mantığı	31
2.2.1. Visual Basic'in Avantajlari	31
2.2.2. Özellik (Property), Metot ve Olay Kavramları	32
2.3. Temel Görsel Programlama Bileşenleri	41
2.3.1. Command Button (Komut Düğmesi)	41
2.3.1. Komut Düğmesi Özellikleri	42
2.3.2. Textbox (Metin Kutusu)	44
2.3.3. Label (Etiket)	47
UYGULAMA FAALİYETİ	49
ÖLÇME VE DEĞERLENDİRME	. 50
ÖĞRENME FAALİYETİ – 3	52
3. PROJELER VE BİLEŞENLERİ	52
3.1. Proje Türleri	. 53
3.2. Proje Ekranı	55
3.3.11k Visual Basic Projemiz	. 56
3.3.1. Proje Konusunun Belirlenmesi	56

3.3.2. Kullanilacak Nesnelerin Tespiti	6
3.3.3. Form Yüzeyine Kontrollerin Yerleştirilmesi	6
3.3.4. Kontrollerin Olaya Yönelik Kodlarının Yazımı	9
3.3.5. Programın Geliştirilmesi	2
UYGULAMA FAALİYETİ	4
ÖLÇME VE DEĞERLENDİRME6	5
ÖĞRENME FAALİYETİ – 4	7
4. VISUAL BASIC HATA DENETLEME ARAÇLARI	7
4.1. Program Modlari	8
4.1.2. Immediate Window (Anlık Durum Penceresi)	8
4.1.3. Watch Window (İzleme Penceresi)	9
4.1.4. Locals Window (Değişken Penceresi)	0
4.1.5. Programı Adımlamak7	1
4.2. Projelerin Derlenmesi ve Çalıştırılabilir Dosyaya Dönüştürülmesi	8
UYGULAMA FAALİYETİ	2
ÖLÇME VE DEĞERLENDİRME8	3
MODÜL DEĞERLENDİRME	5
CEVAP ANAHTARLARI	6
KAYNAKÇA	7

AÇIKLAMALAR

KOD	523EO0312		
ALAN	Endüstriyel Otomasyon Teknolojileri		
DAL/MESLEK	Alan Ortak		
MODÜLÜN ADI	Bilgisayarlı Kontrol - 1		
MODÜLÜN TANIMI	Görsel programlama dilinin kurulum ve kullanım yeterliliklerinin kazandırıldığı öğretim materyalidir.		
SÜRE	40/32		
ÖN KOŞUL			
YETERLİK	Görsel programlama ortamını kullanmak.		
MODÜLÜN AMACI	 Genel Amaç Görsel programlamada çalıştırılabilir dosya işlemini doğru olarak yapabileceksiniz. Amaçlar 1. Görsel programlama ortamını hatasız olarak kurabileceksiniz. 2. Görsel programlama ortamını doğru olarak kullanabileceksiniz. 3. Görsel programlamada kurallara uygun olarak proje oluşturabileceksiniz. 4. Görsel programlamada kurallara uygun olarak proje 		
EĞİTİM ÖĞRETİM	Ortam: Bilgisayar Laboratuvarı		
ORTAMLARI VE DONANIMLARI	Donanım: Bilgisayar		
ÖLÇME VE DEĞERLENDİRME	Her faaliyetin sonunda ölçme soruları ile öğrenme düzeyinizi ölçeceksiniz. Araştırmalarla, grup çalışmaları ve bireysel çalışmalarla öğretmen rehberliğinde ölçme ve değerlendirmevi gerceklestirebileceksiniz.		

iv

GİRİŞ

Sevgili Öğrenci,

Bu modülden itibaren görsel programlama dilini kurmaya ve kullanmaya başlayacaksınız. Görsel programlama dilini şimdiye kadar öğrenmiş olduğunuz programlama dilleriyle karşılaştırdığınızda, programlamaya getirdiği yenilikleri hemen farkedeceksiniz. Bunun yanısıra, editör tabanlı programlama dillerine oranla kolaylıkları mutlaka dikkatinizi çekecektir. Umuyorum ki görsel programlamayı sever ve bu modülü bitirdikten sonra, boş zamanlarınızda bile programlama çalışır hale gelirsiniz.

Görsel Programlama Ortamında Çalışmak modülü 4 öğrenme faaliyetinden oluşmaktadır.

- Görsel Programlama Dilinin Kurulumu
- Görsel Programlama Dilinde Arayüz Oluşturma
- Görsel Programlama Ortamında Proje Oluşturma
- Görsel Programlama Ortamında Hata Denetimi ve Derleme

Öğrenme faaliyetlerinde konu genel olarak anlatılmış ve örneklerle desteklenmiştir. Bu yöntem, yapacağınız uygulamalara rehberlik edecektir. Örneklerde, oluşturulacak ekran görüntüsü ve program kodları ve gerekli açıklamalar verilmiştir. Bu yol ile uygulamalarda yararlanacağınız işlem basamaklarını daha iyi anlayacaksınız.

Görsel programlama dilini öğrenme aşamasında bazı zorluklarla karşılaşacaksınız. Bu zorluklar karşısında sorunu teşhis edip çözümü kendiniz düşünmelisiniz. İlk anlarda yaşayacağınız sorunlar belirli bir aşama sonunda kolay bir hale gelecektir.

ÖĞRENME FAALİYETİ–1

AMAÇ

Görsel programlama ortamını hatasız olarak kurabileceksiniz.

ARAŞTIRMA

Bu öğrenme faaliyetinden önce aşağıdaki hazırlıkları yapmalısınız.

- Görsel Programlama çalışmaları yapmak amacıyla kullanılan programlama dilleri nelerdir? Bunlar hakkında araştırma yapınız.
- Görsel programlama dillerinin kullanım alanları hakkında araştırma yapınız.
- İnterneti kullanarak görsel programlama üzerinde çalışmaların, örnek ve konu anlatımlarının yer aldığı web sitelerini araştırınız.

1. GÖRSEL PROGRAMLAMA NEDİR

Görsel programlama için basit bir tanım yapmak gerekirse "göz göre göre program yapmak" diyebiliriz. Bu programlama anlayışı; önceki yıllarda kullanılan temel programlama dillerindeki bir editör ekran üzerine satır satır yazılan programlamanın yerine, Grafiksel Kullanıcı Arabirimi (GUI)'nin ve bu arabirim içerisinde program geliştirme amacıyla bulunan Entegre Geliştirme Ortamı (IDE) adlı bir araçlar koleksiyonunun almasıyla ortaya çıkmıştır.

Visual BASIC'e gelince, öncelikle BASIC programlama dili temel alınarak ortaya çıkarılmış olduğunu söylemek gerekir. BASIC (Beginners-All Purpose Symbolic Instruction Code) kelime anlamı olarak "Yeni Başlayanlar İçin Çok Amaçlı Sembolik Talimat Kodu" anlamına gelmektedir. BASIC metin tabanlı editörler arasında en çok rağbet göreni olmuştur. Başlangıçta çok basit programların geliştirilmesi amacıyla kullanılırken, bu ilgi dolayısıyla gitgide gelişerek son halini almıştır.

Bu kadar bahsettikten sonra BASIC programlama dili ile yazılmış bir program örneğini inceleyelim.

```
10 REM Bu program ilk on rakamın karelerini hesaplar ve
ekrana yazdırır.
20 CLS
30 PRINT "Squares from 1 to 10"
40 PRINT "Value", "Squared"
50 FOR N = 1 TO 10
60 PRINT N, (N*N)
70 NEXT N
80 PRINT
90 END
```

Şekil 1.1: Örnek BASIC programı kodları

Yukarıda da gördüğünüz gibi BASIC dili her satırın başına bir numara vermeyi gerekli kılmaktaydı. Programcılar da satır numaralarını yine yukarıda gördüğünüz gibi onar onar vermekteydiler. Böylece eklemek istedikleri bir satırı arada "örneğin 55 gibi" bir satır numarası vererek ekleyebilmekteydiler ve program, bilgisayar tarafından satır satır işletilirdi.

Görsel programlama ile ise farklı bir programlama anlayışı gelişti ve programlama ekran üzerine yerleştirilen nesnelere ve onların hareket ve durumlarına göre yazılmaya başlandı. Bu programlama tarzına nesneye dayalı, nesne eğilimli ya da olay yönelimli programlama gibi adlar verildi.

Visual BASIC ile oluşturulmuş bir programın ekran görüntüsünü inceleyecek olursak;

💐 HESAPLAMA	<u>_ </u>	
1.SAYI:		TOPLAMA
2.SAYI:		CIKARTMA
SONUC		CARPMA
		BOLME
TEMIZLE		

Şekil 1.2: Örnek Visual BASIC programı ekran görüntüsü

Gördüğünüz gibi Visual BASIC programı ile oluşturulan programların ekran görüntüleri daha önceden tanımakta olduğunuz pek çok Windows tabanlı programın ekran görüntüleriyle benzerlik göstermektedir.

Buradan yola çıkarak Microsoft'un tüm ofis uygulamalarının ve Windows bileşenleri arasında yer alan hesap makinesi, MsPaint, vb. pek çok programın Visual BASIC ortamında geliştirildiğini söylemeliyiz.

1.1. Görsel Programlama Dilinin Kurulumu

Visual BASIC programını kurmak, daha önceden edinmiş olduğunuz program kurulum tecrübeleri sayesinde sizler için oldukça basit bir uygulama olacaktır. Şimdi isterseniz, bu uygulamanın adımlarını birlikte inceleyelim.

1.1.1. Kurulum

1.1.2. Sürümler

Kurulumunu gerçekleştirecek olduğumuz Visual BASIC 6.0 programı Microsoft Firması tarafından üretilmekte olup Microsoft Visual Studio 6.0 Kurulum CD'si ile birlikte edinilebilir. Üç farklı sürümü bulunmaktadır. Bunlar;

- Learning Edition: Görsel programlama dili kullanarak program geliştirmek için gerekli temel bileşenlere sahiptir. Genelde eğitim amaçlı kullanılmakla birlikte, görsel programlama alanında yeni olan kullanıcılar için tavsiye edilmektedir.
- Professional Edition: Uzman kullanıcılara yönelik olmakla birlikte, internet tabanlı programlar geliştirmek için gerekli olan bileşenleri içerir.
- Enterprise Edition: İstemci-sunucu tabanlı, yüksek performanslı uygulamalar geliştirmek için ve yine internet tabanlı uygulamalar geliştirmek için gerekli bileşenleri içerir.

1.1.3. Kurulum Adımları

Microsoft Visual Studio 6.0 Kurulum CD'sini yerleştiriniz ve aşağıdaki uygulama adımlarını takip ediniz;

Kurulum Sihirbazının devreye girmesi

CD'yi taktıktan bir süre sonra "Visual Studio Installation Wizard" yani "Kurulum Sihirbazı" devreye girer. Ve bu sihirbaz sizi tüm kurulum boyunca yönlendirir.



Şekil 1.3: Kurulum sihirbazı devreye girer.

Kurulum Sihirbazı ile gelen ilk ekran görüntüsü üzerinden Visual Studio hakkında içerik ve lisanslandırma hakkındaki bilgileri içeren Readme dosyasını "View Readme" butonuna basarak ulaşabilirsiniz.

Would Studie Readine - Microsoft Internet Englorer - (Working Offline)	-1013
ne Dit geer Tearrier (off 1960)	
STON STONERSHIM	n 🚺 😂 🖬 tanka ?
Visual Studio Readme	-
© 1008 Microsoft Corporation. All rights reserved.	
Other product and company names herein may be the trademarks of their respective owners.	
Violal Studio Readine contains vital information about Microsoftigh Visual Studio ¹⁴⁶ 6.6 — Development System for Windowegi and the Internet. This number of the products stipped in the pate. This Readine updates the information provided in the ASDV ubrary Visual Studio 6.0 documentation.	information may apply to all or a
Voide/Studio Readore is one of a "family" of Readmes shipped with the suite. All product Readme files reside at the rout of Visual Studio compact of plus several athers called readmextChlmi, where XX signifies a product) use current Web technology so users can find and view their information gus towards, such as Morsoloft pinteme Explorer 4, or in Notepool.	sc 1. These Readine files (install.htm, uickly. They may be viewed with a Web
How do I print all the exellable Readme files at one time from Microsoft Internet Explorer?	-
Contents	
Note Be sure all headings in the table of contents are expanded when you search this Readme for a topic. In this way, you will know when omeng the TOC headings.	the search finds the topic
General Visual Studio Issues	
Installation Notes (install htm) covers general installation issues, including side by side product installations, affecting the Visual Studio 6.0 si	uite of products.
Microsoft SQL Server Must Run Under a Valid User Account and Rot "System Account" for SQL Debegging	
Registering Visual Studio 6.0 Products on the Web	
Viewing Visual Studio Windows In Windows 98	
Visual Studio Help Viewer Issues	
Sample Code Sometimes Does Not Cut and Paste Property	
Users May Update the Version of HTML Help SDK Shipped with Visual Studio	
Einding Information on HTML Tags	
Einding Help for #00 Objects	
For other issues on the Help system for the Visual Studio suite of products, go to:	
1450N"', the Microsoft Developer Network Readme,	
If you acquired any of the following products in the Visual Studio suite, and need specific information, click the product name.	
Daw	Ny Campular

Şekil 1.4: Readme sayfası görüntülenir.

Bu dosyada Visual Studio 6.0 Enterprise Edition ile birlikte gelen tüm programların Readme dosyalarına da ulaşabilirsiniz.

Bu programlar aşağıda sıralanmıştır:

- ► MSDNTM, the Microsoft Developer Network
- → Microsoft[®] Visual Studio[™], Enterprise Edition
- Microsoft® Visual Basic®
- ➢ Microsoft[®] Visual C++[®]
- Microsoft® Visual Database ToolsTM
- ➢ Microsoft[®] Visual FoxPro[™]
- ➢ Microsoft[®] Visual InterDev[™]
- ➢ Microsoft[®] Visual J++™
- Microsoft® Visual SourceSafeTM

Readme Dosyasına bir göz atıp kapattıktan sonra, kuruluma devam etmek için "NEXT" butonuna basınız.

Bir sonraki ekranda karşımıza son kullanıcı lisans anlaşması çıkar.



Şekil 1.5: Son kullanıcı lisans anlaşması görüntülenir.

Bu ekran üzerinde bulunan "*I accept the agreement*" yani "anlaşmayı kabul ediyorum" butonunu tıklar ve ardından aktif olan "Next" butonuna basarız.

Daha sonra karşımıza "Product Number and User ID" yani "program numarası ve kullanıcı kimliği" yazan ekran görüntüsü gelecektir.

Bu ekran üzerinde ilgili alanlara ürünümüzün kutusu ve CD'si üzerinde yazılı olan Product ID' yi gireriz.

Ardından kendi adımızı ve eğer varsa firma adı bilgilerini gireriz.



Şekil 1.6. Ürün numarası ve kullanıcı kimliği ekranı

Kuruluma devam etmek için "*Next*" butonuna basarız. Bir sonraki ekranda karşımıza "*Enterprise Setup Options*" yani "*kurulum seçenekleri*" karşımıza çıkmaktadır. Bu seçeneklerden birincisini, "*Custom*"ı seçerek, tüm kurulumu kendimiz yönlendirebiliriz.



Şekil 1.7. Ürün numarası ve kullanıcı kimliği ekranı

"Custom" seçeneğini işaretleyip "Next" butonuna basarız.

Ardından karşımıza kurulumun yapılacağı klasörün adresini gireceğimiz bir ekran gelmektedir. "*Browse*" butonuna basarak disk alanımızda farklı bir alanı seçebileceğimiz gibi, varsayılan olarak verilmiş olan;

"C:\Program Files\Microsoft Visual Studio\Common" yolunu da seçebiliriz..



Şekil 1.8: Kurulum klasörü belirleme ekranı

Klasör adresini değiştirmeden bırakıp, "Next" butonuna basarak kuruluma devam ederiz.



Şekil 1.9. Kurulum başlıyor

Yukarıda görünen ekran görüntüsüyle kurulumumuz başlamış olur. Hemen ardından;

	Welcome to the Visual Stu	udio 6.0 Enterprise ir	stallation program.	
Setup car any open	not install system files or upo applications.	date shared files if th	e files are in use. Before	continuing, close
VARNING:	This program is protected by	y copyright law and i	nternational treaties.	
/ou may ins	all Visual Studio 6.0 Enterpri	ise on a single comp	uter. Some Microsoft pro	ducts are provide
with additior	al rights, which are stated in	I the End User Licen	se Agreement included v	vith your software
Please take	a moment to read the End U	Jser License Agreem	ent now. It contains all o	f the terms and
conditions these terms	at pertain to this software pr	oduct. By choosing	to continue, you indicate	acceptance of
Please take	a moment to read the End U	Jser License Agreem	ient now. It contains all o	f the terms a
conditions these terms	at pertain to this software pr	roduct. By choosing	to continue, you indicate	acceptance

Şekil 1.10: Kurulum başlatma ekranı

Yukarıda görülen kurulum başlatma ekranıyla son kullanıcı lisans anlaşmasının okunması kullanıcıya tekrar hatırlatılır ve "Continue" butonuna basarak kuruluma devam edilir.

Product ID:	50021-335-7140011-62423
This is your Microsoft p Microsoft for technical	product identification number. If you want to call support, you will be asked for this number.
For your records, pleas your registration card. / the number by clicking	e write down this number in the designated portion of After the software has been installed, you can access the About command on the Help menu.

Şekil 1.11: Program ID kontrol ekranı

Karşımıza gelen bu ekranla satın almış olduğumuz ürünün program ID numarası kontrol edilir. "OK" butonu tıklanarak kurulum sürdürülür.

want installed; clea	r the items you do not want installed.	
nat only part of the o	omponent will be installed. To select all componen	
	Description:	
46350 K	Installs Microsoft Visual Basic and all required components, including core ActiveX controls.	
3446 K	Change Option	
10047 K	10047 K	
49283 K 🛁	<u>S</u> elect All	
6041 K 💌	P	
udio/VB98	Change <u>F</u> older	
486190 K		
	a want installed; clea hat only part of the c 496650 K ▲ 251351 K 67642 K 43754 K 9926 K 3446 K 10947 K 49293 K 49201 K ▼ udio\/VB98 486190 K	

Şekil 1.12: Program kurulum seçenekleri belirleme ekranı

Visual Studio kurulum CD'si içerisinde yer alan tüm programların, kurulum aşamasında seçim işlemi bu şekilde seçilir.

Visual Basic 6.0'ın kullanılabilmesi için gerekli olmayan ya da bizim ihtiyaç duymayacağımız program seçeneklerinin işaretini kaldırıp kurulumu başlatabiliriz.

/isual Studio 6.0 Enterprise - Custom				
In the Options list, select the items you (want installed; clea	ar the items you do not want installed.		
A grayed box with a check indicates the in the Option list, click Select All.	at only part of the	component will be installed. To select all components		
Options:		Description:		
Microsoft Visual Basic 6.0 Microsoft Visual C++ 6.0 Microsoft Visual FoxPro 6.0 Microsoft Visual InterDev 6.0 Microsoft Visual CourseS 46 5.0	46350 K 251351 K 67642 K 43754 K	Setup will allow you to customize which components to install		
ActiveX	3446 K	Change Option		
Data Access Enterprise Tools Graphics	10047 K 49283 K 6041 K 💌	Select All		
Folder for Currently Selected Option: -				
C:\Program Files\Microsoft Visual Stur	dio\VIntDev98	Change <u>F</u> older		
Space required on C:	173238 K			
Space available on C:	2779288 K			
<u>C</u> onti	nue	Cancel		

Şekil 1.13: Program kurulum seçenekleri belirlenir

"Continue" butonuna basılarak kuruluma devam edilir.

Bir sonraki ekran bir uyarı ekranı olacaktır. Bu ekranda veritabanı uygulamaları için gerekli olabilecek bir güncelleme önerilmektedir. "YES" butonuna basıp bu güncellemenin yüklenmesine onay verebiliriz.



Şekil 1.14: VSS DB veri tabanı uygulama güncellemesi

Güncellemenin yüklenmesinin ardından diğer Visual Studio dosyalarının kopyalanması başlayacaktır.



Şekil 1.15: Visual Studio kurulum dosyaları kopyalanıyor.

Kopyalama işlemi sırasında hangi dosyaların, hangi klasörlere yazıldığı ve kalan süre akan işlem göstergesinden takip edilebilir.

Visual Studio	6.0 Enterprise Setup	×		
Destination Fil C:\Microsoft	e: Visual Studio\VIntDev98\bin\VS	ED.DLL		
25%				
Cancel				

Şekil 1.16: İşlem göstergesi

Kurulum işlemi tamamlandığında bilgisayarınızı yeniden başlatmanızı ya da kurulum işleminden çıkmanızı öneren bir uyarı ekranıyla karşılaşırsınız.

Pek çok programda olduğu gibi kurulumun ardından bu uyarının alınmasıyla birlikte, yapılacak en doğru davranış *"Restart Windows"* butonunun tıklanması ve işletim sistemimizin yeniden başlatılmasının sağlanması olacaktır.

2	You need to restart Windows to con	nplete the installation.
	Before you restart, close any open a and save all current documents. Re disk drive. Then, click the Restart V	pplications, including MS-DDS sessions, move any floppy disks still inserted in a /indows button.
	If you choose to restart, the screen i while Setup restarts your machine.	nay go blank or flicker. Please be patient
	<u>R</u> estart Windows	Egit Setup

Şekil 1.17: Kurulum tamamlandı

Ancak henüz kurulum tamamlanmış sayılmaz!

Bilgisayarınız yeniden başlar başlamaz, Visual Studio 6.0 Enterprise Edition ile birlikte gelen diğer program eklentilerini de yüklemek isteyip istemediğinizi soracaktır. Bizim yapacağımız temel seviye uygulamalar için bu eklentilere gerek olmadığı için yüklememize gerek olmayacaktır.

Tüm çıkan eklenti kurulum ekranlarını ardı ardına kapattıktan sonra kurulumunu tamamlamış olduğunuz programınızın internet üzerinden *"Register"* yani üye edilmesi, kayıtlı hale getirilmesi için hazırlanan ekran karşınıza gelir.

Bu ekrandaki üye olma işlemi olmadan da, Visual Studio Programı ve elbette Visual BASIC Programı çalışacak olmakla beraber; üye olmak size internet üzerinden yeni sürümler hakkında bilgi alma, yeni çıkan güvenlik paketleri ya da iyileştirme paketleri,..vb. hakkında bilgi edinme gibi avantajlar sağlayacaktır.



Şekil 1.18.Üyelik ekranı.

1.2. Yardım Menüleri ve Online Yardım

1.2.1. Yardım Menüleri

Visual BASIC programlama diline ait tüm komutları ve bunların kullanım kurallarını bilmeniz oldukça güçtür. Bu nedenle programlarınızı hazırlarken birçok defa yardım almak amacıyla bir kitaba ya da internete ihtiyaç duyarız. İnternette programlama üzerine hazırlanmış yardım alabileceğimiz pek çok site mevcuttur. Bunun yanı sıra Visual BASIC programlama dili kendi içerisinde de bir yardım menüsü barındırmaktadır. F1 tuşuna basılarak yardım menüsü görüntülenebilir.

Visual BASIC' de yardım alabilmemiz için, bunların kütüphanelerini yani yardım içeriğini barındıran MSDN (Microsoft Developer Network)'yi kurmamız gereklidir. MSDN'i Visual BASIC kurulumu tamamlandıktan sonra karşınıza gelecek olan aşağıdaki uyarı ekranını takip ederek kurabilirsiniz. Ancak Visual BASIC kurulum CD'lerinin tamamına ya da MSDN kurulum dosyalarına ihtiyacınız olacaktır.



Şekil 1.19: MSDN kurulum ekranı

MSDN kurulumu tamamlandıktan sonra F1 tuşuna basılarak yardım menüsü açılabilir.

Eğer MSDN 'i kurmadan F1 tuşuna basarsanız aşağıdaki uyarı ekranıyla karşılaşırsınız.



Şekil 1.20: MSDN kurulu olmadan yardım almak istenildiğinde karşılaşılacak uyarı ekranı

1.2.2. Yardım Menülerinin Kullanımı

F1 tuşuna basıldığında karşılaşılacak olan yardım ekranı aşağıdaki şekilde gösterilmektedir.



Şekil 1.21: Yardım ekranı

Gördüğünüz gibi yardım ekranı, internet sayfasına çok benzeyen bir görüntüye sahiptir. "Contents" yazan bölümden konu başlıklarına göre istediğiniz başlığı seçerek, "Index" bölümünden tüm başlık ve alt başlıkları alfabetik sıraya uygun şekilde inceleyerek ya da aradığınız konunun ilk harflerini girerek, "*Search*" bölümünden ise aradığınız konu içeriğini yazıp o konu ile ilgili tüm başlık ve alt başlıklara ulaşarak arama yaptırabilirsiniz.

Aranıp bulunan konunun üzeri tıklanarak aynen bir web sayfasında olduğu gibi yan tarafta görüntülenmesi sağlanabilir.

"Favorites" kısmı ise arama için değil daha çok eski yapmış olduğumuz aramalara kolay ulaşmak için kullanılır. Örneği araştırıp bulduğunuz bir konu ile ilgili yardım ekranı açık iken, Favorites kısmına gider ve *"Add"* butonuna basarsanız, orada yer alan listeye o sayfanın eklendiğini göreceksiniz.

Bir konu ile ilgili yardım sayfası görüntülenirken, bir sayfaya sığmayabilir bu durumda araç çubuğunda yer alan *"Forward"* butonu yardımıyla ilerlenebilir. Konu ile ilintili olabileceğini düşündüğünüz bir alt başlığa gitmek için *"Next"*, bir üst başlığa gitmek için ise *"Previous"* butonu kullanılabilir.

Arayıp bulmuş olduğunuz konu ile ilgili açılan sayfanın hemen üstünde "See Also", "Example", "*Specifics*", "*Applies To*" ve "*Tasks*" gibi linklerin birini ya da birkaçını görebilirsiniz. Bunlar eğer var ise bu konuyla ilintili başka konuları, örnekleri, özellikleri, uygulamaları görmenizi sağlar(bk. Şekil 1.21.). Ayrıca sayfanın içerisinde, konu anlatımı sırasında altı çizili ve mavi renkte göreceğiniz kelimelerinde, o kelimelerle ilgili yardım sayfalarına sizi yönlendiren birer link olduğunu belirtmemiz gerekir(bk. Şekil 1.21.).

Contents Index Search Favorites	Contents Index Search Favorites Type in the keyword to find Axis TickStyle constants Axis TickStyle constants Axis TileActivated event Axis TileActivated event Axis TileJpdated event Axis Til	Contents Index Search Favorites Type in the word(s) to search for: command button List Topics Display Select topic: Found 500 Title Location Rai Using the Comman. Visual Bas. 40 Using form as Cour., Visual Bas. 100 Manipulating Your., Visual Bas. 100	Contents Index Search Favorites Togics: About Active Server Pages command button Form Window Working with Text Boxes
SDK Documentation DK Documentation DK Documentation Windows Resource Kits Specifications Knowledge Base Sechical Articles Backgroundes Backgroundes Books	B channel -B NMAKE option ba "ba (treak-point Address) BA samples (Broadcast Architecture) Babble Applet back P ant Display	Garoping Options w., Visual Bas., 142 Grouping Options w., Visual Bas., 147 Modilying the Form., Visual Bas., 110 Create a Master/D., Visual Bas., 110 Create a Master/D., Visual Bas., 126 Search previous results Match millar words Search titles only	Eemove Display Current topic: Using the Command Button Control Add

Şekil 1.22: Yardım menüleri

1.2.3. Online Yardım

Visual BASIC'de MSDN kütüphanesinin yanı sıra, bu kütüphane içerisinde yer almayan ihtiyaçlarınızı da karşılamak amacıyla online yardım desteği de sunulmaktadır. Bu yardım menülerine "Help" menüsünün içerisinde yer alan "Microsoft on the Web" menüsünden ulaşabilirsiniz.



Şekil 1.23: Online yardım menüleri

Bu menüyü kullanarak internet üzerinde yer alan Visual BASIC kullanımı ile ilgili sıkça sorulan sorulara "*Frequently Asked Questions*", yeni sürüm ve eklentilere, online desteğe, makalelere ve buna benzer pek çok ilintili alana ulaşabilirsiniz.



Şekil 1.24: Türkiye MSDN Visual BASIC 6.0 anasayfası

🙆 : 101 Samples for Visual Basi	c 2005 - Microsoft Internet Explorer	X
Dosya Düzen Görünüm Sikk	(ullanilanilar Araçlar Yardım	1
🌀 Geri 🔹 🕥 🖌 😦 💋	😚 🔎 Ara 🤺 Sik Kullanlarilar 🤣 🔗 - 🤪 🔯 - 🛄 🐲 🐇	Bağlantılar 🂝
Adres 🔕 http://msdn.microsoft.co	m/VBasic/Downloads/Code/101Samples/	💌 🔁 GR
Google -	💽 💽 Search 🔹 🥩 🔊 O blocked 🏘 Check 🔹 🌂 AutoLink 🔹 💬 AutoEll 💽 Options 🥒	
	Microsoft.com Hom	e Site Map 🔺
msdn		
MSDN Home	Developer Centers Library Downloads How to Buy Subscribers Worldwide	
Search for	Microsoft* Visual Basic* Developer Center The most productive tool for building .NET-connected applications	Visual Studio.net
Advanced Search Visual Basic Home	MSDN Home > Visual Basic Home > Downloads > Code Samples > 101 Samples	i
Reference	101 Samples for Visual Basic 2005	
Downloads •	101 Samples demonstrate many of the new features available with Visual Basic 2005 a	and the .NET
Support •	Framework 2.0. For more samples using Visual Basic .NET 2003, see this download.	
Community •		
Product Information +	Download all 101 Samples.	
Previous Versions		
Future Versions	Base Class Libraries - Group 1 Samples Samples Supples Samples Supples Su	
BUILD	Samples	
ARCADE	Web Development Samples Data Access Samples	
	NET Compact Framework Samples NET Compact Framework Samples	
	Base Class Libraries - Group 1 Samples	
http://msdn.microsoft.com/VBasi	c/Downloads/Code/101Samples/#bd	S Internet
🛃 Başlat 🧶 🚱 💽 🔌 🛃	: 101 Samples for Visu 📕 Bell.doc - Microsoft Word 🏠 Microsoft Visual Basic	🌃 « 🛃 🗛 17:43

Şekil 1.25:Online yardım ile hazır örnekleri inceleyebilirsiniz

1.2.4. Programın Başlatılması

Visual Basic programının başlatılması için, başlat çubuğunu kullanarak tüm programlar sekmesinden Microsoft Visual Studio 6.0 butonuna ulaşmanız ve açılan menüden Visual Basic 6.0'ı seçmeniz gerekmektedir.



Şekil 1.26:Visual Basic'in başlatılması

Program açıldıktan sonra, size ilk olarak yeni proje ekranı karşınıza gelecektir. Bu ekran üç sayfadan oluşmaktadır. İlk sayfada oluşturacak olduğunuz projenin türünü belirleyebilirsiniz. İkinci sayfa olan "Existing" bölümüne girerek daha önceden oluşturmuş olduğunuz projelerinizi araştırıp açabilirsiniz. Recent sayfasından ise son olarak üzerinde çalışmalar yapmış olduğunuz projelerinize ulaşabilirsiniz. Şimdi tekrar birinci sayfamız olan New bölümüne geri dönüp, proje türleri hakkında bilgi verelim. Proje türü denildiğinde, Visual Basic ile farklı bileşenler kullanılarak projeler üretebileceğiniz anlatılmak istenmektedir. Yani kullanılan bileşenlerin türüne göre proje türleri de farklılaşmaktadır. Örneğin projenizin içerisinde ActiveX bileşenleri kullanabilir ya da projenizi bir DHTML sayfası yani web sayfası şeklinde hazırlayabilirsiniz. Biz ise bu ekrandan "Standard EXE" seçeneğini işaretleyip ilk çalışmamızı başlatabiliriz.

/	Micros		asi	e 😔	
ew Existing	Recent				
2	27	3-7	畫一		^
Standard EXE	ActiveX EXE	ActiveX DLL	ActiveX Control	VB Application Wizard	
2	5	5	5	5	
VB Wizard Manager	Data Project	IIS Application	Addin	ActiveX Document DI	
Pm 6	Pa 💊	P= 6			*
				¢	
				İptal	
				Yardım	
					_

Şekil 1.27:Yeni proje ekranı 17



İlk çalışmamızı Standard EXE olarak seçtiğimizde karşılaşacağımız Visual Basic çalışma alanımız ise aşağıdaki şekilde gösterilmektedir.

Şekil 1.28: Visual Basic çalışma alanı

UYGULAMA FAALİYETİ

Aşağıdaki işlem basamaklarına göre uygulama faaliyetini yapınız.

İşlem Basamakları	Öneriler
Visual BASIC ya da Visual Studio CD'lerini kurulum için hazırlayınız.	 Kullanacak olduğunuz kurulum CD'lerinin versiyonunu inceleyiniz.
Bilgisayarınızda daha önceden Visual BASIC programının kurulu olup olmadığını kontrol ediniz.	Denetim Masası'ndan Program Ekle/Kaldır kısmını kullanarak yapabilirsiniz.
Kurulumu başlatınız.	Kurulum CD'nizde yer alan ürün anahtarını kontrol edin.
Custom Seçeneğini kullanarak ihtiyaç duyduğunuz program bölümlerini kurunuz.	Kurulum adımlarını yukarıda anlatıldığı şekilde sırayla takip edin.
MSDN kurulumunu gerçekleştiriniz.	MSDN kurulum dosyaları Visual Studio Kurulum CD'lerinizin içinde ya da MSDN kurulum CD'si şeklinde olabilir. Bu dosyalara sahip olup olmadığınızı kontrol ediniz.
Yardım ekranını açınız.	F1 tuşunu ya da Help Menüsünü kullanabilirsiniz.
"Installing Visual Basic" yazarak Visual Basic kurulumu ile ilgili notlara ulaşınız ve kurulumdan önce dikkat edilmesi gereken özelliklerin içerisinden, gerekli olan donanım özellikleri nelerdir, bulunuz.	Yardım ekranından Index kısmını kullanabilirsiniz.
Online yardım kısmından hareket ederek Microsoft Visual Basic sayfalarından Learn Visual Basic linkini bulup, karşınıza çıkan ekranları inceleyiniz.	 İnternet bağlantınızı kontrol ediniz. Yukarıda verilen Online yardım ile ilgili ekran görüntülerini inceleyiniz.

ÖLÇME VE DEĞERLENDİRME

OBJEKTIF TEST (ÖLÇME SORULARI)

Aşağıdaki soruları cevaplayarak bu faaliyette kazandığınız bilgileri ölçünüz.

- 1. Visual Basic ile ilgili aşağıda yapılan tanımlamalardan hangisi yanlıştır?
 - A) Nesne eğilimlidir. C) Olay yönelimlidir. D) Nesne tabanlıdır.
 - B) Nesneye dayalıdır.

A) Enterprise Edition

B) Learning Edition

- 2. Aşağıdakilerden hangisi bir Visual Studio 6.0 sürümü değildir?
 - C) Development Edition
 - D) Professional Edition
- 3. Aşağıdakilerden hangisi Visual Studio Enterprise Edition ile birlikte gelen programlardan değildir?
 - A) Microsoft® Visual C++®
 - B) Microsoft® Visual **FoxProTM**

C) Microsoft® Visual JavaTM

D) Microsoft® Visual Basic®

4. Visual Basic kurulumunun ardından, internetten yazılımın kayıtlı hale getirilmesi için yapılan işlem esnasında açılan pencerenin adı nedir?

A) Register Penceresi

- C) AddIn Penceresi D) Install Penceresi
- B) CheckIn Penceresi
- 5. Visual Basic'de yardım almak için gerekli olan kısayol tuşu nedir?
 - A) F1 C) F5 B) F12 D) F4
- 6. Visual Basic yardım kütüphanesinin adı aşağıdakilerden hangisidir?
 - A) MVBH C) MSDN
 - D) VBLib B) VBHF
- 7. Yardım penceresinde aşağıdaki alt başlıklardan hangisi yer almaz?
 - C) Search A) Contents B) Index
 - D) My Favourites
- 8. Yardım pencerelerinden ilk harfe göre aratma işlemi yapan hangisidir?
 - A) Index C) Search
 - B) Contents D) Favourites

- 9. Yardım pencerelerinden başlık ve alt başlıklara göre aratma işlemi yapan hangisidir?
 - A) Index

A) Index

A) Index

- C) Search
- B) Contents D) Favourites

10. Yardım pencerelerinden içeriğe göre aratma işlemi yapan hangisidir?

- C) Search
- B) Contents D) Favourites

11. Yardım pencerelerinden hangisinde "Add" butonu yer almaktadır?

- C) Search
- B) Contents D) Favourites
- 12. Yardım sayfası içeriğinde yer alan altı çizili kelimelerin anlamı nedir?
 - A) Bu kelime çok önemlidir.
 - B) Bu kelime gereksizdir.
 - C) Bu kelime ile ilgili açıklayıcı bilgi yer almaktadır.
 - D) Bu kelime başka bir konumdan alıntı yapılmıştır.
- **13.** Yardım sayfası içerisinde, aradığınız komut ile ilgili örnekleri görmek isterseniz hangi köprüyü tıklamanız gerekir?
 - A) Examples

C) Applies to

B) Specifics

- D) See Also
- 14. Online yardım almak için Help menüsünden hangi alt menüye tıklanması gerekir?
 - A) Visual Basic On The Web

C) Visual Basic Online Help

B) Microsoft On The Web

D) Visual Basic Help

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete geri dönerek tekrar inceleyiniz.

ÖĞRENME FAALİYETİ–2

AMAÇ

Görsel programlama ortamını doğru olarak kullanabilecek

ARAŞTIRMA

- Arayüz kavramının ne anlama geldiğini araştırınız.
- Programlama bileşeni ne demektir, araştırınız.

2. ARAYÜZ OLUŞTURMAK

Visual BASIC' de arayüz denilince ilk akla gelen *form*'dur. Form Visual BASIC ile kullanıcı arasında iletişimi kuran temel öğedir. Form bir penceredir ve üzerine yerleştirilen nesneler yardımıyla kullanıcı ile programın diyaloğu sağlanır. Aşağıdaki şekilde bir form örneği görüntülenmektedir. Formlar veri girişi, verilerin görüntülenmesi ya da bunların her ikisinin bileşimi şeklinde kullanılabilirler.



Şekil 2.1' de yer alan tüm nesneler birer *kontrol*dür. Komut butonları, yazı kutuları, etiketler vb. tüm kontroller formun üzerine, dizaynı yapan yani programı yazan kişi tarafından yerleştirilir. Visual BASIC size neyi nereye yerleştireceğiniz konusunda hiçbir önermede ya da zorlamada bulunmaz yani dizayn ve program tamamen sizin özgür düşüncelerinizin ürünüdür. Ayrıca sadece kontrollerin yerleşim ve boyutları değil, formun kendisinin de ekrana yerleşimi, boyutu, artalan rengi ve daha pek çok özelliği kullanıcı tarafından ayarlanabilir.

Visual BASIC uygulamalarında kullanılan formlar çeşitli amaçlara göre tasarlanırlar. Örneğin yukarıda görülen formda veri kutuları boş olarak gelip kullanıcının bir ürün girişi yapmasına olanak sağlanabilir, ya da veri giriş kutuları herhangi bir veri ile dolu olarak gelip kullanıcının sırayla veriler arasında dolaşmasına ve düzeltme yapmasına olanak sağlanabilir. Tüm bu işlemler aynı form üzerinde hiçbir değişiklik yapmadan gerçekleştirilebilir. İşte bu noktada "bir formun işlem yapmasını ve yaptığı işlemin ne olduğunu, sizin yazacağınız program belirler" diyebiliriz.

2.1. Görsel Programlama Ortamı

Visual Basic programını kullanmaya başlamadan önce, bu programın temel ekran görüntüsü ve bu görüntü üzerinde yer alan temel bileşenler hakkında bilgi sahibi olmanız gerekir. İşte aşağıda verilen ekran görüntüsü üzerinde yapılan numaralandırmaya göre ekran bileşenlerini sırasıyla tanıyalım.



Şekil 2.2:Visual Basic temel ekran görüntüsü

2.1.2. Başlık Çubuğu

🙀 Project1 - Microsoft Visual Basic [design]



Şekil 2.3:Başlık çubuğu (tasarım durumunda)

Şekilde de gördüğünüz gibi başlık çubuğu iki bölümden oluşur. Birinci bölümde projemizin adı, ikinci bölümde ise daha önceden de görmeye alışık olduğunuz pencere konumlandırma ve kapatma düşmeleri yer alır. Bu başlık çubuğu görüntüsünden ayrıca projemizin şu anda tasarım durumunda olduğunu da anlayabiliriz. Çünkü eğer projemizi çalıştırırsak o anda bu başlık çubuğundaki [design] sözcüğünün yerini [run] sözcüğü alacaktır(Şekil 2.4.). Çalışmayı durdurduğumuzda yine eski halini alır.

🐂 Project1 - Microsoft Visual Basic [run]

Şekil 2.4:Başlık çubuğu (yürütme durumunda)

Yukarıdaki ekran görüntüsünde 🖲 ve 🌀 numaralarla gösterilen tasarım ortamını üzerinde bulunan ekran konumlandırma düğmelerini kullanarak tam ekran yaptığınızda ise başlık çubuğumuzun Şekil 2.5. de gösterilen hali aldığını görebiliriz.

🙀 Project1 - Microsoft Visual Basic [design] - [Form1 (Form)]

Şekil 2.5:Başlık çubuğu (tasarım ekranı, tam ekran yapıldığında)

Burada Project1 çalışılan projeye Visual Basic tarafından verilen takma adı, Form1 ise henüz kaydetme işlemi yapmadığımız ve bir isim vermediğimiz için yine Visual Basic tarafından üzerinde çalıştığımız forma verilen adı göstermektedir. Bu ekran durumunda iken ekran konumlandırma düğmeleri de Şekil 2.6'da gösterilen hali alır.

_	8	×
_	8	×

Şekil 2.6:Ekran konumlandırma düğmeleri(tasarım ekranı, tam ekran yapıldığında)

Burada üstteki düğmeler Proje1'e, alttakiler ise Form1'e aittir. Üsttekilerden kapatma seçilince Visual Basic, alttakilerden kapatma seçilince Form1 kapatılır.

2.1.3. Menü Çubuğu

File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help

Şekil 2.7:Menü çubuğu

Gördüğünüz gibi Visual Basic 6.0 menü çubuğunda 13 adet menü yer almaktadır. Ayrıca her bir menünün içerisinde de farklı alt menüleri olan seçenekler bulunur.

Menüleri, konularımız ilerledikçe ve yeri geldiğinde uygulama yaparken tanıyacaksınız. Bu sebeple şu anda menülerin içeriklerinde neler olduğuna tek tek bakmak çok da gerekli değildir. Fakat, *File, Edit, View, Window, Help* menülerinin daha önceden kullanmış olduğunuz Windows tabanlı uygulamalarla büyük benzerlikler gösterdiği söyleyebiliriz.

2.1.4. Araç Çubuğu

I	bað - ∛a - 1≣	🚔 🔲	X BAR MA 10 01	🖕 m = 😹 🕾 🖻 🕾 🛠 🛱 🕅 Int. (d)	

Şekil 2.8:Standard araç çubuğu

Araç çubukları sıklıkla kullanılan komutlara, her defasında menülerden ulaşmak yerine ekranda yer alan tek bir simgeyi tıklayarak, daha kolay ulaşmamızı sağlar.

Visual Basic ilk açılışında yukarıda gördüğünüz "*Standard araç çubuğu*" ile açılır. Diğer araç çubuklarını ekranda görüntülemek için ise iki farklı yol izlenebilir.

Birinci yol Şekil 2.9'da gösterildiği gibi standard araç çubuğunun üzerinde farenizin sağ tuşunu tıklayarak, açılacak olan menüden seçim yaparak,

İkinci yol ise Şekil 2.10'da gösterildiği gibi, *View* menüsünü kullanarak *Toolbars* Alt menüsü içerisinden seçim yapmak suretiyledir.

S • 5 •	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	♀ ▶ ■ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	Ln 1, Col 1 Debug Edit Ecom Editor	
	 Project1 - Form1 (Form) 	P	✓ Standard	
	Form1	Contraction (General)	Form1 (Code) Microsoft Vis	io UML Solution

Şekil 2.9:Araç çubuklarını görüntülemek için birinci yol



Şekil 2.10: Araç çubuklarını görüntülemek için ikinci yol



Şekil 2.11: Diğer araç çubukları

Yukarıdaki şekilde ise Visual BASIC'in diğer araç çubukları görüntülenmiştir.



Şekil 2.12: Araç kutusu

Çalışma alanımızın sol tarafında yer alan ve çalışmalarımız sırasında sıkça ihtiyaç duyacağımız "*kontrol*" adı verilen nesneleri barındıran penceredir. Ekranda varsayılan olarak gelen bu pencere görülmediği durumlarda "*View*" menüsünden Toolbox komutu ile çıkartılır. Visual BASIC'de yer alan kontroller bunlarla sınırlı değildir. İlerleyen modüllerde diğer kontrollerin de nasıl eklendikleri konusunda bilgi edineceksiniz.

2.1.6. Form Taşıyıcı Penceresi



Şekil 2.13: Form taşıyıcı penceresi

Form taşıyıcı penceresi proje tasarım aşamasındayken form yüzeyinin taşınmasını sağlar, böylelikle tasarım sırasında form yüzeyine nesneleri yerleştirirken isterseniz formunuzu tam ekran yapabilir ya da başka bir formda çalışmak için formunuzu simge durumunda küçülterek gizleyebilirsiniz.

2.1.7. Form Yüzeyi



Şekil 2.14: Form yüzeyi

Form da bir Visual BASIC nesnesidir. Kullanım amacı ise kullanıcı arayüzüne zemin oluşturmasıdır. Diğer nesneler Form yüzeyine yerleştirilerek arayüz oluşturulur. Formların özellikleri, diğer tüm nesnelerin özellikleri gibi programcı tarafından ayarlanabilir. Bunlar formun tam ekran olarak gelmesi, simge durumunda gelmesi, yüksekliği, genişliği, arka zemin rengi, başlığı,...vb. gibi kullanıcı tarafından ayarlanabilen özelliklerdir. Formun ve nesnelerin özelliklerinin ayarlanmasına ilerleyen bölümlerde değinilecektir.

2.1.8. Kod Penceresi



Şekil: 2.15. Kod Penceresi

Metin tabanlı programlama dillerinden daha önce bahsetmiş ve bir BASIC programı kod ekranını sizlere göstererek, Visual BASIC'in nesneye dayalı bir programlama tekniği olduğunu söylemiştik. BASIC dilindeki programın tüm adımlarının satır satır yazılması ve çalıştırılması işleminden farklı olmakla beraber, Visual BASIC'de de nesnelerin form üzerine yerleştirilmelerinin ardından bu nesnelerin durumlarına göre, yapacakları işleri tayin etmemiz gerekecektir.

Örneğin program çalıştırıldığı zaman, form üzerine yerleştirilmiş bir butonun tıklanmasıyla ekrana "BUTONA BASILMIŞTIR" yazdırılmasını isteyebiliriz.

Visual BASIC'deki bu ve buna benzer pek çok durumda yapılacak olaya yönelik işin yazılması işlemine "kod yazımı" denilmektedir.

Visual BASIC'de kod yazımı, kod penceresi içerinde, hangi nesneye ait ve o nesnenin hangi olayına ilişkin ise ilgili prosedüre yazılacaktır. Bu konu ile ilgili örnekler ilerleyen sayfalarımızda verilecektir.



2.1.9. Proje Penceresi

Şekil 2.16: Proje penceresi

Visual Basic uygulamalarında formlar ve modüller kullanılır. Basit uygulamalarda form sayısı bir ya da iki iken ve modüle gerek duyulmazken, uygulamanın derinliği arttıkça daha çok forma ve modüle ihtiyaç duyulabilir. Modüller ise yine gelişmiş uygulamalarda kullanılmakla birlikte, basit bir tanımlama yapmak gerekirse, BAS dosya uzantısına sahip, içerisinde prosedürlerin yer aldığı, pek çok uygulamada çağırılıp kullanılabilen dosyalardır. Formlar gibi bir arabirim elemanı olmadıklarını da söylemek gerekir.

Yukarıdaki şekilde gördüğünüz proje penceresi ile çalışma alanınızda yer alan bir ya da daha fazla proje ile aynı anda çalışabilir, formlar arasında geçiş yapabilir, projenize yeni form ve modül ekleyebilir, çıkartabilirsiniz. Üzerinde çalışmakta olduğunuz formun kod ekranına ya da tasarım ekranına sol üstte görülen butonlar yardımıyla geçiş yapabilirsiniz.

8

butonunu kullanarak ya da

2.1.10. Özellikler Penceresi

Proje penceresine; standart araç çubuğu üzerinden view menüsünden Project Explorer komutu ile ulaşabilirsiniz.

Properties -	Form1		Х
Form1 Form 💽			
Alphabetic	Categor	ized	
(Name)	Fo	rm1	•
Appearance	1 -	3D	
AutoRedraw	Fa	lse	
BackColor		&H8000000F	
BorderStyle	2 -	Sizable	
Caption	Fo	rm1	
ClipControls	Tru	Je	
ControlBox		Je	
DrawMode	13	- Copy Pen	
DrawStyle	0 -	Solid	
DrawWidth	1		•

Şekil 2.17: Özellikler penceresi

Visual BASIC'de kullanılan tüm nesnelerin kendilerine ait özellikleri vardır. Nesne ile ilgili boyut, renk, yazı tipi, başlık, nesne adı ve daha pek çok özellik, özellikler penceresi ile ayarlanır. Nesneler başlangıçta, varsayılan değerleri ile gelir, eğer değiştirmezseniz varsayılan değerler geçerli sayılacaktır. Örneğin üstteki şekilde Form1'e ait özellikler görülmektedir. *Form1* formun varsayılan adıdır ve (*Name*) özelliği alanında görülmektedir, *Caption* özelliği formumuzun üst başlık bilgisidir ve varsayılan olarak o da *Form1* şeklindedir. Üst tarafta iki ekran olduğu dikkat çekmektedir; "*Alphabetic*" sayfasında tüm özellikler alfabetik sıraya göre, "*Categorized*" sayfasında ise farklı özelliklere göre gruplandırılmış olarak görülebilir. Örneğin görüntü ile ilgili olanlar "*Appearance*" yani görünüm başlığı altında sınıflandırılmıştır.

Proje Penceresine; standart araç çubuğu üzerinden butonunu kullanarak ya da view menüsünden Properties Window komutu ile ulaşabilirsiniz.
2.1.11. Form Yerleşim Penceresi



Şekil 2.18: Form yerleşim penceresi

Projenizde birden fazla form ile çalışabileceğinizi söylemiştik, proje çalıştırıldığında bu formların ekranın neresinde görüntüleneceğini ise bu form yerleşim penceresi ile ayarlayabilirsiniz. Bu ekran üzerinde formları fareniz yardımıyla sürükleyip istediğiniz yere bıraktığınızda, program çalıştığında formun orada görüntülendiğini göreceksiniz.

2.2. Görsel Programlama Mantığı

Visual BASIC'de oluşturulacak programların en temel öğeleri formlardır. Formlar oluşturduğunun programın kullanıcı tarafından görülecek yüzüdür, aynı zaman da kullanıcı ile program kodları arasında iletişimi sağlayan kontrollerin, nesnelerin, menülerin taşındığı penceredir.

Visual Basic, Windows tabanlı pek çok uygulamanın (*Word, Excel, Powerpoint, Access, Paint, Hesap Makinesi, Oyunla vb.*) hazırlandığı ortamdır. Böyle düşünüldüğünde Visual Basic ile yapılabilecek çalışmaların, hazırlanacak programların basit ve az zaman alan uygulamalar şeklinde olabileceği gibi, yapılacak detaylı çalışmalarla çok profesyonel programların da tasarlanabileceğini görürüz.

Ancak programlarımız basit ya da detaylı olsun, tek ihtiyacımız olan şeyin form üzerine nesneleri doğru yerleştirmek ve doğru programlamak olduğunu bilmeliyiz.

Genelde tüm programlarda, yapacağınız işlemler için ayrı pencereler açmaktadır. Örneğin, Word programında "Yazdır" komutunu verdiğinizde açılan pencereyi hatırlayın. İşte bu noktada da Visual BASIC'in birden çok pencereyle ya da formla, proje tasarlamamıza imkân sağladığını söylememiz gerekir.

2.2.1. Visual Basıc'in Avantajları

Visual Basic bu ve bunlara benzer daha pek çok özelliğiyle dünya üzerinde üç milyondan fazla programcının kullandığı bir dil halini almıştır. Visual BASIC'in avantajları nelerdir diye sorduğumuzda dört temel özellik sayabiliriz; Visual Basic "**olay güdümlü**" (*event-driven*) bir programlama dilidir. Olay güdümlünün anlamı tasarımcının uygulamaların kontrolünü elinde tutmasıdır. Tasarımcı fare ile bir nesneye tıklandığında ya da klavyeden bir tuşa basıldığında, hangi olayların yürütüleceğine karar verir. Visual Basic programınızda kullandığınız nesneler için, yapılacak olaya yönelik yazmış olduğunuz kodları destekler.

Visual Basic "**nesneye yönelik tasarım**" ilkelerini destekler. Bunun anlamı şudur; siz isterseniz programınızı "nesne" ismi verilen bölümlere ayırabilir, her bir nesneyi tek tek diğerlerinden bağımsız tasarlayabilir ve programın geri kalanından bağımsız olarak test edebilirsiniz. Bu nesnelerin özellikleri ya da olayları ile ilgili yapılacak olan geliştirmelerde, sadece ilgili nesne üzerinde çalışmak yeterli olacaktır, böylece tasarımcıya büyük kolaylık sağlanmaktadır, ayrıca kullanıcı ile etkileşime çok daha açık kodlar yazmanız mümkün olabilmektedir. Nesneye Yönelik Programlama da Visual Basic diğer programlama dillerine göre (Visual C++, Delphi ve diğerleri) eksikliklere sahip olmakla beraber, ihtiyaç duyulabilecek pek çok özelliğe de sahip olduğunu söylemek doğru olacaktır.

Microsoft, Visual Basic'i tam anlamıyla bir "**Windows Uygulamaları Geliştirme Sistemi**" olarak tasarlamıştır. Bu demektir ki yapacağınız programlar Windows ekranı şeklinde olacaktır. Diğer programlama dillerinde sizi oldukça zorlayacak olan, Windows pencerelerinin sahip olabileceği en teferruatlı özellikleri bile kolayca ekleyebileceksiniz. Bu hem programın hızlı tasarlanmasına yardımcı olacak, hem de programlarınızın kullanıcılar tarafından kolay anlaşılmasına ve kullanılmasına olanak sağlayacaktır.

Visual Basic çok büyük genişleme imkânlarına sahiptir. Programlarınızda ActiveX kontrolleri, DLL (Dynamic Linked Libraries- Hareketli Aktarım Kütüphaneleri) dosyaları, add-ins menüsü yardımıyla Microsoft'un sağladığı eklentileri ya da sayısız firma tarafından üretilen ve internetten edinebileceğiniz eklentileri kullanabilirsiniz. Ayrıca Visual Basic 6 ile kendinizde ActiveX kontrolü, DLL ve eklenti oluşturabilirsiniz. Zaten Visual BASIC'i program geliştiriciler için en cazip kılan konuların başında da, bu kişi ya da firmaların diğer Windows tabanlı uygulamaların içerisinde kullanılabilecek (Word, Excel gibi) ActiveX kontrolü ya da DLL' ler yaratma çabaları gelmektedir.

Visual BASIC'in avantajlarını saydıktan sonra, Visual BASIC'de program yazımının nasıl yapılacağından da bahsetmemiz gerekir. Çünkü az önce sözünü ettiğimiz gibi bir form üzerine nesneleri yerleştirmek yetmemekte, birde bu nesnelerin hangi durumlarda, ne gibi işler yapacağının belirtilmesi gerekmektedir.

2.2.2. Özellik (Property), Metot ve Olay Kavramları

Bir forma yerleştirilen öğeler, kontroller, nesneler ve menü'dür. Bunlar için yapılabilecek renk, yazı tipi, boyut, konum gibi ayarlamalara *Özellik (Property)* denir.

Bir kontrolün ya da nesnenin özelliğini öğrenmek ya da değiştirmek için, önceki bölümde söz ettiğimiz Özellik Penceresi ya da Kod Yazımı Yöntemi kullanılır. Özellik penceresi, nesneyi ya da kontrolü seçtikten sonra alt alta listelenen başlıkların değiştirilmesi şeklinde olmaktadır. Kod yazımı ise, nesneyi seçtikten sonra üzerinde çift tıklama yapılarak

açılan kod penceresinden nesne için gereken özelliğin yazılması şeklinde olmaktadır. Genel kullanım şekli:

Nesne.özellik=değer şeklindedir.

Örnekler:

Command1.Caption = "Çıkış"	'Command1 isimli komut tuşunun başlığı "Çıkış"
yapıldı.	
Text1.Height= 500	'Text1 isimli metin kutusunun genişliği 500 pixel
yapıldı.	
List1.MousePointer = 13	'List1 isimli listenin mouse göstergesi "oklu kum
saati" yapıldı.	
Label1.Caption = Text1.Text	'Text1'in içeriği Label1'in başlığına yazıldı.
Label1.BackColor = 1000	'Label1 nesnesinin arka alanı kırmızı yapıldı

Özellikler pek çok nesne için aynı olmakla beraber, bazı nesnelere ait farklı özellikler de olabilmektedir. Örneğin Data kontrolüne ait Database özelliği, veri tabanı kullanımı ile ilgili olmayan diğer nesnelerde bulunmaz.

Hangi öğelerin hangi özelliklere sahip olduğunu yardım menülerinden öğrenebilirsiniz. Formunuzun üzerine yerleştirdiğiniz herhangi bir kontrolü seçip F1'e basarsanız, açılan pencerede o kontrolle ilgili yardım alabilirsiniz. Ve açılan ekrandan Properties köprüsünü tıklayarak özellikleri görebilirsiniz.



Şekil 2.19: Bir nesnenin ya da kontrolün özelliklerinin görüntülenmesi

Örnek:

Form üzerine yerleştirilen bir komut butonunu seçip özellikler penceresini inceleyelim.

Araç kutusundan Command Buton kontrolünü üzerine bir kez tıklayarak seçiniz.

Form1 isimli form yüzeyine geçtiğinizde fare işaretçinizin "+" şeklini aldığını göreceksiniz. Bu durumda farenin sol tuşunu tıklayıp, bırakmadan sürükleyerek istediğiniz büyüklükte komut düğmesi oluşturunuz.

Form üzerindeki komut düğmesini seçip, ekranın sağ köşesinde yer alan Özellik Penceresini (Properties Window) inceleyiniz.



Şekil 2.20: Bir nesnenin ya da kontrolün yerleştirilmesi ve özelliklerinin görüntülenmesi

- Caption yani başlık özelliğini "Buton" olarak değiştirin.
- Height yani yükseklik özelliği 700 yapın.
- Şimdi başlığını ve yüksekliğini değiştirdiğiniz butonunuzu ve formunuzu çalışırken görmek için klavyeden F5 tuşuna basın.
- Calışan projenizi penceresinden kapatarak sonlandırın.

- Projenize geri döndükten sonra form üzerinde herhangi bir boş alanı çift tıklayın. Kod penceresinin açıldığını göreceksiniz.
- İmlecin konumlandığı noktada Formunuzun Load yani başlangıç olayının içindesiniz bu alana Command1 yazıp nokta koyduktan sonra bir listenin açıldığını göreceksiniz.



Şekil 2.21: Bir nesnenin ya da kontrolün özelliklerinin kod penceresinde görüntülenmesi

- Bu listeden Height özelliğini bulup, üzerinde sekme (TAB) tuşuna basınız ve ardından =1000 yazınız.
- Ardından alt satıra inip tekrar Command1. yazdıktan sonra açılan listeden Caption özelliğini bulup, üzerinde sekme tuşuna bastıktan sonra karşısına ="DUGME" yazınız.



Projenizi F5 tuşuna ya da yukarıdaki Standard araç çubuğunda yer alan butonuna basarak çalıştırınız.

Sonuç:

Formunuza yerleştirdiğiniz tüm kontrollerin ve nesnelerin özellikleri hem özellik penceresinden hem de kod ekranı kullanılarak ayarlanabilir.

Metot; öğenin kullandığı komuttur. Başka bir ifadeyle öğeye yapacağı işin söylenmesidir.

Her öğenin programımız açısından gerekli olan özelliklerini tanımlamamız gerektiği gibi, programımızın işleyişinde bu nesnenin, nasıl bir role sahip olduğunu, hangi durumda ne yapacağını ya da nelere sebep olacağını belirtmemiz gerekir.

Metotların belirtilmesi kod yazımıyla mümkündür. Nesnemizin ya da kontrolümüzün üzerinde çift tıklandığında açılan kod penceresine o nesne ile ilgili metod yazılır. Genel Kullanım sekli:

Nesne.Metot = [Parametreler] şeklindedir.

oject1 - Form1 (Form)	
Project1 - Form1 (C	ode)	
Form		- Load
Private Sub label1. End Se Mouse Move Move Move Se Name OLEDr DEDr Parent	Form_Load () Pointer	

Şekil 2.22: Bir nesnenin ya da kontrolün metodlarının görüntülenmesi

Kod penceresine geçtiğinizde ve kod yazacağınız nesnenin adını verip noktaya bastığınızda daha öncede gördüğünüz gibi açılan pencerede bir takım özellikler listelenecektir, bu listede özellikler 🍽 sembolü ile metodlar ise 🎂 sembolü ile gösterilmektedir.

Örnekler:

ä

Form1.Printform Form1'i yazıcıya gönderir. Form1.Line (10, 50)-(3000, 50) Form1 üzerinde verilen koordinatlara çizgi çizildi. Form1 yüzeyi temizlendi Form1.Cls

Metotların arasında da özellikler de olduğu gibi, her kontrol ya da nesnede geçerli olanlar ve öğeye özgü olanlar bulunmaktadır.

Hangi öğelerin hangi özelliklere sahip olduğunu yardım menülerinden öğrenebilirsiniz. Formunuzun üzerine yerleştirdiğiniz herhangi bir kontrolü seçip F1'e basarsanız, açılan pencerede o kontrolle ilgili yardım alabilirsiniz. Ve açılan ekrandan Methods köprüsünü tıklayarak özellikleri görebilirsiniz.

Örnek:

Metotları kullanabileceğimiz bir örnek geliştirelim.

Formunuzun üzerine dört adet buton yerleştirip, başlıklarını aşağıdaki gibi düzenleyiniz.



Şekil 2.23: Örnek ekran görüntüsü.

"FORMA YAZ" başlıklı ve adı Command1 olan butonunuzun üzerinde çift tıklayınız. Açılan kod ekranına:Print "MERHABA" yazınız. "Formun üzerine MERHABA yazdırdık"

Tekrar form tasarım ekranına dönerek "FORMU TEMIZLE" başlıklı ve adı Command2 olan butonunuzun üzerinde çift tıklayınız. Açılan kod ekranına:Form1. Cls yazınız. "Bu form1'in Cls metodudur". Formun yüzeyini temizler."

Form tasarım ekranına dönerek "FORMU YAZDIR" başlıklı ve adı Command3 olan butonunuzun üzerinde çift tıklayınız. Açılan kod ekranına:Form1. PrintForm yazınız. "Bu form1'in PrintForm metodudur. Form1'i yazıcıya gönderir."

Tekrar form tasarım ekranına dönerek "FORMA ÇIZ" başlıklı ve adı Command4 olan butonunuzun üzerinde çift tıklayınız. Açılan kod ekranına:Form1. Line (100,500) - (1000,50) yazınız. "Bu form1'in Cls metodudur". Form1 üzerinde verilen koordinatlara uygun çizgi çizer."



Şekil 2.24: Örnek ekrana uygun kodlar

Visual Basic Programlarınızı yazarken, bir formun üzerine kontrol ve nesneler yerleştirebileceğinizi söylemiş ve bunların her birinin özellikleri ve metotları vardır demiştik. İşte bu kontrol ya da nesnelerin bir şekilde uyarılmasına da "olay" denilmektedir. Örneğin bir komut tuşunu tıklamanız, klavyede bir tuşa basmanız, fareyi hareket ettirmeniz, bir formu yüklemeniz, bir metin kutusunun içeriğini değiştirmeniz ya da bir ComboBox (Seçim Kutusu)' tan seçim yapmanız bir olaydır. Bu bahsedilen işlerden birini yaptığınızda ilgili kontrol uyarılır ve bir olay gerçekleşmiş olur.

Örnek verecek olursak, fareyi tıkladığımızda "Click" olayı, bir tuşa basıp bıraktığımızda "KeyPress" olayı, bir formu büyülttüğünüzde "Resize" olayı ya da bir metin kutusu üzerinde iken Tab tuşuna basarsanız "LostFocus" olayı gerçekleşir.

Hangi kontrolün hangi olaylara sahip olduğunu yine VB yardım menüsünden öğrenebilirsiniz. Örnek olarak CommandButton'un sahip olduğu özellik, metot ve olayları görmek için yardım menüsünde CommandButton'u seçtiğinizde ya da formunuzda CommandButton'u seçip F1'e bastığınızda, aşağıdaki pencere karşınıza gelecektir. Bu pencerede Properties linkine tıklarsanız bu kontrolun sahip olduğu özelliklerin, Methods linkine tıklarsanız sahip olduğu metodların ve Events linkine tıklarsanız bu kontrolun sahip olduğu olayların listesine ulaşırsınız. Bu listeden bir seçim yaptığınızda ise seçiminizle ilgili yardım ekranı karşınıza gelecektir. Bu ekranda seçiminizle ilgili kullanım şekli, ve örnekler yer alacaktır.

😵 MSC	IN Libi	rary Vis	ual Stu	dio 6.0							
Qosya	Dügen	Görünüŋ	a gat t	(ardm							
Göster	Bul	0rceki	J. Sorraki	< Geri	⊑> Ileri	Our Dur	() Yenile	Giriş	Yazder		
Co	mma	andB	utto	on C	onti	ol					^
<u>See</u>	Also	Exam	le l	Proper	ties	Metho	dsE	vents	>		
Use a When some	a Con n chos times	nmand ien, a (called	Buttor Comm a push	n contr andBu butto	rol to l itton a n.	oegin, appear	interrup s pushe	ot, or ed in a	end a pr ind so is	ocess.	
Synt	ax										
Com	mano	Butto	n								
Rem	arks										

Şekil 2.25: Bir nesnenin ya da kontrolün yardım ekranı

Örnek:

CommandButton nesnesi için girilebilecek olayların listesini inceleyelim.
 Formunuzun üzerine bir komut düğmesi yerleştiriniz.

🗧 Form1													_	L		×
	1::::			• •	-		ł		ł	÷	÷	-		÷	***	
1					-			-	-	-	*	-			***	
: Con	imand1				1		1		1	1	1	1		1	1	
÷.,	202	1.3	11	11	÷	ŧ	ŧ.		÷	÷.	÷	ŧ	ŧŧ	÷	*	
	a series and series and			2.3		2.3						12	5 5		12	2.3

Daha sonra üzerini çift tıklayınız.

•
-

Şekil 2.27: Kod yazım ekranı

Karşınıza gelen kod penceresindeki olayın Command1_Click yani Command1'in tıklanması olduğunu göreceksiniz.

- Kod ekranının sağ tarafında nesne ya da kontrolün adı, solda ise olayın yazdığı dikkatinizi çekecektir.
- Şimdi Command1 nesnesi için kullanılabilecek diğer olayları görmek için sağ taraftaki olay kutusunun yanındaki oku tıklayarak listeyi açıp olayları inceleyiniz.

Project1 - Form1 (Code)					
Command1	Click				
Private Sub Command1_Click() End Sub	Click DragDrop DragOver GotFocus KeyDown KeyPress KeyUp LostFocus MouseDown MouseDown MouseMove MouseUp OLECompleteDrag				

Şekil 2.28: Command1 nesnesinin olayları

- Visual Basic'de kontrollere ait olayları yazarken aslında o kontrole ait bir olay için prosedür yazmış oluruz.
- Visual Basic Programlama Dili temel olarak nesnelere ya da kontrollere ait prosedürlerin yazılmasından ibarettir.

2.3. Temel Görsel Programlama Bileşenleri

Bu bölümde, projelerinizde kullanacağınız temel kontrol ve nesnelerden önemli görülen ve ilerleyen bölümlerde yapılacak örneklerde kullanılacak olanların tanıtılması hedeflenmiştir.

2.3.1. Command Button (Komut Düğmesi)

Visual Basic kontrolleri içerisinde en çok kullanılanı komut düğmeleridir. Komut düğmeleri, tıklandıkları zaman genellikle bir işlemi başlatır, durdurur ya da bitirir. Komut düğmelerinin hemen hemen tüm programlarda kullanıldığını söyleyebiliriz.

👟 Form1	_OX
TIKLA	ITINIZ!!!

Şekil 2.29: Forma yerleştirilmiş bir komut düğmesi

2.3.1. Komut Düğmesi Özellikleri

Properties - co	mmand1 🛛 🗵			
command1 Cor	mmandButtor 💌			
Alphabetic Cat	egorized			
(Name)	command1			
Appearance	1 - 3D			
BackColor	8H00FFFF0			
Cancel	False			
Caption	&TIKLAYINIZ!!			
CausesValidation	True			
Default	False			
DisabledPicture	(Icon)			
DownPicture	(None)			
DragIcon	(None)			
DragMode	0 - Manual			
Enabled	True			
Font	Comic Sans MS			
Height	615			
HelpContextID	0			
Index				
Left	960			
MaskColor	8H00C0C00			
MouseIcon	(None)			
MousePointer	0 - Default			
OLEDropMode	0 - None			
Picture	(None)			
RightToLeft	False			
Style	1 - Graphical			
TabIndex	0			
TabStop	True			
Tag				
ToolTipText				
Тор	840			
UseMaskColor	False			
Visible	True			
WhatsThisHelpID	0			
Width	2895			
C aption Returns/sets the in an object's title	text displayed bar or below an			

Name (İSİM) özelliği:Form üzerine yerleştirilen butonlar otomatik olarak Command1, Command2,... şeklinde isimlendirilir. Eğer isterseniz bu özelliği kullanım amacınıza uygun olarak değiştirebilirsiniz. Ancak, kullanacağınız kontrollerin, gün geçtikçe çoğalacağı ve çeşitleneceğini düşünecek olursak kontrollere isim verirken bu bir komut butonuysa cmdbasla, cmdyazdir, cmdtemizle gibi isimler vermek yerinde olur.

Not: Visual Basic Programınızda kod yazarken bu isimleri çağıracağınızı ve büyük küçük harflere dikkat etmeniz gerektiğini unutmayınız.

Caption (BAŞLIK) özelliği: Komut düğmesi üzerinde görüntülenecek yazının belirlenmesini sağlar. Örneğin yukarıdaki şekilde komut düğmesinin başlığı TIKLAYINIZ!!! olarak belirlenmiştir.

Butonun başlığı yapacağı işi anlatmalıdır bu kullanıcıya kullanım kolaylığı sağlar. Butona başlık verilirken harflerden herhangi birinin önünde & işareti kullanılırsa bu sayede o harfin altı, yukarıdaki şekilde gördüğünüz gibi çizilir ve ALT+*harf* şeklinde butona ulaşılabilir. Örneğin üstteki şekilde ALT+T ile TIKLAYINIZ!!! butonuna ulaşılabilir. Bu özelliği kod yazarak da verebilirsiniz.

Kodu: Command1.Caption= "&TIKLAYINIZ!!!" Seklinde olacaktır.

Style(Sitil) özelliği:Bu özellik Standard ya da Graphical olabilir. Bu özelliği Graphical yaparak butonun üzerine resim konulması, artalan rengi verilmesi, üzeri tıklandığında bir resim görüntülenmesi gibi bir takım özelliklere izin vermiş olursunuz.

BackColor(Artalan) özelliği:Butonun arkaplan renginin belirlenmesini sağlar. Sitil Özelliği Graphical olmalıdır. Örneğin yukarıdaki şekilde artalan rengi açık mavi seçilmiştir.

Enabled(**Aktiflik**) özelliği:Butonun kullanılıp kullanılmayacağını belirleyen özelliktir. True ve False yapılarak butonun tıklanabilir olup olamayacağı belirlenir.

💐 Form1		<u>_ </u>
	ba <u>s</u> la	
	<u>durdur</u>	
	<u>b</u> itir	

Şekil 2.30: Aktif ve pasif butonlar

Yukarıdaki şekilde basla ve durdur butonları aktif iken, bitir düğmesi pasiftir, tıklanamaz, yani Enabled özelliği False yapılmıştır.

Font(Yazı Biçimi) Özelliği:Komut düğmesi üzerine yazılacak olan yazının türü ve boyutu buradan düzenlenir. Tüm kontrollerde olan bir özelliktir.

Picture(**Resim**) Özelliği:Butonun üzerinde bir resim görüntülemek için kullanılır. Daha önceden butonumuzun Sitil özelliği Graphical seçilmiş olmalıdır. Resmin boyutuna dikkat etmek gerekir, eğer büyük boyutlu resimler yerleştirilirse, butonumuzun boyutu yetersiz kalacağından resim eksik görüntülenebilir. Picture özelliği tıklandığında açılan iletişim kutusuyla resim bulunabileceği gibi, kod penceresiyle de adres belirtilip resim yerleştirilebilir.

Kodu:

Command1.Picture.LoadPicture ("C:\Program Files\Microsoft Visual Studio\Common \Graphics\Icons\Computer\key04.ico) şeklinde verilebilir.



Şekil 2.30: Üzerlerine resim yerleştirilmiş butonlar

DisabledPicture ve DownPicture özellikleri:Bu iki özellik Picture özelliği gibi düğmenin üzerine resim yerleştirmek için kullanılır. DisabledPicture butonun Disabled özelliği true yapıldığı yani butonun kulanım dışı ya da pasif bırakıldığı durumlarda görüntülenecek olan resmi, DownPicture özelliği ise butonun üzerine tıklandığında görüntülenecek olan resmi belirlenmek için kullanılır.

Visible(görünürlük) özelliği:Bu özellik form üzerine yerleştirilmiş olan komut düğmesinin program çalıştırıldığında görüntülenip görüntülenmeyeceğinin belirlenmesini sağlar. True olduğunda buton görülür, False yapıldığında ise görüntülenmez. Örneğin programın çalışması esnasında, "ÇIKIŞ" butonunun gözlenmesi gibi amaçlarla kullanılabilir ya da yetkisiz kullanıcıların programı açtıklarında "KASA HESABI" butonunu görmeleri engellenebilir.

Command1.Visible=True Command1.Visible=False

Default(varsayılan) özelliği:Programınız çalıştığında hangi butonun şeçili geleceğine, başka bir ifadeyle program çalıştığında Enter' a basmanın hangi butona basmak anlamına geleceğine karar verilmesini sağlar. Aynı form üzerinde yer alan sadece bir butonun Default özelliği True yapılabilir.

Cancel(çıkış) özelliği:True yapıldığında, Esc tuşuna basmak ile düğmeye tıklamak aynı işlemi yerine getirir. Bu özellik yalnızca tek bir komut düğmesi için True yapılabilir. Örneğin, programdan çıkmak için sadece ESC tuşuna basmak yeterli olacaktır.

2.3.2. Textbox (Metin Kutusu)

Metin kutularını iki farklı amaç için kullanabiliriz. Bunlardan ilki, kullanıcının programa veri girişi yapmasını sağlayacak bir ekran alanının yaratılması, ikincisi ise programın çalışmasına bağlı olarak elde edilen sonuç ya da bilgilerin kullanıcıya gösterilmesini sağlayan bir ekran alanı oluşturulmasıdır.

2.3.2.1. Metin Kutusu Özellikleri

Appearance (görünüm) özelliği: Metin kutuları ve diğer pek çok kontrol düz(flat) ve 3 boyutlu (3D) görünüm alabilir. Bu özelliğin değeri 0 olduğunda düz, 1 olduğunda ise 3 boyutlu görünüm seçilmiş olur.

Text1.Appearance=0 şeklinde kod ekranından da girilebilir.

Alignment (yaslanma) özelliği: Yazının metin kutusu içerisinde nasıl hizalanacağını belirler.

0	Left Justify	Sola yanaşık (varsayılandır)
1	Right Justify	Sağa yanaşık
2	Center	Ortalı

Locked (kilitli) özelliği:Bu özellik seçilirse metin kutusunun içerisine yazılı bilgiler değiştirilemez.Varsayılan olarak False gelir, özelliği aktif etmek için ise True yapmak gerekir.

MaxLength (maksimum uzunluk) özelliği: Buraya yazacağınız değer, hazırlamış olduğunuz metin kutusu alanına kullanıcının en fazla kaç karakterlik bilgi girebileceğini belirtir. Bu özelliğin varsayılan değeri 0 ' dır. Bu sınırsız anlamına gelmektedir.

MultiLine (çoklu satır) özelliği: Girilecek değer birden çok satırı içeriyorsa bu özellik True yapılır ve kullanıcı isterse satır sonunda Enter'a basarak bir alt satıra geçip bilgi girişine devem edebilir. False değeri ise, kullanıcının girişini tek satırlık bir metin olarak sınırlar.

PasswordChar (şifreli karakter) özelliği:Bu özellik, metin kutusuna bir parola alanına benzer özellik kazandırır. Parola alanında kullanıcının yazdıkları ile ekranda görülenler arasında farklılıklar gözlenebilir. Örneğin siz şifreyi girerken metin kutusunun içinde her karakterin yerine bir * işareti gösterilebilir. Bu alana hangi karakter girilirse ekranda o gözlenir.

ScrollBars (kaydırma çubukları) özelliği:

0 Metin kutusunda kaydırma çubuğu yok .

1 Metin kutusunda sadece yatay kaydırma çubuğu var .

- 2 Metin kutusunda sadece dikey kaydırma çubuğu var.
- 3 Metin kutusunda hem dikey hem de yatay kaydırma çubuğu var

Text (yazı) özelliği:Text özelliği kullanıcıya programda metin kutusu içerisine bilgi yazdırmak, ya da metin kutusu içerisindeki bilgiyi almak için kullanılır.

Text1.Text= "Adınızı Giriniz" Text1'in içerisinde "adınızı giriniz" yazacaktır.

Ya da Text2.Text=Text1.Text Text1'in içinde ne varsa alınıp Text2'ye de aynısı yazılacaktır.

Örnek:

- Form üzerine iki textbox ve bir CommandButton yerleştiriniz.
- Commandbutton'un Caption özelliğini Kopyala olarak değiştirin.



Şekil 2.31: Üzerlerine kontroller yerleştirilmiş form tasarım ekranı

- Formun üzerinde herhengi bir alanı çift tıklayıp, açılan kod ekranına Text1.Text= "Adınızı Giriniz" yazınız
- Form tasarım ekranına geri dönüp, CommandButton'un üzerinde çift tıklayınız, açılan kod ekranına: Text2.Text=Text1.Text yazınız.

Project1 - Form	1 (Code)	_ [] ×
Form	▼ Load	•
Private S Text2.Tex End Sub	ub Command1_Click() t = Text1.Text	
Private S Text1.Tex End Sub	ub Form_Load() t = "Adinizi Giriniz'	

Şekil 2.31: Kod yazım ekranı

Programınızı çalıştırınız.

🐂 Form1	
Adinizi Giriniz	Kopvala

Şekil 2.32: Program açılış ekranı

Adınızı yazınız yazan alana isim ve soyadınızı girip Command1'e tıklayınız.

💐 Form1	<u>- </u>
Mustafa Nazman	Kopyala
Mustafa Nazman	

Şekil 2.33: Program çalıştırıldıktan sonrası

> Text2'nin de Text1 ile aynı içeriğe sahip olduğunu göreceksiniz.

ToolTipText (araç ipucu bilgisi) özelliği:Metin kutusunun üzerine mouse ile gelindiğinde metin kutusu hakkında açıklayıcı bilgi verir. Kullanıcılara kolaylık sağlamak adına çok kullanışlı bir özelliktir.

i Form1	×
Tanitici Notar	
Bu alana firmanizi ya da kendinizi tar	iitici bilgiler giriniz.
Ilgilenilen Birim	
Cevap Adresi	

Şekil 2.34: Alan ile ilgili açıklama notu

2.3.3. Label (Etiket)

t) A

Metin görüntülemeye yarayan ve programın çalışması esnasında kullanıcı tarafından içeriği değiştirilemeyen bir kontrol nesnesidir. İçerik Caption özelliği ile girilir ya da programın çalışması esnasında, yazılmış olan program kodları ile değiştirilir. Ekranda bir işlem sürerken, işlem hakkında bilgi vermek gibi amaçlarla kullanılabileceği gibi, veri girilecek metin kutusu alanları ve diğer kontroller hakkında bilgi vermek amacıyla sıklıkla kullanılır.

2.3.3.1. Etiket Özellikleri

AutoSize (otomatik boyutlandırma) özelliği: Etiketler tasarım aşamasında form üzerine yerleştirilir ve Caption(başlık) bölümüne yazıları yazıların boyutlandırması yapıldıktan sonra, eğer etiket alanına yazı sığmıyor ise kenarlarından çekerek etiket büyütülür. Bu özellik True yapılırsa, etiket, Caption özelliğiyle belirtilen metni, tasarım aşamasında yüzeyine sığdıramasa bile, herhangi bir büyütme yapmamıza gerek kalmaksızın, uygun biçimde sığdırmak üzere otomatik olarak büyür.

Backstyle(artalan sitili) ve BackColor(artalan rengi) özellikleri: Etiketin BackStyle özelliği Opaque ya da Transparent olabilir. Eğer Opaque seçilirse, BackColor özelliğinden etikete bir artalan rengi verilebilir, yok eğer Transparent seçilirse etiket saydam bir görünüm kazanır ve BackColor özelliği hiçbir işe yaramaz, etiketin üzerine yerleştirildiği formun BackColor özelliğinde verilmiş olan renk, etiketinde artalan rengi olur.

UYGULAMA FAALİYETİ

Aşağıdaki işlem basamaklarına göre uygulama faaliyetini yapınız.

İşlem Basamakları	Öneriler
Visual Basic programlama dili ekran arayüzünü açınız.	
Açılış ekranından "Standart.Exe" seçeneği ile yeni projenizi başlatınız.	
Araç kutusunu gizleyip tekrar görünür hale getiriniz.	
Standart araç çubuğunu gizleyip ardından tekrar görünür hale getiriniz.	
Proje penceresini gizleyip ardından tekrar görünür hale getiriniz.	
 Özellikler penceresini gizleyip ardından tekrar görünür hale getiriniz. 	 Visual Basic programlama dili arayüz ekranında bulunan tüm menüleri, araç
Form yüzeyinize istediğiniz kontrolleri yerleştiriniz ve kaldırınız.	çubuklarını ve diğer pencereleri, çalışmaya başlamadan önce üzerilerinde
Form yüzeyine yerleştirdiğiniz kontrollerin özelliklerini Özellikler penceresini kullanarak değiştiriniz.	 > Her adım öncesinde kitabınızdaki ilgili
Form yüzeyine yerleştirdiğiniz kontrollerin özelliklerini gerekli kodları yazarak değiştiriniz.	bolunlu bulup gözden geçiriniz.
 Form yüzeyine yerleştirdiğiniz kontrolleri kopyalamayı deneyiniz. 	
Form yüzeyine yerleştirdiğiniz kontrollerin aynı hizada ve aynı genişlikte olmalarını sağlayınız.	
Form yüzeyine yerleştirdiğiniz kontrollerle ilgili yardım menülerini kullanmayı deneyiniz.	

ÖLÇME VE DEĞERLENDİRME

OBJEKTİF TEST (ÖLÇME SORULARI)

Aşağıdaki soruları cevaplayarak bu faaliyette kazandığınız bilgileri ölçünüz.

- **1.** Visual Basic ile kullanıcı arasında iletişimi sağlayan temel öğe aşağıdakilerden hangisidir?
 - A) Komut Düğmeleri
- C) Formlar

B) Etiketler

D) Metin Kutuları

C) Yazı Kutuları

- **2.** Aşağıdakilerden hangisi bir form üzerinde bulunabilecek kontrollerden birisi değildir?
 - A) Kontrol Çubuğu
 - B) Komut Butonları D) Etiketler
- 3. Aşağıdakilerden hangisi Visual Basic temel ekran bileşenlerinden değildir?
 - A) Başlık ÇubuğuC) Menü Çubuğu
 - B) Standart Araç Çubuğu D) Debug Araç Çubuğu

4. Programın çalıştırılması esnasında başlık çubuğunda hangi ifade görülecektir?

- A) Progress C) Design
 - B) Execute D) Run
- **5.** Visual Basic araç çubuklarını görüntülemek ya da gizlemek için hangi menü kullanılır?
 - A) View C) Project
 - B) Edit D) File

6. Aşağıdakilerden hangisi Visual Basic araç çubuklarından birisinin adı değildir?

- A) Form Editor C) Edit
 - B) FormD) Debug

7. Kontrol adı verilen nesneleri ekran üzerinde barındıran pencerenin adı nedir?

- A) Kontrol Kutusu C) Araç Kutusu
 - B) Kontroller D) Araç Çubuğu
- **8.** Visual Basic'de gerçekleştirilecek olaya yönelik yapılacak olan işin yazılması işlemine ne ad verilmektedir?
 - A) Kod yazımı C) İş girişi
 - B) Kodlama D) İşleme

9. Proje Penceresine aşağıdaki butonlardan hangisi ile ulaşabilirsiniz?

A)	-8	C)	8
B)	**	D)	8

10. Özellikler Penceresi'ne (Properties Window) hangi menüden ulaşabiliriz?

- B) Format D) File
- **11.** Aşağıdakilerden hangisi Visual Basic programlama dilinin avantajlarından birisi değildir?
 - A) Olay güdümlü (event driven) bir programlama dilidir.
 B) Nesneye yönelik tasarım ilkelerini destekler
 C) Windows uygulamaları geliştirme sistemi olarak tasarlanmıştır.
 D) Hepsi
- **12.** "Bir forma yerleştirilen öğeler, kontroller, nesneler ve menü'dür. Bunlar için yapılabilecek renk, yazıtipi, boyut, konum gibi ayarlamalara denir."

A) Metod	C) Olay
B) Yöntem	D) Özellik

13. "Öğeye yapacağı işin söylenmesine denir."

A) Metod

- B) Yöntem
- **14.** Kod yazımı sırasında nesnelere verilebilecek özellikler ve metodlar listelendiğinde, metodların yanında hangi sembol yer alır?

C) Olay

D) Özellik



DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete geri dönerek tekrar inceleyiniz.

ÖĞRENME FAALİYETİ-3

AMAÇ

Görsel programlamada kurallara uygun olarak proje oluşturabileceksiniz.

ARAȘTIRMA

- Programlama yaparken kullanılan proje kavramı neleri içermektedir?
- Bir Visual Basic projesinde neler yer alır, araştırınız.

3. PROJELER VE BİLEŞENLERİ

- Su ana kadar öğrenmiş olduğunuz bilgilerin neler olduğunu düşünecek olursak;
- Visual Basic' i bilgisayarınıza başarıyla kurdunuz,
- Visual Basic uygulamasının nasıl başlatıldığını öğrendiniz,
- Formun ne olduğunu ve nasıl kullanılacağını biliyorsunuz,
- > Özellik, metot ve olay kavramlarını öğrendiniz,
- Nesne ve kontrol bileşenlerini tanıyorsunuz,
- En sık kullanılan kontrol bileşenlerini ve özelliklerini öğrendiniz,
- Nesne ve kontrollerin özellikleri nasıl girilir biliyorsunuz,
- Nesne ve kontrollerin olaylarına nasıl kod yazılır gördünüz.

Şimdi tüm bu öğrendiklerinizi bir araya toplayıp, bir Visual Basic projesi yaratabilirsiniz.

Visual Basic' de uygulama geliştirmek proje adı verilen yapı ile mümkün olabilmektedir. O halde her yeni Visual Basic uygulaması, yeni bir projeyi gerekli kılacaktır. Bir proje Form ya da formlar, formlar üzerine yerleştirilen kontrol nesneleri ve bunların varsa program kodları ve bağımsız modüllerden oluşur.

Proje dosyasının uzantısı. ".vbp" şeklindedir. Visual Basic 4.0, 5.0 ve 6.0 da bu uzantı kullanılmıştır, bunlardan önceki versiyonlarda Visual Basic proje dosyasının uzantısı .mak şeklindedir. Ancak, .mak uzantılı projeler de Visual Basic 6.0 tarafından tanınmaktadır.

3.1. Proje Türleri

Visual Basic Programını başlattığınızda ya da File(Dosya) Menüsünden New Project komutunu verdiğinizde, karşınıza içerisinde proje türlerinin listelendiği bir ekran gelmektedir.



Şekil 3.1: New project penceresi

New Project penceresi Visual Basic 6.0 ile neler yapılabileceğini özetlemektedir.

Standart EXE:

Genelde kullanılan ve sık kullanılan kontrollerin yer aldığı uygulamalar içindir. Bizim bu kitap içerisinde yapacağımız tüm alıştırmalar bu tipte olacaktır.

ActiveX EXE:

Farklı programlardan kendi bilgilerine erişim imkânı tanıyan programlar geliştirmek içindir.

ActiveX DLL:

Farklı programlar tarafından kullanılabilecek fonksiyon ve yordam kümeleri içeren kütüphanelerdir.

ActiveX Control:

Kullanıcının kendi kontrol nesnelerini tasarlayıp kullanmasına imkân tanır. Kullanıcının tasarladığı kontrol nesneleri ActiveX Kontrol nesneleridir.

ActiveX Document DLL:

İnternette yayınlanmak üzere doküman hazırlamak için kullanılır. Bu dokümanlar bir internet sunucu kullanılarak görüntülenmek istendiğinde içerilerinde yer alan işlevleri gerçekleştirebilmek için bu işlevleri içeren bir kütüphaneye ihtiyaç duyar. Bu kütüphaneler ActiveX Document DLL Kütüphaneleridir. Bu kütüphanelerdeki işlevler işlevi kullanan dokümanla aynı bellek alanında çalıştığı için işlem içi bileşenlerdir.

ActiveX Document EXE:

ActiveX Document DLL ile aynıdır. Farklı olan kesim, içerdiği işlevlerin işlem dışı olarak çalışmasıdır. İşlem içi bileşenler daha hızlı çalışma özelliğine sahiptir.

ADD IN:

Visual Basic ortamına bazı işlemlerin otomatikleştirilmesi için eklenebilen bir bileşen yaratmak için kullanılır.

VB Application Wizard:

Programın bir kısmını otomatik olarak üretmek için kullanılan bir bileşendir. Birçok seçeneği sayesinde programın kullanıcı arabiriminin çoğunu otomatik olarak üretir.

Data Project:

Visual Basic ile veritabanı uygulamaları yapmak amacıyla kullanılır. Veri tabanı uygulamaları ile ilgili kontrol nesnelerini ve proje bileşenlerini otomatik olarak açar.

DHTML Application:

Visual Basic ile web sayfası üzerinde çalışabilecek uygulamalar geliştirmek amacıyla kullanılır.

ISS Application:

Bir web sunucu üzerinden çalıştırılabilecek uygulamalar geliştirmek için kullanılır.

Visual BASIC Enterprise Edition Controls:

Standart EXE görüntüsünden farklı olarak, kullanmakta olduğumuz sürüm Enterprise Edition sürümü olduğu için bazı ekstra kontrol bileşenlerini de araç çubuğumuza eklemektedir.

Visual Basic ile yapılabilecekler kısaca yukarıda belirtildiği gibidir.

Bir projeye başlamadan önce, projemizin bu yukarıda saymış olduğumuz türlerden, hangisine daha yakın olduğunu tespit etmek gerekir. Bunun yanı sıra, bu zor bir karar gibi gözüktüğü için, belirtmekte fayda var ki; projeler genellikle Standard EXE türünde hazırlanmaktadır.

3.2. Proje Ekranı

Proje türlerinden Standart EXE seçildikten sonra karşımıza gelen proje ekranı aşağıda gösterilmektedir.



Şekil 3.2: Visual Basic proje ekranı

Bu ekran görüntüsü ve üzerindeki bileşenleri daha önceki bölümlerde, ayrıntılı bir şekilde tanıtmıştık.

Her birisi ayrı ayrı öneme sahip olan bu bileşenleri, projemiz üzerinde yapacağımız çalışmalarda sık sık kullanma ihtiyacı duyacaksınız.

Burada dikkatinizi çekmek istediğim nokta; başlık çubuklarında ve Project Explorer Penceresinde, projemizin adının "*Project1*" olarak geçiyor olması. Projemize şu anda bir isim vermediğimiz için Visual Basic projemize bu adı otomatik olarak verir, eğer bu projeyi kapatmadan File(Dosya) menüsünden New Project komutunu tıklarsanız, yeni açılacak olan projenin adı da "*Project2*" olacaktır.

Projemizi kaydetmek için File menüsünden *Save* Project komutunu veririz. Visual Basic bize önce proje içerisinde kullanılan bileşenlerin isimlerini sorup, kaydetmek ister. Bunlar varsa modül, ardından formlardır. Her bileşene ayrı ayrı o anda isim verebileceğimiz gibi, tasarım aşamasında vermiş olduğumuz isimler de burada getirilebilir. Bileşenleri ve Projeyi, projenizi tanıtacak bir isim vererek oluşturmuş olduğunuz bir klasörün altına kaydetmeniz, ileride projelerinize kolaylıkla ulaşmanız adına doğru bir davranış olacaktır.

3.3.İlk Visual Basıc Projemiz

3.3.1. Proje Konusunun Belirlenmesi

Ilk önce projemiz için bir konu tespit etmeliyiz. Projemizin içerisinde, özellikleriyle tanıdığımız etiket, komut düğmesi ve metin kutularının yer alması gerektiğini hatırlayalım.

İlk projemizin konusunu bir hesap makinesi olarak belirledik.

3.3.2. Kullanılacak Nesnelerin Tespiti

Simdi düşünelim, hesap makinesi neler yapabilecek? Simdilik hesap makinemizin dört işlemi gerçekleştirmesini yeterli görüyorum.Hatta girilen iki sayıyı işleme sokmasını ve bu iki sayı arasında çıkartma, toplama, çarpma ve bölme işlemleri yapmasını istiyorum.

Evet simdi de ihtiyacımız olan ekran görüntüsünü bir düsünelim.

- \triangleright Sayıları girebilmemiz için iki adet metin kutusu
- \geq Sonucu görebilmemiz için bir adet metin kutusu ya da etiket
- \geq İşlemi seçmek için dört adet buton(toplama, çıkartma, çarpma, bölme butonları)
- \triangleright Sayıları nereye gireceğimizi ve sonucu nereden alacağımızı anlatan Etiketler.

3.3.3. Form Yüzeyine Kontrollerin Yerleştirilmesi

Form tasarımını herkes kendine uygun bir tarzda gerçekleştirebilir. Ancak ekranda oluşacak olan görüntünün kullanıcılar için çok önemli olduğunu unutmayın. Çünkü kullanıcılar kodları ve programın nasıl çalıştığını bilmezler, onları tek ilgilendiren, programın doğru çalışması ve kolay kullanılmasıdır. İste bu noktada tasarımcı, tasarımını sadece kendisi için değil, herkes için yaptığını asla unutmamalıdır. Her zaman, en kolay kullanım ve en çabuk anlaşılabilme ilkeleriyle programlarını hazırlamalıdır.

Gelelim hesap makinemize, dilerseniz ekran tasarımını oluşturalım, ben kontrolleri aşağıdaki şekilde şimdilik rastgele yerleştirdim. Sizler de form üzerine gerekli nesneleri verlestiriniz.

																			C	1												_		_		
C	1	ų	F	DI	ņ	hi	ļ										÷	÷	÷														1	Ę	1	-
	L	a	b	el	1	*	* *	1		-	÷ Te	ex	d 1	÷ R	4	4	÷	÷	4	1						Ī	1 an an an an an an an an an an an an an	C	on	h	nai	nc	; 11	¥		~~~~
	L	a	Ь	el.	2									[T	ex	t2	2					2.1	49	and the second second	Sector A	Ţ.	24	Ce	on	m	ar	nd	12		1
	L	a	Ь	eli	3										*	÷ Te	÷ ея	i) dG	9 3	ŷ	ŷ	ÿ.	ÿ.	ÿ.	÷.			0000	C	on	m	a	nd	13	A SAL	
				いまいまいまい	いまいまいまい	いまいまいまい		いまいまいまい	Contractory of		Contractory of		Contractory of	Contraction of													i I	* 24	; Co	on	nino	i ar	nd	÷ 4	*	1
																	Contraction of the local distance of the loc	Constant of the other of the other o	Constant of the other states													こうこう さいきいきい		Contractor (e.)	いきいきいきいきいきい	いたいたいたいたいたい

Şekil 3.3: Hesap makinesi form tasarım ekranı

Şimdi de bu nesnelerin düzenlenmesine; öncelikle form üzerine doğru bir şekilde alt alta, yan yana yerleştirilmesine ve elbette aynı boyutlara getirilmesine çalışalım.

Önce etiketlerden başlayalım. En üstte yer alan Label1 isimli etiketin boyutlarını mouse yardımıyla ayarlayın. Daha sonra durmasını istediğiniz konumu seçin ve fare yardımıyla yerleştirin.

🖷 Project1 - Forr	n1 (Form
💐 Form1	
_abel1	ext1
Label2 975	<mark>x 255</mark> ext2

Şekil 3.4: Label1 boyut

Ben boyutunu 975x255 şeklinde ayarladım. (bk.Şekil 3.4)

Konumunu da mouse ile belirledim, ayrıca kontrol ettiğimde Properties Penceresinden Left özelliğinin yani formun sol başından içeriye doğru olan mesafesinin 120, Top özelliğinin yani formun yine sol üst başından aşağıya doğru olan mesafesinin 240 olduğunu gördüm.(bk.Şekil 3.5)

Ardından Ctrl tuşuna basılı tutarak sırasıyla Label3, Label2 ve Label1'i tıklayarak seçtim. (bk.Şekil 3.6)

Format (Biçim) menüsünden Make Same Size (aynı boyuta getir) alt menüsüne ve oradan da Both (her ikisi de) komutuna ulaşarak tüm Etiketleri aynı boyuta getirilir.(bk.Şekil 3.7)

Etiketlerin formun sol yanına olan mesafesinin eşit olmadığını düşünüyorsanız, Format menüsünden Align (yaslama) alt menüsüne girip Lefts (sol taraflar) komutuyla aynı hizaya getirebilirsiniz.

Etiketlerin arasındaki mesafenin eşit olmadığını da düşünebilirsiniz. Bu durumda da yine etiketlerin üçü de seçiliyken Format menüsünden Vertical Spacing (Dikey Mesafeler) 'e ve oradan da Make Equal (eşit yap) komutuna ulaşarak, bu mesafeyi de ayarlayabilirsiniz.

Properties - La	bel1 🛛 🗵								
	•								
Alphabetic Cat	egorized								
Alignment	0 - Left Justify								
Appearance	1 - 3D								
AutoSize	False								
BackColor	8H8000000								
BackStyle	1 - Opaque								
BorderStyle	0 - None								
Caption									
DataField									
DataMember									
DataSource									
DragMode	0 - Manual								
Enabled	True								
Font									
ForeColor	8H8000001								
Height	255								
Left	120								
LinkItem									
LinkMode	0 - None								
LinkTimeout	50								
LinkTopic									
MousePointer	0 - Default								
OLEDropMode	0 - None								
RightToLeft	False								
Tag									
ToolTipText									
Тор									
UseMnemonic	True								
Visible	True								
WhatsThisHelpI	0								
Width	975								
WordWrap	False								
Top Returns/sets the between the inter	Top Returns/sets the distance between the internal top edge of								

Şekil 3.5: Label1 konum

Project1 - Form1 (f
Form1
abel?
abel2
abel3

Şekil 3.6: Tüm etiketler seçilir.

ect	Format Debug Run	Qu	ery Diagra	am <u>T</u> oo
Z	Align		ha	i ai
	<u>M</u> ake Same Size	•	🛱 Width	
	🛱 Size to Gri <u>d</u>		Height	
Pra	Horizontal Spacing			-
	⊻ertical Spacing	•		
	<u>C</u> enter in Form	•		:::::
ab -	Order	٠		Comm
Bap	Lock Controls		:	Com
Babe	43	۰۰ t3	•••••	Com

Şekil 3.7: Etiketler aynı boyuta getirilir.

Bu şekilde Format menüsündeki komutların bir kısmını kullanmayı öğrendiniz. Formunuzda yer alan tüm nesneleri bu komutları kullanarak düzenleyiniz ve altta görülen ekran görüntüsünü elde ediniz.

💐 Form1			
Label1	Text1	Command1	
Labal2	-	Command2	
Labeiz	Text2	Command3	
Label3	Text3	Command4	

Şekil 3.8: Düzenlenmiş kontroller

Şimdi sıra geldi kontrollerimizin özelliklerinin ya da Properties pencerelerinin ayarlanmasına. Öncelikle tüm nesnelerin ve tabii ki formumuzun Caption özelliklerini (metin kutularında Caption özelliği yerine Text özelliği vardır) değiştirip uygun başlıklar yazalım ve aşağıdaki şekli elde edelim.

💐 Hesap Ma	kinesi		<u> </u>
1.SAYI		TOPLAMA	
2 6 4 1	-	ÇIKARTMA	
2.5811	1	ÇARPMA	
SONUÇ		BÖLME	

Şekil 3.9: Başlıklar girildikten sonra

Sonuç kutusuna hiçbir değer girilmeyeceği için Enabled özelliğini False yapalım. F5 tuşuna basıp çalıştırırsanız Text3'e değer giremediğinizi göreceksiniz.

İsterseniz bir önceki bölümde öğrenmiş olduğunuz yazı biçimi, renk verme, görünüm gibi özellikleri değiştirerek farklı form tasarımları yapabilirsiniz. Ancak ben bu kadarını yeterli görüyorum ve kod yazım aşamasına geçiyorum.

3.3.4. Kontrollerin Olaya Yönelik Kodlarının Yazımı

Programımızın kullanıcı arayüzü tamamlandıktan sonra, kullanıcının görmediği ancak programımızın temel taşı olan, onun çalışmasını sağlayan kodların yazımına geçebiliriz.

Visual Basic'in olay yönelimli (*Event Driven*) bir programlama dili olduğunu söylemiştik. Olay güdümlünün anlamı tasarımcının uygulamaların kontrolünü elinde tutmasıdır. Tasarımcı fare ile bir nesneye tıklandığında ya da klavyeden bir tuşa basıldığında,

hangi olayların yürütüleceğine karar verir. Visual Basic programınızda kullandığınız nesneler için, yapılacak olaya yönelik yazmış olduğunuz kodları destekler.

İşte bu noktada her bir nesneye yönelik olayı ya da olayları düşünmemiz gerekir.

Bizim projemizde yani hesap makinesinde biz sayıları gireriz ancak işi yapan butonlardır.Toplama butonunu yani Command1'i ele alalım;Toplama butonu tıklandığında, Program Text1'in içindeki yazıyı alacak, Text2'nin içindeki yazıyı alacak ve toplayacak.

Uyarı: YAZILAR TOPLANAMAZ!!

Yeniden düşünelim;

Toplama butonuna tıklandığında, Program Text1'in içindeki yazıyı alıp sayıya dönüştürecek, Text2'nin içindeki yazıyı alıp sayıya dönüştürecek ve bu iki sayıyı toplayacaktır.

Visual Basic' de metini sayıya dönüştüren komut VAL' dir. Value, yani değerden gelmektedir.

"Metin kutusuna yazdığımız zaten bir rakam değil miydi?" dediğinizi duyar gibiyim, ancak unutmayın, Visual Basic derleyicisi (yorumlayıcısı) metin kutusundaki değerleri daima metin olarak algılar. Eğer bu değer bir sayı ise ve işleme sokulacaksa derleyiciyi gerekli dönüşümü yapması için uyarmamız gerekir. Ve bu işi de *Val(Text1.Text)* diyerek yaparız.

Peki, bu kodu nasıl gireceğiz?

Gayet kolay form tasarım ekranınızda toplama butonunuzun üzerini çift tıklayın. Açılan kod ekranı aşağıdaki şekilde olacaktır.

Command1 🛛 👻	Click	*
Private Sub Command1_Click()	
End Sub		

Şekil 3.10: Toplama butonu çift tıklanınca açılan kod penceresi

Command1 nesneyi, Click ise olayı göstermektedir. Private Sub ile başlayan ve End Sub ile biten alan ise sizin bu nesne için yazacağınız prosedür ya alt program alanıdır. Click olayı Command Button için varsayılan olarak gelmektedir. Çünkü sıklıkla bu olay kullanılmaktadır. Değiştirmek için sağ üstteki olay kutusundan farklı bir olay seçip farklı bir prosedür oluşturabilirsiniz.

Şimdi bu olay olduğunda yapılacak olan işi yazalım.



Gördüğünüz gibi kod yazmanın konuşmaktan bir farkı yok. Sadece Visual Basic' in dilinden konuşmak gerekiyor.

Şimdi kodumuzu girelim ve Command1_Click Prosedürünü tamamlayalım.

Kod yazarken, nesnenin adını yazıp nokta koyduktan sonra nesnenin özellik ve metodlarının listelendiğini daha önce söylemiştik. Şimdi de kod girerken "*Text3*." dediğinizde bir pencere açılır. Siz yazmaya devam ettikçe en yakın özellik listeden karşınıza getirilir, bu anda boşluk tuşuna ya da *Tab* tuşuna basabilirsiniz, böylece hem yazmaktan hem de yanlış yazmaktan kurtulmuş olursunuz.

Val() komutunun parantezini kapatmayı unutmayın!

🗖 Project1 - Form1 (Code)		
Command1	Click	•
Private Sub Command1_Click() Text3.Text = Val(Text1.Text) End Sub	+ Val(Text2.Text)	

Şekil 3.11: Toplama butonunun prosedürü

F5 tuşuna basarak projenizi çalıştırınız. Text1 ve Text2'ye sayı girdikten sonra TOPLAMA butonuna basınız. Doğru sonucu aldıysanız. ÇIKARTMA, ÇARPMA ve BÖLME butonlarının prosedürlerini de oluşturup tekrar projenizi çalıştırınız.

command4 🗾	Click	
Private Sub Command1_Click() Text3.Text = Val(Text1.Text) End Sub	+ Val(Text2.Text)	
Private Sub Command2_Click() Text3.Text = Val(Text1.Text) End Sub	- Val(Text2.Text)	
Private Sub Command3_Click() Text3.Text = Val(Text1.Text) End Sub	* Val(Text2.Text)	
Private Sub Command4_Click() Text3.Text = Val(Text1.Text) End Sub	/ Val(Text2.Text)	

Şekil 3.12: Tüm butonlar için kodlar girildi

🖷, Hesap Ma	kinesi	_ 🗆 ×
1.SAYI:	66	TOPLAMA
2.SAYI:	22	CIKARTMA
SONUC	3	CARPMA
		(BUTWE)

Programımız tamamlandığında aşağıdaki ekran görüntüsüne sahip olacaktır.

Şekil 3.13: Program tamamlandı

3.3.5. Programın Geliştirilmesi

Programlarınızı tamamladıktan sonra, genellikle bir şeylerin eksik olduğu hissine kapılacaksınızdır. Bu programcıların genel alışkanlıklarından birisi gibidir.

Bu programı tamamladıktan sonra, ne eksik diye düşündüğümde aklıma iki şey geldi;

Yeni bir hesap yapılacağı zaman, metin kutularının içlerinin silinip tekrar yazılması biraz uzun sürüyor, bunun yerine konulacak bir butona basılıp metin kutularının içerisi temizlenebilse iyi olurdu. Programdan çıkmak için sol üst köşedeki, program adının yanında yer alan ikonu tıklayarak açılan menüden ya da sağ üst köşedeki kapat butonundan yararlanıyoruz. Oysa ki form üzerine yerleştirilecek bir çıkış butonu kullanım kolaylığı sağlayabilirdi.

Evet, şimdi bu düzenlemelerin nasıl yapılacağının öğrenelim.

3.3.5.1. Metin Kutularının İçlerinin Temizlenmesi

Formun üzerine yerleştireceğimiz bir buton yardımıyla bu işlemi gerçekleştirmemiz mümkün olacaktır. Ancak ondan önce yazacağımız koda dikkatinizi çekmek istiyorum.

Metin kutusunun içine boşluk yazdırmak ile metin kutusunun içinin temizlenmiş olacağını söyleyebiliriz. Bunun komutu ise:

Text1.Text= "" şeklindedir.

Formumuzun üzerine yerleştireceğimiz "TEMİZLE" başlıklı ismi ise Command5 olan butonumuzun üzerini çift tıklayıp, açılan kod ekranında Command5'in Click olayına aşağıdaki kodları yazalım;

```
Private Sub Command5_Click()
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
End Sub
```

Gördüğünüz gibi oldukça basit birkod yazımıyla bu iyileştirme çalışmasını başarmış olduk.

💐 Hesap Ma	kinesi	<u>_ ×</u>
1.SAYI:	77	TOPLAMA
2.SAYI:	11	CIKARTMA
SONUC	7	CARPMA
TEMIZLE		BULME

Şekil 3.14: Temizle butonu eklendi. 3.3.5.2. Programa Çıkış Butonu Eklenmesi

Uygulamalarınızda çok sık kullanacağınız bu buton için form yüzeyinize bir buton yerleştirin. Command6 adını alacak bu butonun başlığını "ÇIKIŞ" olarak değiştirin. Ardından üzerini çift tıklayıp aşağıdaki kodu ekleyin.

Private Sub Command6_Click() End End Sub

Sadece bir "End" ifadesi çıkış butonu için yeterli olan koddur.

Böylece bu iyileştirme de gerçekleştirildi ve programımız yandaki görüntüyü aldı ve çalışmaya hazır hale geldi.

🗃. Hesap Mak	inesi	<u>_ 🗆 ×</u>
1.SAYI:	33	TOPLAMA
2.SAYI:	3	CIKARTMA
SONUC	11	CARPMA
TEMIZLE	CIKIS	

Şekil 3.15: Çıkış butonu eklendi.

UYGULAMA FAALİYETİ

Aşağıdaki işlem basamaklarına göre uygulama faaliyetini yapınız.

İşlem Basamakları	Öneriler
Yeni bir Visual Basic Projesi oluşturup, adını İlkprojem olarak belirleyin.	Projenizi açılışta Standard EXE olarak tespit ediniz.
Projenize istediğiniz adette komut düğmeleri, etiketler ve metin kutuları yerleştirip bunların ekran üzerinde hizalanmalarını temin edin.	 Projenizi açılış ekranında sorulan klasör adresine kaydediniz, ayrı bir klasörde tutmanız tavsiye edilir. Format menüsünü kullanınız.
Ekrana yerleştirmiş olduğunuz nesneleri kullanarak bir öğrenciye ait, üç yazılı ve iki sözlü notunun girilebileceği, ve ders ortalamasının ekranda gösterileceği bir form ekranı tasarlayınız.	Tasarım sırasında özgür düşününüz, ve kullanıcı için en kolay giriş ekranını tasarlamaya gayret ediniz.
 Gereksiz olan nesneleri siliniz. 	Kod ekranına geçiş için nesnenin üzerini çift tıklamanız yeterlidir. Ya da Project Windows'u kullanabilirsiniz.
Programınızda öğrenci notlarını, üzerine tıklandığında hesaplayacak olan buton için gerekli olan kodları yazınız.	F5 tuşunu ya da Project menüsünden RUN komutunu kullanınız.
Programınızı çalıştırıp deneyiniz.	• Hatalı satırlar işaret edilmektedir.
Hatalarınız var ise düzeltiniz.	Hatanızı bulamazsanız, üstte anlatılan uygulama örneklerini inceleyiniz.
Programınıza "TEMİZLE" ve "ÇIKIŞ" butonları ekleyiniz.	

ÖLCME VE DEĞERLENDİRME

Aşağıdaki soruları cevaplayarak bu faaliyette kazandığınız bilgileri ölçünüz.

OBJEKTIF TEST (ÖLCME SORULARI)

- 1. Aşağıdakilerden hangisi Visual Basic ile oluşturabileceğiniz proje türlerinden birisi değildir?
 - A) DHTML Application
 - B) ISS Application

C) Document Project D) Standart EXE

D) Align

- 2. Make Same Size komutu hangi menüde yer alır? A) Tools C) Project B) View D Format
- 3. Alt alta yerleştirilmiş olan nesnelerin arasındaki mesafenin eşit olmadığını gördüğünüzde, hangi komut yardımıyla bunları düzeltme yoluna gidersiniz? C) Vertical Spacing
 - A) Horizantal Spacing
 - B) Make Same Size
- 4. Programın çalışması esnasında herhangi bir nesneye müdahale edilememesini istersek, hangi nesne özelliğini "False" yapmamız gerekir?
 - A) Visible C) Enabled B) TabIndex D) Locked
- 5. Programın çalışması esnasında asıl çalışmayı gerçekleştiren, programın temel yapı taşı olan ancak kullanıcıların göremedikleri satırlara ne ad verilir?
 - A) Program kodları C)Akıs diyagramı B) Program adımları D) Algoritma
- 6. Tasarımcının uygulamaların kontrolünü elinde tutması, fare ile bir nesneye tıklandığında ya da klavyeden bir tuşa basıldığında, hangi olayların yürütüleceğine karar vermesine Visual Basic'in hangi özelliği karşılık gelmektedir?
 - A) Nesneye yönelik olması
 - B) Windows tabanlı uygulamaları desteklemesi
 - C) Kolay bir arayüze sahip olması
 - D) Olay yönelimli ya da olay güdümlü olması
- 7. Metin kutularının içerisindeki değerlerin rakam olarak algılanmasını sağlayan komut hangisidir?

A) Val()	C) Rakam()
B) Sayı()	D) Num()

8. Visual Basic' de hazırladığımız projemizi çalıştırmak için hangi fonksiyon tuşunu kullanırız?

A) F11	C) F7
B) F5	D) F4

9. Metin kutularının içerisinin temizlenmesi için aşağıdaki verilmiş olan komut satırlarından hangisi kullanılmalıdır?

A) Text1.Text= ""	C) Text1.Text=0
B) Text1.Text= ""	D) Text1.Text= "NULL"

10. Programa çıkış butonu eklendikten sonra, bu butonun kod sayfasına hangi komut girilmelidir?

A) !End! B) End C) "END" D) Unload Form

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete geri dönerek tekrar inceleyiniz.
ÖĞRENME FAALİYETİ-4

AMAÇ

Görsel programlamada kurallara uygun olarak proje derleyebileceksiniz.

ARAŞTIRMA

Programın derlenmesi ne demektir? Araştırınız.

Hata mesajları neden ortaya çıkmaktadır, araştırınız.

Çalıştırılabilir dosyalar hakkında bilgi toplayınız. Bu dosyaların uzantıları nelerdir, öğreniniz.

4. VISUAL BASIC HATA DENETLEME ARAÇLARI

Programlarda meydana gelen hatalara *bug* adı verilir. Bu *bug*'ların bulunmasına ve düzeltilmesine ise *debugging* denilir. Bizde burada bug'a hata ve *debugging*'e de hata denetimi adını vereceğiz. Mantıksal hataları bulmak için birkaç farklı yöntem kullanabilirsiniz.

Bunlardan birincisi *desk-checking* ya da adımlama yöntemidir. Bunun için elinize bir boş kağıt alır, değişken isimlerinizi yan yana yazar ve programı kafanızdan adım adım çalıştırırsınız. Her adımda değişkenlerin altına değişen değerlerini yazar ve hata olup olmadığını izlersiniz.

İkincisi ise *Print* ya da yazdırma metodudur. Bu metodu kullanmak için ise programınızda değişken değerlerinin değiştiği satırlara birer *Print* komutu koyarak, ekranda programın her adımında değişkenlerin aldıkları değerleri izleyebilirsiniz.

Visual Basic hata denetleyicisi ise her iki metodunda kullanımına imkân sağlayacak şekilde tasarlanmıştır.

Programın çalışması esnasında programı duraklatıp o anki değişken değerlerini okutabileceğiniz gibi, programı adım adım çalıştırıp her adımda ortaya çıkan değişiklikleri de izleyebilirsiniz. Bu gözlemleri yapmanızı sağlayacak üç yardımcı pencere bulunmaktadır. Bunlar, *Immediate Window, Watch Window* ve *Locals Window*' dur. Bunlara hata giderme (*Debug*) pencereleri denilir.

4.1. Program Modları

Visual Basic Programınız üç modda bulunabilir. Bunlar tasarım (*Design*) modu, çalışma (*Run*) modu ve durdurma (*Break*) modudur. Her üç modda Visual BASIC başlık çubuğundan anlaşılabilir.

, Project1 - Microsoft Visual Basic [design]							
File Edit View Project Format Debug Run Query Diagram Iools Add-Ins Window Help							
😼 - 🐂 - 👕 🚅 🖶 🕺 ங 🛍 🛤 🕫 🕫 🕨 🔳 😻 🗃 🔒 🦉 🛠 🔁 🔔 📩 📩 2040, 240 🛛 🖞 480 x 195							

Şekil 4.1: Tasarım durumunda iken başlık çubuğu

i	, Projec	t1 - M	licrosoft	Visual E	asic [b	reak]							_ 🗆 🗵
Name of	Eile Edit	⊻iew	Project	Format	Debug	Run	Query	Diagram	<u>T</u> ool	s A	dd-Ins <u>W</u> indow <u>H</u> elp		
	100-1	4 , • °	1			3 #	5	C×)	н		¥87878	Ln 8, Col 1	

Şekil 4.2: Durdurma durumunda iken başlık çubuğu

'n,	Pro	oject	1 - 1	1icroso	oft V	isual E	Basic [run]									
Eil	e E	dit	⊻iew	Proje	t F	ormat	Debu	g <u>R</u> un	Query	Dijagra	m <u>T</u> o	ols	Add-Ins	Window	Help		
	3	• 12	y - "		-	3		R #	5	CH	► 1			日日	s 🛠 🖥	Ln 8, Col 1	

Şekil 4.3: Çalışma durumunda iken başlık çubuğu

Tasarım modunda formunuzun üzerine kontroller yerleştirir, özelliklerini ayarlar, gerekli kodlarını yazarsınız. Çalışma modu (Run Time), başlama (Start) butonuna basıldığı anda başlamış olur. Durdurma modu ise bir çalışma zamanı hatası (run-time error) oluştuğu anda otomatik olarak ortaya çıkar. Bununla birlikte, programın çalışması esnasında isterseniz siz de Ctrl+Break tuşlarına birlikte basarak, Run menüsünden Break komutunu tıklayarak ya da araç çubuğundan Break butonuna II tıklayarak durdurma moduna geçiş yapabilirsiniz. Program durdurma modunda iken, Immediate Window'u kullanarak değişkenlerin o anki durumlarını görüp gözlemleyebilir, değiştirebilir ya da nesnelerin özelliklerini ayarlayabilirsiniz. durdurma moduna geçtiğiniz anda araç çubuğu üzerinde yer alan başlatma (Start) butonunun adının sürdür (Continue) butonu olarak değiştiğini görebilirsiniz. Ona tıklayarak programınızı kaldığı yerden devam ettirebilirsiniz.

4.1.2. Immediate Window (Anlık Durum Penceresi)

Anlık durum penceresini görüntülemek için CTRL+G tuşlarına birlikte basmanız ya da View menüsünden Immediate Window komutunu tıklamanız gerekir. Bu pencere tasarım modunda da kullanılabilmesine karşın genellikle durdurma modunda kullanılmaktadır.

Bu pencerenin içerisinde yer alan editöre benzer alana bir ifade yazar ve Enter'a basarsanız ifadede yer alan eğer bir komut ise, ya da bir hesaplama ise bir kereliğine çalıştırılır.

Formda yer alan,

Print ifade

Şeklindeki bir kullanımda ifadenin sahip olduğu değer görüntülenir. Eğer program çalışıyorsa var olan değişkenlerin değerleri de bu şekilde görüntülenebilir. Ancak değişkenin sürekli değişen değerleri ardı ardına değil, az önce yukarıda da söylediğimiz gibi sadece bir kez o anki değeriyle görüntülenir. Bu sebeple Anlık durum penceresi (*Immediate Window*)'ni durdurma modu (*Break Mode*) 'nda kullanmak daha doğrudur.

Aşağıdaki şekilde üç farklı ifade anlık durum penceresinde görüntülenmektedir.



Şekil 4.4: Anlık durum penceresi

Not:Bu pencerede yer alan satırların üzerine gidip Enter'a basarak satırda yer alan ifadeyi tekrar çalıştırabilirsiniz.Visual BASIC'in eski versiyonlarında Immediate Window, Debug Window adını almıştır.

4.1.3. Watch Window (İzleme Penceresi)

Programınızı tasarlarken ifadelerinizi iki türlü oluşturabilirsiniz; izleme ifadesi ya da durdurma ifadesi. Durdurma ifadelerinin de iki türü vardır; doğru olduğunda programı durduranlar ve değeri değiştiğinde programı durduranlar. İfade her nasıl olursa olsun izleme penceresi (*Watch Window*) ile takip edilebilir.

İzleme penceresi, izleme ve durdurma ifadelerinde meydana gelen değişiklikleri her an, doğru olarak görüntüler. Aşağıdaki şekilde bu ifade türlerinin izleme penceresi ile takibini görüyorsunuz.

Expression		Value Type		Context	
12	ad	0	Integer	Form1.CmdButton_Click	
ď	num	8	Variant/Double	Form1.CmdButton_Click	
È.	say	1	Single	Form1.CmdButton_Click	

Şekil 4.5: İzleme penceresi (Watch Window)

Sembol	Açıklaması
ಕ್ಷಣ	İzleme İfadesi
個	İfade doğru olduğunda dur
4 <u>1</u> 2	İfade değiştiği anda dur

Tablo 4.1: İzleme sembolleri

İzleme penceresine bir ifade eklemenin en kolay yolu, kod penceresinde yazdığınız değişken isminin üzerinde sağ tıklama yapmak ve "Add Watch" komutunu vermektir. Bunun sayesinde açılan Add Watch diyalog penceresinden üç farklı izleme tipinden birisini seçebilirsiniz. İzlemeyi kaldırmak için ifadenin üzerini sağ tıklayıp "Delete Watch" komutunu vermeniz, izlemenin tipini değiştirmek için ise yine ifadenin üzerinde sağ tıklayıp "Edit Watch" demeniz gerekir.

4.1.4. Locals Window (Değişken Penceresi)

View menüsünden "Locals Window" butonu ile ulaşılan değişken penceresi, Visual BASIC 5.0 'la birlikte gelen bir özelliktir. Bu pencere otomatik olarak programınızın çalışmakta olan prosedürü içerisinde yer alan değişkenlerin adlarını, değerlerini ve tiplerini gösterir. Bu ekrandan programın çalışması esnasında gösterilen değişkenlerin değerlerini değiştirebilir ve ayrıca kullanmış olduğunuz kontrollerin özelliklerini inceleyip, değişiklikler yapabilirsiniz.

Locals					
CmdButton_Click		÷			
Value	Туре				
	Form1/Form1				
10	Variant/Double				
1	Single				
"mustafa"	String				
	CmdButton_Click Value 10 1 "mustafa"	CmdButton_Click Value Type Form1/Form1 10 Variant/Double 1 Single "mustafa" String			

Şekil 4.6: Değişken penceresi (Locals Window)

4.1.5. Programı Adımlamak

Programlarımız çalıştırılırken, derleyici tarafından program içerisindeki dallanmalara uygun olarak satır satır çalıştırılır. Her bir satırda bir işlem yerine getirilir ve bir işleme başlanmadan önce ondan önceki ifadenin mutlak yerine getirilmesi gerekmektedir. Biz de eğer istersek Visual BASIC'in bize verdiği imkanlar sayesinde bu çalışmayı izleyebiliriz. Buna adımlama (stepping ya da step into) denilmektedir. Her adımda değişkenlerin değerleri, ifadeler, koşullar hata ayıklama pencereleri ile gözlenebilir ve değişken değerleri istenilirse değiştirilebilir.

Adımlama kullanıcı tarafından girilen ilk prosedürün ilk olayı ile başlar. Eğer kullanıcı tarafından bir sonraki adımda başka bir alana yönlendirme yapılmamış ise program satır satır adımlanacaktır.

Başka bir adımlama tekniği de, breakpoints yani durdurma-noktası'dır. Kullanıcı eğer isterse programın belli satırlarına durdurma noktaları koyabilir. Bunlar sayesinde programı adımlamaya başladığımızda program ilk durdurma noktasına kadar çalışır ve durur, ardından tekrar adımladığımızda ise ikinci durdurma noktasına kadar çalışır ve durur. Böylece kontrollü bir şekilde istediğimiz noktalarda gerekli kontrollerimizi yapıp, programımızı denetleyebiliriz.

Bu denetimleri yapabilmek için kullanacağımız kısayol tuşları aşağıda listelenmiştir. Bu tuşları kullanmak yerine Debug menüsünden de aynı işlemleri gerçekleştirmek mümkündür.

\triangleright	Step Into:		F8
\triangleright	Step Over:		Shift+F8
\triangleright	Step Out:		Ctrl+Shift+F8
\triangleright	Breakpoint oluşturmak:	O satıra gidip	F9 a basınız
\triangleright	Breakpoint'i kaldırmak:	Breakpoint koyduğunuz satıra gidip	F9'a basınız.
\triangleright	Tüm Breakpoint'leri tem	izlemek:	Ctrl+Shift+F9
\triangleright	Bir sonraki ifadeye geçiş	:	Ctrl+F9

	Bir sonraki breakpoint'e ya da programın sonuna gidiş:	F5
\triangleright	Programın çalışması sırasında kursöre gidiş için:	Ctrl+F8

Örnekler:

- Yeni bir proje hazırlamak üzere Visual BASIC'i çalıştırınız.
- Projenizde yer alan formunuzun üzerine, bir komut butonu ve bir etiket nesnesi yerleştiriniz.
- Formunuzu bu iki nesnenin boyutlarına yetecek oranda küçültünüz.
- Komut butonunuza *CmdButton*, etiket nesnesine ise *Lbl1* adını veriniz.
- CmdButton'un başlık özelliğini BASLAT olarak, Lbl1'in de başlık özelliğini GÖSTERGE olarak değiştirin.
- Komut butonunun üzerini çift tıklayıp aşağıdaki prosedürü yazın.

```
Private Sub CmdButton_Click()
Dim num As Single
Lbl1.Caption = ""
num = Val(InputBox("bir sayi giriniz:"))
num = num + 1
num = num + 2
Lbl1.Caption = num
End Sub
```

Not : inputbox nesnesi veri girişi yapmak amacı ile kullanılır. Aşağıdaki şekildeki gibi bir uyarı ekranına benzer şekilde pencerenin açılmasıyla dışarıdan veri girilmesine imkan sağlar.

En basit kullanımı ;Değişken=InputBox("Pencere Başlığı") şeklindedir. Daha ayrıntılı kullanım için MSDN kütüphanesine başvurabilirsiniz.

X
OK
Cancel
Cancel

Girilen değer değişkene aktarılır.

Yazmış olduğunuz programda yer alan num değişkeninin üzerinde mouse ile sağ tıklama yapıp açılan menüden "Add Watch" komutunu veriniz.

🖉 Project 1	- Form1 (Code)	-IOX
CrndButto	n 🔽 Click	-
Priv End	ate Sub CmdButton_Click() Dim num As Single Lbl1.Caption = "" num = Val(InoutBox("bir sayi giriniz:")) nu % Cot num = Copy Lb Paste Su E List Properties/Methods List Constants Quick Info Parameter Info At Complete Word Toggle + Su Qbject Browser	
	Add Watch Definition Last Position Hide	

Şekil 4.7: Add Watch komutu

Açılan ekranda OK butonuna basınız.

xpression:		ОК
ium		Cancel
Context Procedure:	CmdButton_Click	Help
Module:	Form1	
Project:	Project1	
Watch Type	1	
 Watch E 	xpression	
C Break W	hen Value Is True	
C Break W	hen Value Changes	

Şekil 4.8: Add Watch penceresi

Wat	ches			×
Exp	ression	Value	Туре	Context 🔺
66	num	<out context="" of=""></out>	Empty	Form1.CmdButton_Cl

Şekil 4.9: Add Watch ekranı

Açılan ekranda num değişkeninin izleme penceresine yerleştirildiğini göreceksiniz.

- View menüsünden Locals Window penceresini açmak için gereken komutu tıklayın.
- Programı adım adım çalıştırmak için, F8'e basınız, ardından BASLA butonuna tıklayıp tekrar F8'e basınız. Sarı renk almış olan izleme satırının Lbl1.Caption= "" ifadesinde olduğunu, yani bir sonraki adımda bu işin yapılacağını belirttiğini göreceksiniz. Bu arada Watch Window ve Locals Pencereleri de aşağıdaki şekli alacaktır.

CmdButton Click		•	
₽	rivate Sub CmdBu Dim num As Si Lbl1.Caption num = Val(Inp num = num + 1 num = num + 2 Lbl1.Caption	utton_Click() .ngle = "" utBox("bir sayi giri - - = num	.niz:"))
E	nd Sub		1

Şekil 4.10: Programın adımlanması

Single	Form1.CmdButton_Cl
	Single

Şekil 4.11: Watch penceresi

Locals			×
Project1.Form1.Cm	dButton_Click		
Expression	Value	Type	
🛨 Me		Form1/Form1	
num	Ŭ,	Single	-

Şekil 4.12. Locals Penceresi

- F8'e basınız. Lbl1.Caption= "" satırının işletildiğini ve (ilk kez kullanımına tanık olduğunuz ama ilerleyen modüllerde sıkça kullanacağınız) "InputBox" (Giriş Kutusu) nesnesinin çağırıldığı satıra gidildiğini göreceksiniz.
- F8'e tekrar basınız, InputBox nesnesindeki yazım alanına 4 rakamını giriniz ve "OK" butonuna basınız. Watch ve Locals pencerelerinin de aşağıdaki görüntüyü aldığını göreceksiniz.

ОК
Cancel

Şekil 4.13: Veri giriş kutusu (InputBox)

Watches				×
Expression	Value	Туре	Context	
6 ർ num	4	Single	Form1.CmdButtor	n_Cl
		1.000		
				-

Şekil 4.14: Watch penceresi



Şekil 4.15: Locals penceresi

- "num = num + 1" satırını çalıştırmak için tekrar F8'e basınız.
- Mouse ile kod ekranınızdaki "num" ifadelerinin ya da değişken isimlerinin herhangi birinin üzerine gidip birkaç saniye imleci orada tutunuz. Değişkenin o anki değeri küçük bir kutu içerisinde görüntülenecektir.

Сп	ndButt	ton	- Click	
₽	Priv	vate Sub CmdE Dim num As S Lbl1.Caption num = Val(In num = num + num = num + Lbl1.num = 5pn	<pre>sutton_Click() single a = "" sputBox("bir sayi (1 2 a = num</pre>	giriniz:"))
	End	Sub		-

Şekil 4.16: Değişken değerlerinin mouse ile görüntülenmesi

- Araç çubuğundaki End butonuna basarak programı sonlandırın.
- Kursörünüzü "num = num + 2" yazan satıra götürüp F9 tuşuna basınız. Satırın yanında kırmızı bir noktanın oluştuğunu ve satırda yer alan ifadenin üzerinin kırmızıyla çizilip, yazı renginin beyaza dönüştüğünü göreceksiniz. Bunun anlamı bu satıra bir durdurma noktası ya da Break Point konulmuş olmasıdır.
- F5 tuşuna basıp, programı çalıştırın. Veri Kutusu'na 4 girin. Program ilk üç satırı işletecek ve üçüncü satırdaki BreakPoint'de duracaktır. Durdurma noktasının olduğu satır işletilmeyecektir.
- Ctrl + G' ye basarak ya da View menüsünden Immediate Window'u açınız. Eğer daha önceden açılmış ve içerisinde yazılmış satırlar mevcut ise silerek temizleyiniz. Aşağıdaki satırı yazınız. Print "num=";num
- Enter'a basınız.

mmediate	2
Print "num=";num num= 5	
	•

Şekil 4.17. Immediate Penceresi

- num=5 ifadesi breakpoint koyduğumuz satırın henüz işletilmediğini göstermektedir. Watch ve Locals Pencerelerinde de değişkeninizin değerini takip ediniz.
- ▶ F7'ye basarak kod penceresine geri dönünüz.
- Kursörünüzü num=num+1 yazan satıra götürünüz ve Ctrl+F9'a basınız. Bu işlem bu satırın bir sonraki adımda işletilecek satır olduğunu söylemektedir.
- ➢ F8'e basarak işaretli satırı çalıştırınız.
- Ctrl+G'ye basarak Immediate Penceresine dönünüz. Print komutunun olduğu satıra gidip Enter'a basarak bu satırı tekrar işletiniz. num değişkeninin şimdiki değerinin 6 olduğunu göreceksiniz. Watch ve Locals Pencerelerini de inceleyiniz. Ardından kod penceresine geri dönünüz.

Print "num=";num	1
num= 6	
num= 5	

Şekil 4.18: Immediate penceresi

- Kursörü Breakpointin olduğu satıra taşıyınız ve F9 a basarak Breakpoint' i kaldırınız.
- F5'e basarak programınızı çalıştırınız, kaldırdığınız breakpoint satırı da işleme sokulacak ve sonuç 6+2=8 olacaktır. Watch, Locals ve Immediate Pencerelerinde 8 değerini göremezsiniz çünkü program bitmiş ve bu ekranlar devre dışı kalmıştır.

💐 Form1	
Baslat	8

Şekil 4.19: Program sonu

Not:Programınızı adımlarken form ekranınızın görünmez hale geldiğini düşünebilirsiniz, ancak Windows görev çubuğuna bakarsanız, formun adı şeklinde bir butonun orada olduğunu göreceksiniz ve bu butonu tıklayarak istediğiniz an dönüp form ekranını izleyebilirsiniz.

4.2. Projelerin Derlenmesi ve Çalıştırılabilir Dosyaya Dönüştürülmesi

Projenizle ilgili hata denetlemelerini mümkün olduğu kadar detaylı yaptıktan sonra, artık projeniz derlenmeye ve dağıtılmaya hazır hale gelmiş demektir.

Hata denetimini çok iyi yapmak gerekir, çünkü derlenmiş bir programı Visual Basic geliştirme ortamının dışarısında çalıştırılmaktadır, bu nedenle projeniz artık Visual Basic kod penceresinde düzenlenemeyecektir.

Programınızın doğru olarak çalıştığına emin olduktan sonra, File menüsünden Make komutunu veriniz.Visual BASIC "Make Project" diyalog kutusunu açacaktır. Bu pencereden derlenmiş uygulamanızı kaydetmek istediğiniz klasörü seçiniz. Visual BASIC projeye vermiş olduğunuz ismi varsayılan çalıştırılabilir dosya ismi olarak atayacaktır.



Şekil 4.20: Make Project EXE

OK butonuna basmadan önce Options butonunu tıklayarak Project Properties ekranını inceleyelim.(Bu ekrana Project Menüsünden Properties komutuyla da ulaşabilirsiniz.) Bu ekran derlenecek olan uygulamanıza versiyon bilgisi girmenize olanak sağlar. Eğer programınızı ileride geliştirmeyi düşünüyorsanız versiyon bilgisi size yardımcı olacaktır. Versiyon numarası ve tanımlama bilgileri kaynak kodlarla birlikte bulunurlar.

Version Number Major: Minor: Revision: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Application Title: Hesap makinesi Icon: Form1
Type: Value Comments Company Name	n M V
ommond Line Arguments	

Şekil 4.23: Project Properties diyalog ekranı Make sekmesi

Formlarınızdan genellikle birincisini isim ve ikon özelliklerinin alınacağı form olarak tanımlarsınız. Formunuza daha önceden vermiş olduğunuz *Icon* özelliği burada kullanılmaktadır.

Compile sekmesine tıklayarak derleme için yapılabilecek ayarlamaları görebilirsiniz.

Jake	- Project Properties	
0 (0 (0 (0 (0 (0 (0 (0 (0 (0 (Compile to P-Code Compile to P-Code Optimize for East Code Optimize for Small Code Create Symbolic Ng Optimization Advanced Optimizations	ro(tm) Debug Info
)LL Bas	se Address:	

Şekil 4.23: Project Properties diyalog ekranı Compile sekmesi

Derlenmiş olan projenizin olabildiğince hızlı çalışmasını isterseniz Compile Native Code seçeneğini işaretleyiniz. (Eğer P-Code seçeneğini işaretlerseniz, kullanıcınızın sistem klasöründe Visual Basic için gerekli olan DLL dosyalarının bulunmasını isteyecektir. Native code daha hızlı çalışır ve daha az dosyaya ihtiyaç duyar, ama yine de DLL dosyalarının servislerine ihtiyaç duymaktadır.)

Project Properties diyalog kutusunu kapattıktan sonar OK butonuna basınız. Visual Basic kodlarınızı derleyecektir. Hiçbir derleme hatası ortaya çıkmazsa Visual Basic .EXE uzantılı çalıştırılabilir dosyanızı oluşturmuştur (üst sağ köşede derleme durumunu izleyebilirsiniz).

Visual BASIC'den çıkarak Başlat menüsünden Çalıştır seçeneğini tıklayınız. .EXE dosyanızın bulunduğu konumu belirtip çalıştırınız. Formun ikonunun çalıştırma sırasında görev çubuğunda da yer aldığını göreceksiniz.

Böylece programınızı çalıştırılabilir bir dosya haline getirmiş oldunuz, şimdi bu dosyayı bir hafıza birimi ile alıp başka bir arkadaşınızın bilgisayarında çalıştırmayı deneyiniz.

Ayrıca "Package and Deployment Wizard" kullanarak programınızı disket ya da CD yardımıyla kurulup kaldırılabilen bir hale dönüştürebilirsiniz.

Package and Deployment Wizard'ı çalıştırmak için Add-Ins menüsünden Add-In Manager komutunu veriniz. Açılan diyalog penceresinden "Package and Deployment Wizard" ı işaretleyiniz. Ardından altta yer alan Load Behavior kutusundan Loaded/Unloaded ve Load on Startup kutularını işaretleyiniz.

Available Add-Ins	Load Behav	vior 🔺	OK
Component Services Add-In for VB 5.0/6.0	Startup / Lo	aded	
DTC Framework Registrar			Cancel
Microsoft Visio UML Visual Basic Addin	Startup / Lo	aded	
Package and Deployment Wizard	Startup / Lo	aded	
Source Code Control	Startup / Lo	aded	
VB 6 ActiveX Ctrl Interface Wizard			
VB 6 ActiveX Doc Migration Wizard			
VB 6 Add-In Toolbar			
VB 6 API Viewer			
VB 6 Application Wizard			
VB 6 Class Builder Utility			
VB 6 Data Form Wizard		-	
(/) C D - t= O L'= - t \//= 4			Help
Jescription		- Load Behavior	
Package and Deployment Wizard	4	157 1 2 2 2 2 2 2 1 1	a da anta
		I♥ Loaded/UI	nioaded
		✓ Load on Si	tartup
		Loga office	

Sekil 4.24: Package and Deployment Wizard'ın yüklenmesi

Add-Ins Menüsüne baktığınızda Package and Deployment Wizard'ın eklendiğini göreceksiniz.



Şekil 4.25: Add-Ins menüsü

Bu komut bir sihirbazı işaret etmektedir. Ve bu sihirbaz sayesinde programlarınızı bir ".EXE" dosyası yapma yani çalıştırılabilir dosya haline getirme işleminden farklı olarak, hiç Visual BASIC olmayan bir bilgisayar da dahi çalıştırılabilir hale getirecek kurulum paketleri haline getirebilirsiniz. Yani birçok zaman kurulumunu yaptığınız oyun ve program kurulum CD'lerinde olduğu gibi uygulamanızı taşınabilir hale getirebilirsiniz.

Package and Deployment Wizard'ın kullanımını ve çalışmasını siz değerli öğrencilerimizin gayretlerine bırakıyoruz.

UYGULAMA FAALİYETİ

Aşağıdaki işlem basamaklarına göre uygulama faaliyetini yapınız.

İşlem Basamakları	Öne	riler
Visual Basic'i başlatıp yeni bir proje oluşturunuz.	Öğrenme	faaliyetinde
Formunuzun üzerine istediğiniz tasarımı yaparak, nesnelerin konumlarını tespit ediniz.	gerçekleştirilmiş uygulamalardan alabilirsiniz.	olan destek
 Nesnelerinizin özelliklerinde gerekli değişiklikleri yapınız. 		
Programınız için gereken kodları giriniz.		
Kullandığınız değişkenlerden birisini Watch Window ekranına aktarınız.		
Değişkenlerinizden birisini Local Window ekranına aktarınız		
Programınızı adımlayarak çalıştırınız.		
> Watch ve Local Window pencerelerini inceleyiniz.		
Programın çalışması esnasında değişkenlerinizin aldıkları değerleri kontrol ediniz.		
Programınızda kullandığınız değişkenlerinizden birisinin, değer değiştirdiği bir satıra gidip BreakPoint koyunuz.		
Programınızı adımlayıp Breakpoint'e kadar çalışmasını sağlayınız.		
Immediate Window ekranını açarak değişkenlerinizden birisinin değerini bu pencereden sorgulattırınız.		
F8 ile işaretli satırı çalıştırınız.		
Immediate Window'dan değişken değerini tekrar sorgulattırınız.		
> Breakpoint'i kaldırınız.		
 Programı çalıştırınız. 		
> Watch ve Locals pencerelerini tekrar inceleyiniz.		

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları cevaplayarak bu faaliyette kazandığınız bilgileri ölçünüz.

OBJEKTİF TEST (ÖLÇME SORULARI)

- 1. Programların çalışması sırasında ortaya çıkan hatalara ne ad verilir?
 - A) Debug C) Bugging
 - B) Bug D) Buggins
- **2.** Aşağıdaki seçeneklerden hangisi Visual Basic'de yer alan hata denetlemeye yardımcı pencerelerden birisi değildir?
 - A) Immediate WindowB) Watch Window

- C) Locals Window
- D) Form Layout Window
- **3.** Aşağıdaki seçeneklerden hangisi Visual Basic'de yer alan program modlarından birisi değildir?
 - A) Derleme Modu C) Tasarım Modu
 - B) Çalışma Modu

- D) Durdurma Modu
- 4. Run-Time Error (çalışma zamanı hatası) oluştuğunda ortaya çıkan program modu
 - aşağıdakilerden hangisidir? A) Derleme Modu
 - A) Derleme ModuC) Tasarım ModuB) Çalışma ModuD) Durdurma Modu
 - **5.** Immediate Window (anlık durum penceresi)'u görüntülemek için basmanız gereken tuş kombinasyonu hangi şıkta doğru olarak verilmiştir?
 - A) CTRL+IC) CTRL+GB) CTRL+AD) CTRL+M
 - **6.** İzleme ve durdurma ifadelerinde meydana gelen değişiklikleri her an doğru olarak görüntüleyen pencereye ne ad verilir?
 - A) Anlık Durum Penceresi(Immediate Window)
 - B) İzleme Penceresi(Watch Window)
 - C) Değişken Penceresi(Locals Window)
 - D) Form Konumlandırma Penceresi(Form Layout Window)
 - 7. Programın çalışması esnasında gösterilen değişkenlerin değerlerini değiştirebildiğiniz ve ayrıca kullanmış olduğunuz kontrollerin özelliklerini inceleyip değişiklikler yapabildiğiniz pencerenin adı hangi şıkta doğru olarak verilmiştir.
 - A) Anlık Durum Penceresi(Immediate Window)
 - B) İzleme Penceresi(Watch Window)
 - C) Değişken Penceresi(Locals Window)
 - D) Form Konumlandırma Penceresi(Form Layout Window)

- 8. Programın çalışması esnasında, çalışmayı satır satır izleme işine ne ad verilir?
 - A) Adımlama C) İzleme
 - B) Denetleme D) Takip
- **9.** Kontrollü bir şekilde programın istenilen satıra kadar çalıştırılıp o noktada bekletilmesini sağlayan adımlama tekniğinin adı nedir?
 - A) Stepping C) Step Into
 - B) Breakpoints D) Step Over
- 10. Step Into işlemini gerçekleştirmek için hangi kısayol tuşu kullanılır?
 - A) Ctrl+F9 C) F9 B) Shift+F8 D) F8
- **11.** Breakpoint oluşturmak için hangi kısayol tuşu kullanılır?A) Ctrl+F9C) F9
 - B) Shift+F8 D) F8
- 12. Projemizi çalıştırılabilir bir dosya haline getirmek için hangi komutu veririz?
 - A) File—Make C) Debug—Make

D) Run-Make

- B) Project—Make
- DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete geri dönerek tekrar inceleyiniz.

MODÜL DEĞERLENDİRME

UYGULAMALI TEST (YETERLİK ÖLÇME)

Modülde yaptığınız uygulamaları tekrar yapınız. Yaptığınız bu uygulamaları aşağıdaki tabloya göre değerlendiriniz.

AÇIKLAMA: Aşağıda listelenen ölçütleri uyguladıysanız	Evet	sütununa,						
uygulamadıysanız Hayır sütununa X işareti yazınız.	1							
DEĞERLENDİRME ÖLÇÜTLERİ	Evet	Hayır						
Visual Basic'i hatasız bir şekilde kurabildiniz mi?								
Visual Basic yardım dosyalarını yükleyebildiniz mi?								
Visual Basic yardım ekranlarını kullanabildiniz mi?								
Visual Basic'in Online yardım desteğini kullanabildiniz mi?								
Visual Basic'de yeni bir proje oluşturabildiniz mi?								
Visual Basic Araç Çubukları'nı kullanmayı öğrendiniz mi?								
Visual Basic ekranınıza proje penceresini yerleştirebildiniz mi?								
Visual Basic ekranınıza özellikler penceresini yerleştirebildiniz mi?								
Form ekranınıza istediğiniz kontrolleri yerleştirebildiniz mi?								
Nesnelerin özelliklerini özellikler penceresini kullanarak								
değiştirebildiniz mi?	değiştirebildiniz mi?							
Nesnelerin özelliklerini kod penceresini kullanarak değiştirebildiniz								
mi?								
Programınız için gereken kodları yazabildiniz mi?								
Programınızı çalıştırıp hatalarınızı düzeltebildiniz mi?								
Programınızı geliştirmeye yönelik düşünceler geliştirebildiniz mi?								
Programınızı adımlayarak çalıştırabildiniz mi?								
Programınızı BreakPoint kullanarak adımlayabildiniz mi?								
Watch Window kullanarak değişken değerlerinizi kontrol edebildiniz								
mi?								
Locals Window kullanarak değişken değerlerinizi kontrol edebildiniz								
mi?								
Immediate Window kullanımını öğrendiniz mi?								
Programınızı derleyip, çalıştırılabilir dosya haline dönüştürülebildiniz								
mi?								

DEĞERLENDİRME

Hayır cevaplarınız var ise ilgili uygulama faaliyetini tekrar ediniz. Cevaplarınızın tümü evet ise bir sonraki modüle geçebilirsiniz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ-1 CEVAP ANAHTARI

1	D	5	Α	9	В	13	Α
2	С	6	С	10	С	14	В
3	С	7	D	11	D		
4	Α	8	Α	12	С		

ÖĞRENME FAALİYETİ-2 CEVAP ANAHTARI

1	С	5	Α	9	С	13	Α
2	Α	6	B	10	Α	14	В
3	D	7	С	11	D		
4	D	8	Α	12	D		

ÖĞRENME FAALİYETİ-3 CEVAP ANAHTARI

1	С	4	С	7	Α	9	Α
2	D	5	Α	8	В	10	В
3	С	6	D				

ÖĞRENME FAALİYETİ-4 CEVAP ANAHTARI

1	В	4	D	7	С	10	D
2	D	5	С	8	Α	11	С
3	Α	6	В	9	В	12	Α

KAYNAKÇA

- HOLZNER Steven, The Coriolis Group, Visual Basic 6 Black Book, 1998.
- NORTON Peter, Peter Norton's Guide to Visual Basic 6, Macmillan Computer Publishing, 1998.
- PALA Zeydin, İhsan KARAGÜLLE, Visual Basic 6.0 Pro, Türkmen Yayınevi, İstanbul, 2002.