

# Ders 1:

## C# KURULUMU ve EDITÖRÜ

Revizyon: 15.02.2016

Hazırlayan: Mehmet Emin ÖZCAN

URL: [gantep.edu.tr/~ozcan](http://gantep.edu.tr/~ozcan)

Mail: [ozcan@gantep.edu.tr](mailto:ozcan@gantep.edu.tr)

( Bu dokümanda gördüğünüz yanlışlıkları veya referansı unutulmuş alıntıları lütfen yukarıdaki mail e yazın.)

Amaç: Aşağıdaki becerileri geliştirmek

- NTP programını (örn: C#) temin edip bilgisayar kurma.
- C# Editörünü, kıs-yollarını, C# in yapısını tanımak.
- Yeni, Aç, Kaydet, Düzenle işlemlerini yapmak.

İçindekiler:

	<u>KONU</u>	<u>SAYFA</u>
1.	Nesne Tabanlı Programlama	2
2.	Derste İşlenecek Konular:	2
3.	Dersin Programlama Dili C#	3
4.	Öğrencilerime Tavsiye:	3
5.	C# 2013 Kurulumu	3
6.	C# Kuralları:	4
7.	Toolbox	8
8.	Properties	8
9.	Klavye Kısa-Yol Tuşları	9
10.	Anahtar Kelimeler	9
11.	Uygulamalar	9
12.	Referans:	10
13.	Ders1 Soruları	11

## NESNE TABANLI PROGRAMLAMA

Nesne tabanlı programlama (NTP), Nesne yönelimli programlama (NYP) (İngilizce: OOP), bir bilgisayar programlama yaklaşımıdır. Liste 1.1 de görüleceği gibi pek çok programlama dili tarafından desteklenmektedir.

### Liste 1.1: NYP yi destekleyen Programlama dilleri [1]

ABAP, Ada95, AmigaE, BETA, Blue, Boo, C++, C#, Ceylon, Chapel, Clarion, CLU, COBOL, Cobra, ColdFusion, Common Lisp, COOL, CorbaScript, Curl, D, Dart, DataFlex, Dylan, E, Eiffel, Sather, Elixir, Falcon, Fortran2003, FPr, FreeBASIC, F-Script, F#, Gambas, GraphTalk, IDLscript, J, J#, JADE, Java, Groovy, Join Java, X10, Julia, Lasso, Lava, Lexico, Lingo, LISP, Logtalk, MATLAB, Modula-2, Modula-3, Nemerle, NetRexx, Noop, Oberon (Oberon-1), Oberon-2, Object Pascal, Delphi, Free Pascal, Turbo Pascal, Object REXX, Objective-C, OCaml, Omnis Studio, OpenEdge, Oz, Mozart, Perl, PHP5, Power Builder, Prototype-based, ABCL, Agora, Cecil, Cel, ECMAScript, ActionScript, JavaScript, JScript, Etoys, Io, Lua, Lisaac, MOO, NewtonScript, Obliq, REBOL, Self, Python, REALbasic, Revolution, Ruby, Rust, S, R, Scala, Seed7, SenseTalk, Simula, Smalltalk, Self, Bistro, Squeak, Squirrel, Swift, TADS, Tcl, Xotcl, Snit, incr Tcl, Ubercode, Vala, Visual Basic, Visual Basic .NET, VBScript, VBA, Visual FoxPro, Visual Prolog, XBase++ , ZZT-oop [1]

NYP, aşağıda özellikleri dolayısıyla programlama bakımını ve aynı yazılım üzerinde birden fazla kişinin çalışmasını kolaylaştırmaktadır.

- Yazılımda birimselliği (modularity).
- Bilgi gizleme (information hiding).
- Veri soyutlama (data abstraction).
- Çok biçimlilik (polymorphism).
- Kalıtım (inheritance).

NYP nin birimselliğine göre; her bilgisayar programı, etkileşim içerisinde olan birimler veya nesnelere kümesinden oluşmaktadır. Bu nesnelere her biri, kendi içerisinde veri işleyebilir ve diğer nesnelere ile çift yönlü veri alışverişinde bulunabilir. Hâlbuki NYP'den önce var olan tek yaklaşımda (Yordamsal programlama), programlar sadece bir komut dizisi veya birer işlev (fonksiyon) kümesi olarak görülmektedirler.

Günümüzde çok çeşitli nesne tabanlı programlama dilleri olmasıyla beraber, en popüler diller sınıflar üzerine kurulmuşlardır (class-based). Bu dillerde nesnelere sınıfların birer üyesidir ve nesnelere tipini de bu sınıflar belirlerler.

Nesne yönelimli programlama dilleri yukarıda adı geçen tüm öğelere sahip olurken, Ada, JavaScript, Visual Basic gibi nesne tabanlı programlama dilleri birkaçından yoksundur, bu dillerin başlıca yoksunluğu kalıtıma sahip olmamalarıdır. [2]

### Nesnelere ve sınıflar

NYP'yi destekleyen programlama dilleri genellikle tekrar kullanım ve genişletilebilirlik açısından, prototipler ve sınıflar şeklinde kod kalıtımına sahiptirler ve NYP'deki nesnelere kimi zaman gerçek dünyada bulunan şeylere karşılık gelebilir. Mesela bir grafik programı "kare", "üçgen" ve "yuvarlak" gibi nesnelere sahip olabilir. Bazen nesnelere daha soyut şeyleri de temsil edebilirler, mesela bir hesap makinesi programı logaritma almanızı ya da metreyi fit'e çevirmenizi sağlayacak nesnelere sahip olabilir. İşte bu bahsi geçen nesnelere belli sınıflar altında bulunurlar, mesela "Kasım" adında bir nesne Aylar sınıfının bir üyesi olabilir ya da "Faktöriyel" olarak isimlendirilmiş bir nesne Fonksiyonlar sınıfında yer alıyor olabilir. [2]

### DERSTE İŞLENECEK KONULAR:

MEB'in liselerde okutulan Nesne Tabanlı Program dersinin müfredatında kalıtım konusu da dâhil olmak üzere NYP neredeyse tüm konuları işlenmiştir. Bu yüzden dersimizde NTP adı altında NYP göreceğiz.

1. C# Kurulumu ve Editörü
2. Değişken ve Operatörlerin Kullanımı
3. Karar Kontrol ve Döngü Kontrol Deyimleri
4. Fonksiyonlar
5. Aduzayı, Sınıf, Alan ve Metot
6. Diziler
7. Lokal ve Global Referanslar
8. Dizi ve Koleksiyonlar
9. Standart Bileşenler
10. Gelişmiş Bileşenler
11. Veritabanı
12. Bileşen oluşturma

## DERSİN PROGRAMLAMA DİLİ C# :

- En iyi dillerden biri.
- Öğrencilerimin en sevdiği, aşına olduğu dil
- Öğrenmesi kolay, basit, ilk öğrenilecek diller arasında
- C# NYP'yi tam destekler.
- Microsoft bu programa çok yatırım yaptı.
- Derleme süresi kısa.
- Geniş alanlara yayılmış milyonlarca programcı var. Piyasada çok C# uygulaması mevcut.
- Kullanımı kolay.
- (Büyük şirketlerde) Yaygın kullanılıyor.
- PC, Cep, Smart uygulamaları destekler.
- Platform bağımsızdır.
- Windows lisansı olana bedavadır.
- Rakibi Java'ya benzer kod yapısı vardır.
- Java kadar performansa sahip değil.
- Kod-Maymunu diye bilinen PHP ye göre performansı oldukça iyidir.
- En iyi derleyici editörü sahiptir. Java nın editör yönünden sıkıntıları var.
- Nesne yönünden C++ dan öndedir.
- Delphi kapatılmış. Artık geliştirilmiyor.

## ÖĞRENCİLERİME TAVSİYE:

Konuyu okuyup, örnekleri pratik yapmamak anlamsızdır. Boş iştir. Yanındaki arkadaşının yaptığı uygulamayı seyretmek de anlamsızdır. Bazıları derki iyi bir yazılımcı olmak için 50000 ile 100000 satır arası kod yazmak gerekir[4]. Maalesef MYO da makine bölümünden mezun olan tornayı frezeyi, inşaattan mezun olan inşaatla ilgili işleri rahat yapabildiği halde; benim bilgisayar bölümü öğrencilerim iki sayıyı toplayan program yazamıyor.

Hâlbuki dünyanın en zenginleri bilgisayar programıyla ilgilenen şirketlerdir. Programlama

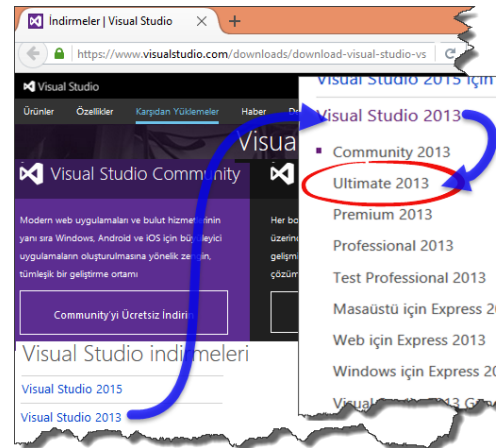
alanında bayağı boşluk var ve bu boşluğu dolduran köşeyi döner. Ülkemizin de programcıya ihtiyacı var. Benim öğrencilerim için tavsiyem çok pratik yapsınlar. Her gün her gün program yazıp çalıştırsınlar. Varsın hata versin çalışmasını önemli değil, önemli olan uzun süre bu işle uğraşmaktır.

Svetlin Nakov [4], program yazmayı araba sürmeye, suda yüzmeye benzetir ve sadece kitap okumakla yüzmeye öğrenilmez, öğrenmek için çok pratik yapmalısın der.

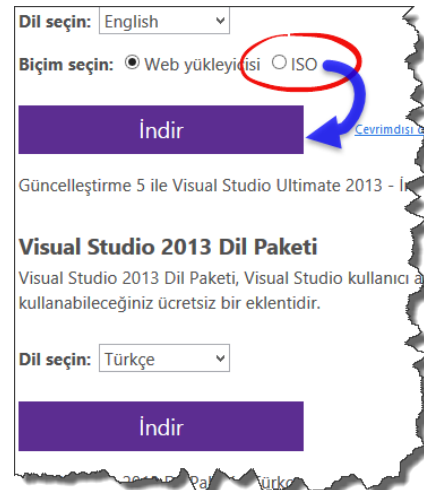
## C# 2013 KURULUMU

1.1) [www.microsoft.com/visualstudio/tur/downloads](http://www.microsoft.com/visualstudio/tur/downloads) adresinden Visual Studio 2013 seçilir.

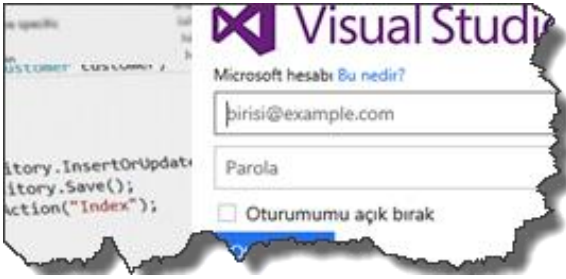
1.2) Gelen listeden indirmek istenen sürüm seçilir. Biz Ultimate 2013 sürümünü seçtik.



1.3) İso dosyasını indirme linki tıklanır.

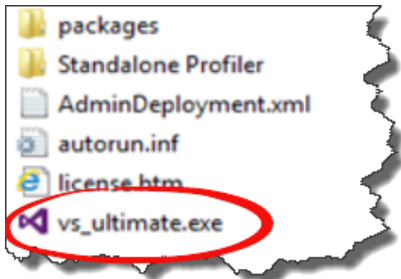


1.4) Microsoft Live hesabı ile bağlantılı olan Visual Studio sayfasına yönlendirilir. Burada Live hesabı ile giriş yapacağız.



1.5) Bu ekrandaki gerekli bilgileri doldurulur ve Ultimate 2013 yazısını yer tıkladıktan sonra dosyam bilgisayarımıza iner.

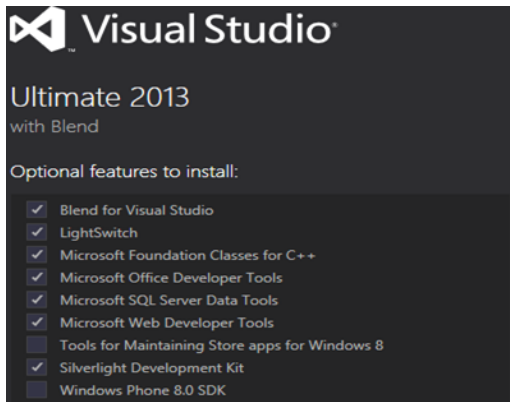
1.6) İndirilen iso dosyası tıkladıktan sonra içindeki vs\_ultimate.exe çalıştırılır.



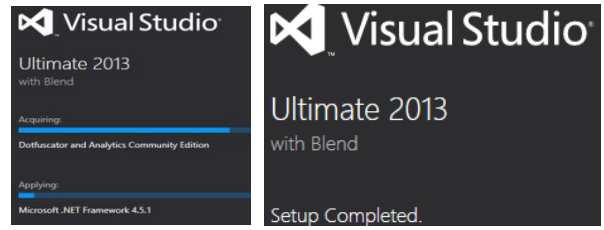
1.7) Gelen ekranda, kurulum için onaylanması gereken yerler seçilir.



1.8) Kurulacak toollarında seçiminden sonra Install tuşu tıklanarak kurulum başlatılır.



1.9) Setup Completed ile kurulum işlemi sona ermiş olur.



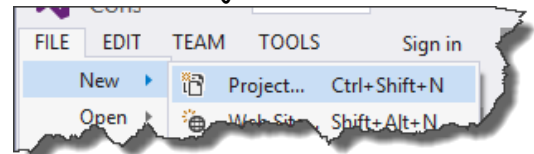
## C# KURALLARI:

1.10) C# veya CSHARP si-şarp diye okunur.

1.11) C# derleyicisini açmak için aşağıdaki komut tıklanır.

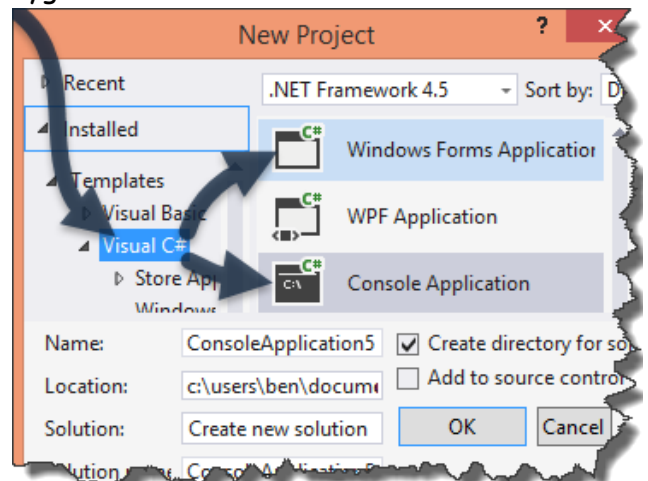
Başlat → Programlar → Microsoft Visual Studio

1.12) C# programı genelde birden fazla dosyadan oluştuğu için program yerine proje denir. Konsol veya Form uygulamaları için önce bir yeni proje oluşturmak gerekir. Yeni Proje oluşturmak için Ctrl+Shift+N veya File→New→Project tıklanır.



Klavye ve ekranın birleşmesine "konsol" denir. Görsel olmayan, az yer kaplayan DOS ekranı gibi program için konsol uygulaması tercih edilir.

Windows da kullanılan pencereci, fareli görsel ekran isteniyorsa Windows Form uygulaması kullanılır.



Konsol uygulaması için

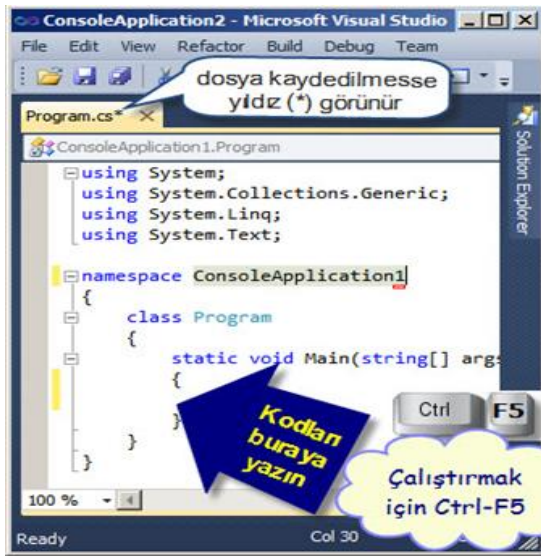
Visual C# → Console Application

Form uygulaması için

Visual C# → Windows Form Application

Seçeneklerini tıkların.

- 1.13) Yeni Console uygulaması oluşturduğumuzda aşağıdaki Program.cs dosyası yazılı olarak ekrana gelecektir. Bundan sonra yapmamız gereken Main fonksiyonu içindeki süslü parantez { } arasına kodları yazmaktır.



- 1.14) Açıklama cümleleri programın işleyişine etki etmez. Programı dahi iyi anlamak için programcıya bilgi vermeyi amaçlar. Birkaç satırlık veya aralık bir yerde açıklama yapılacaksa /\* \*/ kullanılır. Kendinden sonra sadece o satırın devamında açıklama cümlesi yapılacaksa // kullanılır. C# Bir ara XML türü açıklama satırı olan /// kullandı, /// web sayfalarında gördüğümüz açılır kapanır türde, kapanınca içeriği görünmez olan açıklama satırıdır.

```
int a,b, /*Açıklama cümlesini*/ c,d,e;  
int a,b, /* Açıklama  
cümlesini */ c,d,e;  
int a,b,c,d,e; // Açıklama satırı
```

- 1.15) static void Main(string[] args) { }  
Main fonksiyonu programın giriş noktası olarak kabul edilir. Bilgisayar, Çalışmaya Main Fonksiyonundan başlar. Main fonksiyonu

olmasa program çalışmaya başlayamaz. Static öneki Main fonksiyonunu program sonlanana kadar hayatta kalmasını, sonlanmamasını sağlar. void değişken tipi Main fonksiyonunun geriye değer döndürmeyeceği anlamındadır. Main tipi int veya void olabilir.

```
static void Main()  
{  
    // KODLARI BURAYA YAZIN  
}
```

veya

```
static int Main()  
{  
    KODLARI BURAYA YAZIN  
    return (0);  
}
```

string[] args; program çalıştırıldığında program isminden sonra girilen parametreleri içerir. Örneğin: programımın adı dir olsun. Dir programımı çalıştırmak için dos ekranına dir yazıp enter ı tıklamam yeter. Fakat Dir ali veli.txt yazarsak args[0]="ali" ve args[1]="veli.txt" olur.

- 1.16) class Program { }

Program isimli sınıf (class) oluşturur. C# da Her şey nesnedir. Nesneyi oluşturmak için nesne sınıfının oluşturulması şarttır. Dolayısıyla C# da yapılan her şey mutlaka bir sınıf (class) içinde yapılması şarttır. Kullanılmazsa program çalışmaz. Bu konu Sınıf ve nesne dersinde detaylı işlenecektir.

- 1.17) namespace aduzayı{ } sınıfları içine alan klasör veya dosya görevi gören bir yapıdır. Ön ek olarak kullanılır. Namespace ler using kelimesi ile çağrılırlar. Aduzayı isimli dersinde detaylı işlenecektir.

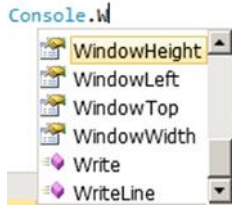
- 1.18) Console Application seçeneğini seçmişsek; C#, program yazmamız için içinde bazı kod satırlarının bulunduğu bir editörü ekrana getirir. ( Bu kodları silip kendi programımızı yazabiliriz. )

1.19) noktalı virgül ( ; ) her komuttan sonra mutlaka kullanılır. Komut bitti anlamına gelir. Yan yana birkaç kez kullanılmasında sakınca yoktur.

```
static void Main(){ ; ; ; ; ; ; ; ; }
```

1.20) Ctrl+Boşluk Tuşu, program yazarken yazdığımız kod a en yakın kodları satır satır listeler. Listedekilerden birini seçildiğinde ilgili yere kod satırı yapıştırılır. Ctrl+Boşluk Tuşu nu kullanmanızı özellikle öneriyoruz, bu sayede hatalı kod yazmasanız, daha hızlı kod yazarsınız, hatırlamadığınız kod u hatırlamış olursunuz.

Not: Aşağıdaki resimde W harfi tıklandığında W harfi ile başlayan listenin ekrana gelmesi görülmektedir.



1.21) Insert Snippet, komut bloklarını kod sayfasına yapıştırır. Örnek: for yazıp Tab+Tab tuşlarını tıklarsanız veya fare sağ tuşundan insert-Snipped→C#→for u tıklarsak aşağıdaki gibi for kod bloğu kendiliğinden ekrana yazılır.


```
for (int i = 0; i < length; i++)  
{  
  
}
```

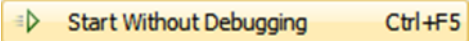
1.22) Write() yaz, WriteLine() ise yaz sonra bir satır aşağı in komutudur. Önerine System.Console ön ekini getirilmelidir. Programın başında using System; komutu varsa sadece Console getirmek yeterli olur. C# 2015 de using static System.Console; kodu uygulanmışsa Write(); komutu ön ek almadan kullanılabilir.


```
System.Console.Write("Merhaba MYO");  
// Merhaba yazar ve imleç o satırda kalır.  
System.Console.WriteLine(88);  
// Merhaba yaz sora imleç alt satıra in
```


1.23) ReadLine() OKU sonra bir satır aşağı in komutudur. Önüne System.Console ön ekini getirilmelidir. Programın başında using System; komutu varsa sadece Console getirmek yeterli olur. Sadece string veri okurlar. int veri okuyamazlar.

```
Console.ReadLine(); // Klavyeden bir yazı oku.  
String okunan=Console.ReadLine();  
// Klavyeden okunanı okunan değişkene ata.
```

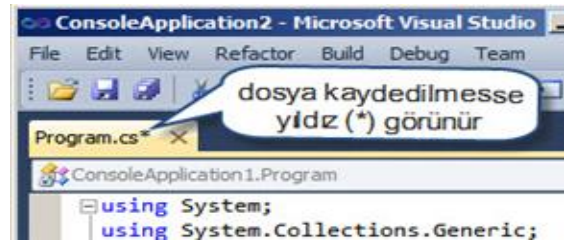
1.24) Ctrl-F5 Programı derler, kaydeder ve çalıştırır; program durduğunda da bir tuşa basmanızı bekler. Ctrl-F5 yerine Debug -> Start Without Debugging veya  ikonu kullanılabilir. Ctrl-F5 genelde konsol uygulamaları için kullanılır.

 Start Without Debugging Ctrl+F5

F5 ise genelde Windows uygulamalarını derleyip çalıştırmak için kullanılır. F5 yerine Debug -> Start veya  ikonu kullanılabilir.

 Start Debugging F5

1.25) Program değişiklik yapıldığı halde Kayıt yapılmamışsa, dosya isminin yanında yıldız (\*) işareti görünür Aşağıdaki resimde Program.cs\* daki \* a dikkat edin.



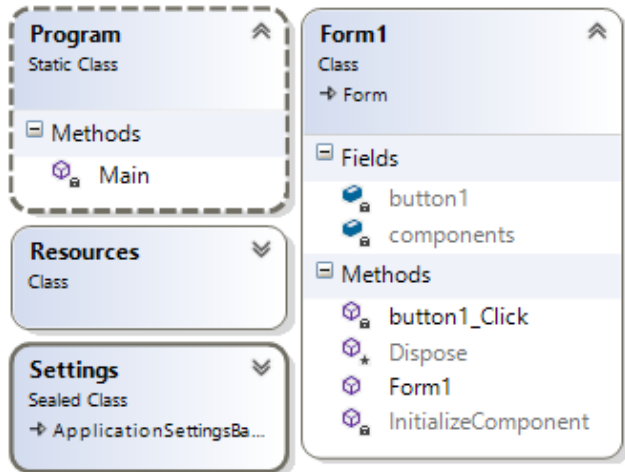
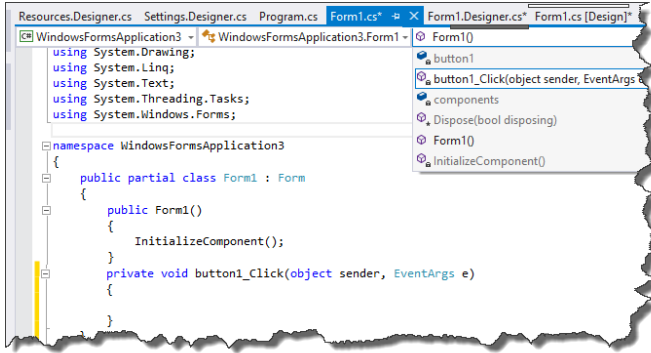
1.26) Kayıt yeri genelde belgelerimin altındadır. ...Belgeler\VisualStudio 2013\Projects

1.27) Uygulamamızı farklı yere kaydetmek (save) etmek için bence hiç C#'dan yardım almayın. Windows Gezini ile Belgelerim 'in ...Documents\Visual-Studio-2013\Projects\... dizinine gidin. Yaptığınız projeyi kopyalayıp istediğiniz başka bir dizine kaydedin.

1.28) Önceki projelerinizden birini çalıştırmak için File→Recent-Projects-and-Solutions sekmesinden projenizi seçip çalıştırabilirsiniz

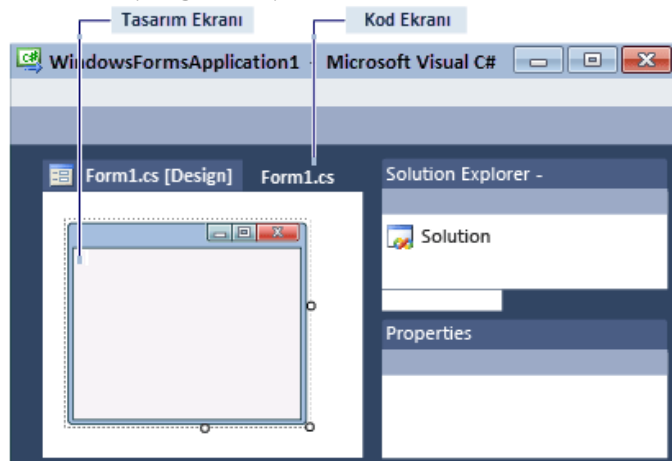
1.29) Yeni Form uygulaması oluşturduğumuzda bazıları aşağıda listelenmiş olan dosyalar oluşur.

- Form1.cs [Design] (Tasarım ekranıdır )
- Form1.cs (events ları barındırır)
- Form1.Designer.cs (nesne ve alan ı barındırır)
- Program.cs (Main metodunu barındırır)
- Resources.Designer.cs
- Settings.Designer.cs



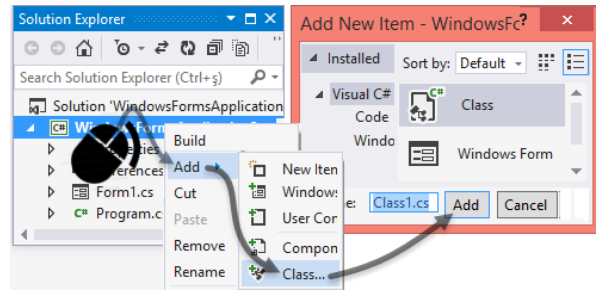
Dosyaların Class Diagram Görüntüsü

1.30) Solution Explorer, Dosyalama işlemleri için; Tasarım ekranı, Forma Nesne eklemek için; Properties, Nesne özelliklerini/olaylarını ayarlamak için ve Kod Ekranı da kalan birkaç kod u programa yazmak için kullanılır.

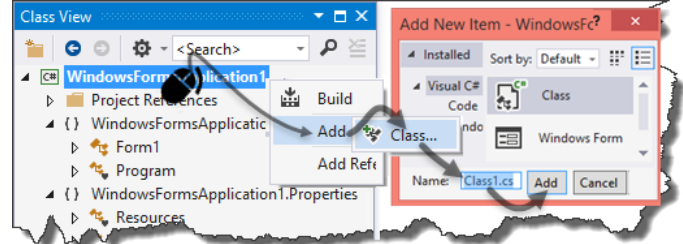


1.31) Projeye yeni sınıf eklemek için (kısayol tuşu: ctrl + shift+A dir.)

- Solution Explorer→Proje→FareSağTuşu→Add→Class

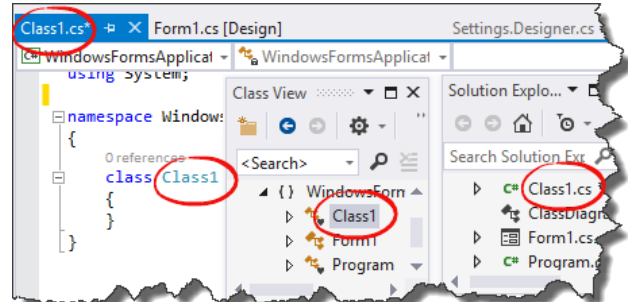


- Class View→FareSağTuşu→Add→Class



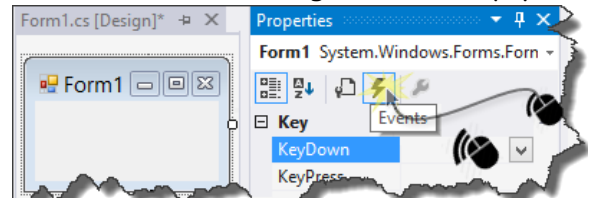
Class View için, View Menüünden seçebilirsiniz.

- Project Menü → Add Class
- dosyaların birinin içine yeni class elle yazılır. Şayet Class ekleme başarılı ise aşağıdaki resinde işaretli alanlarda görüldüğü gibi Class1 isimli sınıf projemize eklenmiş olur.

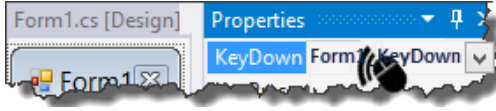


1.32) Mevcut sınıflardan (class) birinde kes, kopyala, yapıştır, ismini-değiştir gibi işlemler yapılması istendiğinde; Ya elle yapılır, yâda Solution Explorer, Class-View gibi pencerelerden Class1 yazılı yerin üzerinde fare sağ tuşu tıklanır, seçeneklerden ilgili işlem seçilerek yapılır.

1.33) Nesneye olay (Events) eklemek için, önce Properties-Penceresi deki olay (Events) ikonu/sekmesi tıklanır. Daha sonra ilgili olay satırındaki kutucukta gerekli işlem yapılır.

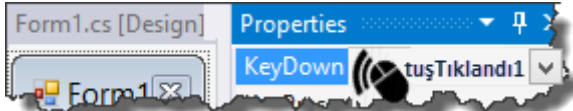


1.34) Olay kutucuğu boşsa ve fare ile çift tıklanırsa: olay ve ismi otomatik oluşturulur.



```
private void Form1_KeyDown(object sender, KeyEventArgs e) {
}
```

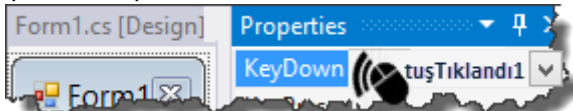
1.35) Olay kutucuğu boşsa ve içine bir isim yazıp çift-tıklanırsa//enter'lanırsa bu isimde bir olay oluşturulur.



```
private void tuşTıklandı1(object sender, KeyEventArgs e) {
}
```


1.36) Olay kutucuğu doluyken fare ile çift tıkladığında kod sayfasındaki olay kodlarına gider.

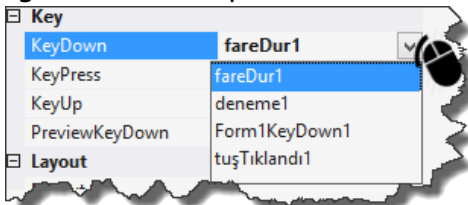
1.37) Olay kutucuğu içinde bir isim varsa ve bu isim değiştirildikten sonra çift tıklansa / enter'lansa yeni isimde farklı bir olay oluşturulur, eski olay kodları silinmez kod sayfasında pasif olarak durur.



```
private void Form1_KeyDown(object sender, KeyEventArgs e) {
    // Buradaki komutlar göz ardı edilmiş
}

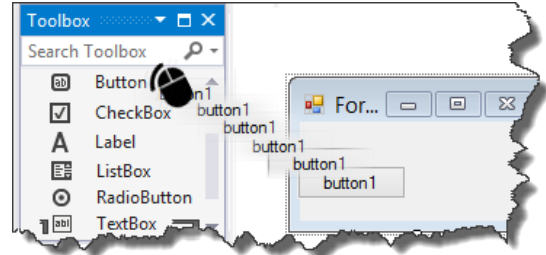
private void tuşTıklandı1(object sender, KeyEventArgs e) {
    // bu olay kullanılıyor
}
```

1.38) Olay kutucuğu sağındaki, menüden seçmek ikonu  tıklanırsa, kod satırında yazılı olabilecek tüm metotlar listelenir. Bu listeden seçtiğimiz metot olay olarak kabul edilir.



## TOOLBOX

1.39) Bileşenlerin bulunduğu listeye Toolbox denir. Ekranda görünmüyorsa View menüsünden görünür yapılabilir. Tasarım ekranındayken aktif olur. Forma eklenmek istenen bileşen toolbox dan Forma sürüklenip bırakılarak kullanılır. Aynı bileşenden birden fazla kullanılacaksa sürükleme işlemi ctrl tuşu basılıyken yapılır.

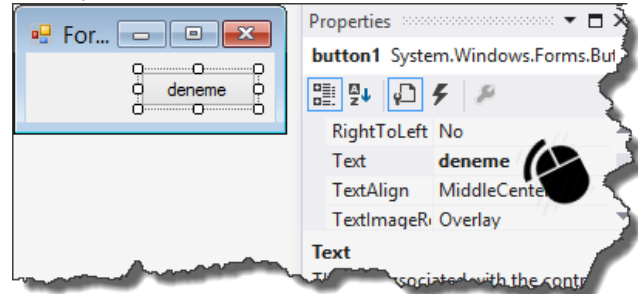


toolbox dan form'a Button sürükleyip-brakma

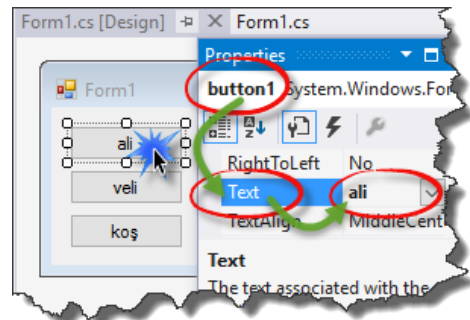
## PROPERTIES

1.40) Bileşen özellik/olay 'larını tasarım zamanında değiştirmek için kullanılan ekrana Properties denir. Ekranda görünmüyorsa View menüsünden görünür yapılabilir. Kısayolu F4 dür.

Olay için , özellik  için tıklanır.



Button1.Text inin deneme olarak değiştirilmesi



Yukarıdaki resimde Properties ekranında button1 isimli nesnenin Text alanına 'ali' yazma işlemi görülmektedir. Bunun işlemin kod karşılığı button1.Text="ali"; dir



## KLAVYE KISA-YOL TUŞLARI

1.41) Menüden seçimini, klavyeden yaptıran tuşa kısa-yol-tuşu (ing.Keyword) denir.

(Not: Laptoplarda Function (Fn) tuşundan sonra kullanılabilir)

Pencereler Arası Geçiş	
Properties (Özellik) penceresini aç	F4
Solution Explorer'ı aç	Ctrl+Alt+L
Toolbox'ı aç	Ctrl+Alt+X
Kod/Designer penceresini aç	F7 / Shift+F7
Dosya/pencere arası gezinme	Alt + F7 / Ctrl + Tab
Form/KodSayfası arası geçiş	Ctrl + Alt, S

Çalıştır/Hata ayıkla(Run/Debug)	
Çalıştır (Debug yaparak)	F5
Çalıştır (Debug yapmadan)	Ctrl+F5
Önceki/sonraki adıma Debug	F11 / F10 / Shift + F11
Projeyi derle çalıştır	Ctrl + Shift + B / F6
Çalışmayı Durdur	Shift+F5

Yardım	
Seçili öğeyi helpde ara	F1
Hazır kod ekleme (Insert Snippet)	Ctrl+K, Ctrl+X
İnsert Snippet	Ctrl+Boşluk
Hata Listesi	Ctrl + ç, E
Komutun tanımına gider.	F12
Komutu gider.	Alt + F12

Arama	
Referansı dosyasını adresini	Shift + F12
Dosyada ara	Ctrl + F / Ctrl + Shift + F
bul	F3 / Shift + F3
Seçilene ara	Ctrl + F3
Bul ve Değiştir	Ctrl + H / Ctrl + Shift + H

Yansıma kısayolları	
Arayüze Çıkar (Metod'u)	Ctrl + R, I
alanı kapsülle (alan'ı)	Ctrl + R, E
Yeniden Adlandır (Metod'u)	F2 or Ctrl + R, R
Remove Parameter	Ctrl + R, V
Reorder Parameters	Ctrl + R, O
Insert Snippet Listesi	Ctrl + K, X
insert Snippet penceresi	Ctrl + K, S
Projeye yeni dosya ekle	Ctrl+Shift+A
Projeye mevcut dosya ekle	Shift+Alt+A
Komutu Tamamlar	text + Tab + Tab
Snippet Yöneticisi	Ctrl + K, B
seçili öğeyi yeni metod a taşı	Ctrl + R, M

Daha fazla kısa yol için aşağıdaki kelimeleri aratın.  
Visual Studio 2013 Keyboard Shortcuts

## C# ANAHTAR KELİMELERİ

1.42) Program içinde kullanılan özel bir görevi olan kelimelere anahtar kelime (ing.Keyword) denir. Anahtar kelimeler genelde komutlardır.

Tablo 1, Anahtar kelime listesi

Anahtar Kelime Listesi					
abstract	as	base	bool	break	byte
case	catch	char	checked	class	const
continue	decimal	default	delegate	do	double
else	enum	event	explicit	extern	false
finally	fixed	float	for	foreach	goto
if	implicit	in	int	interface	internal
is	lock	long	namespace	new	null
object	operator	out	override	params	private
protected	public	readonly	ref	return	sbyte
sealed	short	sizeof	stackalloc	static	string
struct	switch	this	throw	true	try
typeof	uint	ulong	unchecked	unsafe	ushort
using	virtual	void	volatile	while	

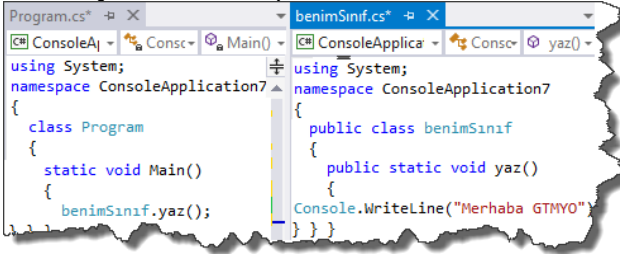
## UYGULAMALAR

### Uy 1.1: Konsolda Sınıf Ekle, Fonksiyon Çağır.

- Yeni Konsol Projesi oluşturun. (bak. 1.12, 1.13)
- Solution Explorer kullanarak, **benimSınıf** isimli bir sınıf (class) ekleyin. (bak. 1.31)
- benimSınıf isimli sınıfın içine aşağıdaki metodu ekleyin.

```
public static void yaz() {  
    Console.WriteLine("Merhaba GTMYO");  
}
```
- Main Metoduna aşağıdaki kod satırını ekleyin.

```
benimSınıf.yaz();
```
- Programı çalıştırın. (bak. 1.24)
- Yaptığınız bu uygulamayı kaydedildiği dizinden kopyalayıp flash diskinize kaydedin. (bak. 1.25, 1.26, 1.27)
- C# Editörünü kapatın.
- C# Editörünü çalıştırın.
- En son yaptığınız uygulamayı Resent-Project den bulup çalıştırın. (bak. 1.28)



```
Program.cs  
using System;  
namespace ConsoleApplication7  
{  
    class Program  
    {  
        static void Main()  
        {  
            benimSınıf.yaz();  
        }  
    }  
}
```

```
benimSınıf.cs  
using System;  
namespace ConsoleApplication7  
{  
    public class benimSınıf  
    {  
        public static void yaz()  
        {  
            Console.WriteLine("Merhaba GTMYO");  
        }  
    }  
}
```

### Uy 1.2: Form'da Sınıf Ekle, Fonksiyon Çağır.

- Yeni Form Projesi oluşturun. (bak. 1.12)
- Solution Explorer kullanarak, **benimSınıf** isimli bir sınıf (class) ekleyin. (bak. 1.31)
- benimSınıf isimli sınıfın içine aşağıdaki metodu ekleyin.

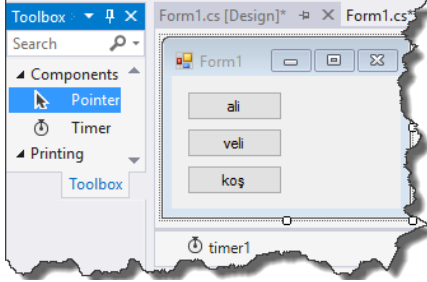
```
public static string metin()  
{  
    Return "Merhaba GTMYO";  
}
```
- Toolbox dan bir adet Button bileşeni alıp Form1 e ekleyin (bak. 1.40)
- Button1 i çift tıklayıp içine aşağıdaki kodu yazın.

```
Button1.Text= benimSınıf.metin();
```
- Programı çalıştırın. (bak. 1.24)
- Yaptığınız bu uygulamayı kaydedildiği dizinden kopyalayıp flash diskinize kaydedin. (bak. 1.25, 1.26, 1.27)
- C# Editörünü kapatın.
- C# Editörünü çalıştırın.
- En son yaptığınız uygulamayı Resent-Project den bulup çalıştırın. (bak. 1.28)

### Uy 1.3: Verilen kodları Form ekranı için ayarla

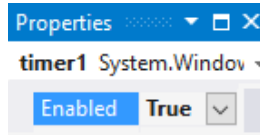
(Not: Kodları anlayamayabilirsiniz önemli değil. Şu an önemli olan verilen kodları Editöre koyup gerekli düzenlemeleri yapabilmektir.)

- Yeni Form Projesi oluşturun. (bak. 1.12)
- Toolbox penceresini kullanarak (bak. 1.39) üç adet Button ve bir adet timer nesnesini form nesnesi üzerine sürükleyip bırakın.



- Properties penceresini kullanarak (bak. 1.40) form üzerindeki nesnelere aşağıdaki gibi 'ali', 'veli' ve 'koş' olarak değiştirin.

Timer1.Enabled=true yapın.



- Aşağıdaki iki pencere iç içe, içteki pencere içindekileri kodları kopyalayın

```
using System; using System.Drawing;
using System.Windows.Forms;
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

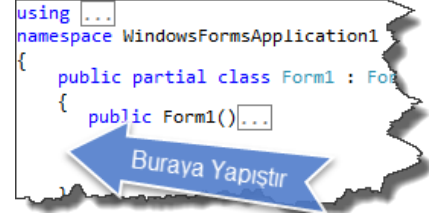
        private void button3_Click(object sender, EventArgs e)
        {
            button2.Left += 10;
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            button1.Left += 10;
            if (button1.Left > Width)
            {timer1.Enabled = false; Text="Kaybettiniz";}
            if (button2.Left > Width)
            { timer1.Enabled=false; Text="Şampiyonsunuz";}
        }
    }
}
```

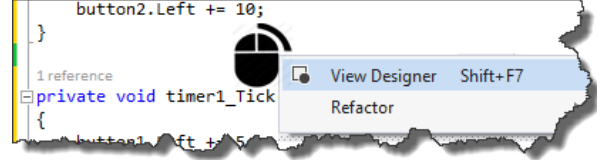
- Form üzerinde fare→SağTuş→View Code veya F7) yapıp kod ekranına geçin.



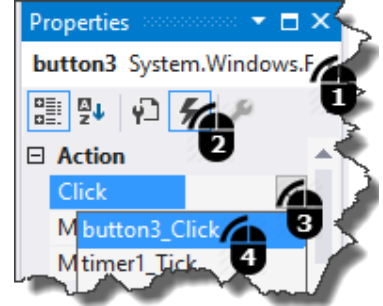
- Aşağıda görünen "Buraya Yapıştır" ile işaretli alana yapıştırın.



- Kod sayfasında fare→SağTuş→View Designer veya Shift+F7 Form ekranına geçin.



- Properties penceresinden (bak. 1.40) sırayla;
  1. Açılır menüden button1 i seçin.
  2. Olay (Events) ikonunu tıkklayın.
  3. Click satırındaki açılır menüyü tıkklayın.
  4. Menüden button3\_Click i seçin.



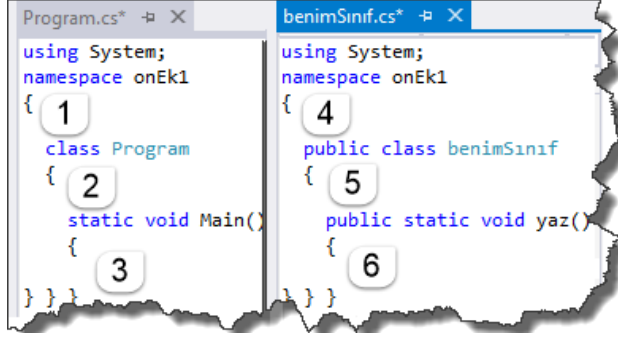
- Events dan timer1.Tick ini timer1\_Tick olarak ayarlayın. (bak. 1.40) yani sırayla;
  1. Properties Menüsünden timer1 i seçin.
  2. Listedeki Tick i bulun.
  3. Satırın sağındaki açılır menüden timer1\_Tick i seçin.
- Programı çalıştırın. (bak. 1.24)
- Yaptığınız bu uygulamayı kaydedildiği dizinden kopyalayıp flash diskinize kaydedin. (bak. 1.25, 1.26, 1.27)
- VS editörünü kapatın. Tekrar açın.
- En son yaptığınız uygulamayı Resent-Project den bulup çalıştırın. (bak. 1.28)

### REFERANS:

- [1]. List of object-oriented programming languages en.wikipedia.org/wiki/List\_of\_objectoriented\_programming\_languages, 13 October 2015
- [2]. Svetlin Nakov, Fundamentals of Computer Programming with C# ,2013,(The Bulgarian C# Programming Book), ISBN 978-954-400-773-7
- [3]. <http://agilecoder.net/2014/05/28/visual-studio-cheat-sheat/>

## DERS-1 SORULARI

1.1) `if(2<3);` ifadesi Hangi alana yazılabilir.



- a. 1,4
- b. 2,5
- c. 3,6
- d. 5,6
- e. 4,3

1.2) Nesne için hangisi söylenemez?

- a. Diğer nesnelerle etkileşim içinde olabilir.
- b. Değer taşıyabilirler.
- c. Program kodları (fonksiyon) içerirler.
- d. Verileri gizleyemez.
- e. Kalıtım ile başka birimleri kopyalayabilirler.

1.3) C# kurulumu için hangisi söylenemez?

- a. Kurulum CD si [www.microsoft.com](http://www.microsoft.com) 'dan parasız olarak temin edilebilir.
- b. Kurulum Windows işletim sistemine yapılır.
- c. Windows orijinal değilse extra ücret ister.
- d. Kurulumu kısa ama zordur.
- e. Donanımı yeterli olmayan bilgisayarlarda kasma yapar.

1.4) C# için hangisi söylenemez?

- a. C# ile üç boyutlu oyun yazılabilir.
- b. Derleme süresi kısadır.
- c. Piyasada çok sayıda dokümanı vardır.
- d. En iyi programlama dillerindendir.
- e. İnternet programlamada kullanılmaz.

1.5) C# da hangi tuş tıklanırsa for kelimesi aşağıdaki koda dönüşür.

```
for (int i = 0; i < length; i++)  
{  
    }  
}
```

- a. Tab
- b. Alt+Tab
- c. Ctrl+Tab
- d. Shift+Tab
- e. Tab+Tab

1.6) C# da wh yazıp Tab+Tab tuş tıklanırsa ekrana aşağıdakilerden hangisi yazılır.

- a. Tab+Tab
- b. whTabTab
- c. while (true) { }
- d. void wh(){ }
- e. int wh(){return 0; }

1.7) Aşağıda görülen 'cw' harfinin sonunda iken Tab tuşü iki kez tıklanırsa 'cw' ne olur?

```
class Prg{static void Main(){    CW  }}
```

- a. cwTabTab a dönüşür.
- b. Hata ayıklaması yapılır.
- c. System.Console.WriteLine(); a dönüşür.
- d. Mevcut sayfayı kapanır. Yeni sayfa açılır.
- e. Hiçbir şey değişmez

1.8) Mevcut Projeye, class A{ int b; } eklenecek. Aşağıdakilerden hangisi yapılmalıdır?

- a. Class Explorer açılıp içine yazılır.
- b. Solution Explorer açılıp içine yazılır.
- c. ctrl+shift+A, Name:A.cs, Add i tıklanır, güzel parantez içine int b; yazılır.
- d. File→New→Project→Class tlanır Add A yazılır. İçine int b; yazılır.
- e. Hiçbir

1.9) Programın kod satırlarında alininsı=3; yazılı yerinde hata mesajı veriyor ne yapmak gerekir?

- a. ctrl+boşluk tuşu ile otomatik düzeltilir.
- b. Ctrl+Tab tuşu ile otomatik düzeltilir.
- c. Tab+Tab tuşu ile otomatik düzeltilir.
- d. Ctrl+P tuşu ile otomatik düzeltilir.
- e. Hiçbir

1.10) Projenin uygun yerine aşağıdaki kodlar ekledikten sonra hangi ayar yapılır.

```
private void button3_Click(object sender, EventArgs e)  
{  
    button2.Left += 10;  
}
```

- a. button3→Properties→Events→Click deki menüden button2\_Click i seç
- b. SolutionExplorer daki pencereye de ekle.
- c. Programın tekrardan yaz.
- d. F5 tuşunu tıkla.
- e. Hiçbir