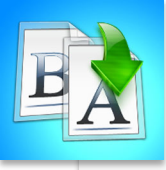


# NESNE TABANLI PROGRAMLAMAMANIN TEMELLERİ VE C#



## İÇİNDEKİLER

- Nesne Tabanlı Programlama
  - Temel Tanımlar
  - Temel Yapı, Özellikler ve İlkeler
  - Neden Nesne Tabanlı Programlama
  - Neden C# ve C# Avantajları
  - Alternatif Yöntemler ve Eleştiriler
- Visual Studio.NET 2019
  - C# ile Geliştirmek
  - C#'ın Güçlü Programlama Özellikleri
  - Kurulum
  - Geliştirme Ortamı



## HEDEFLER

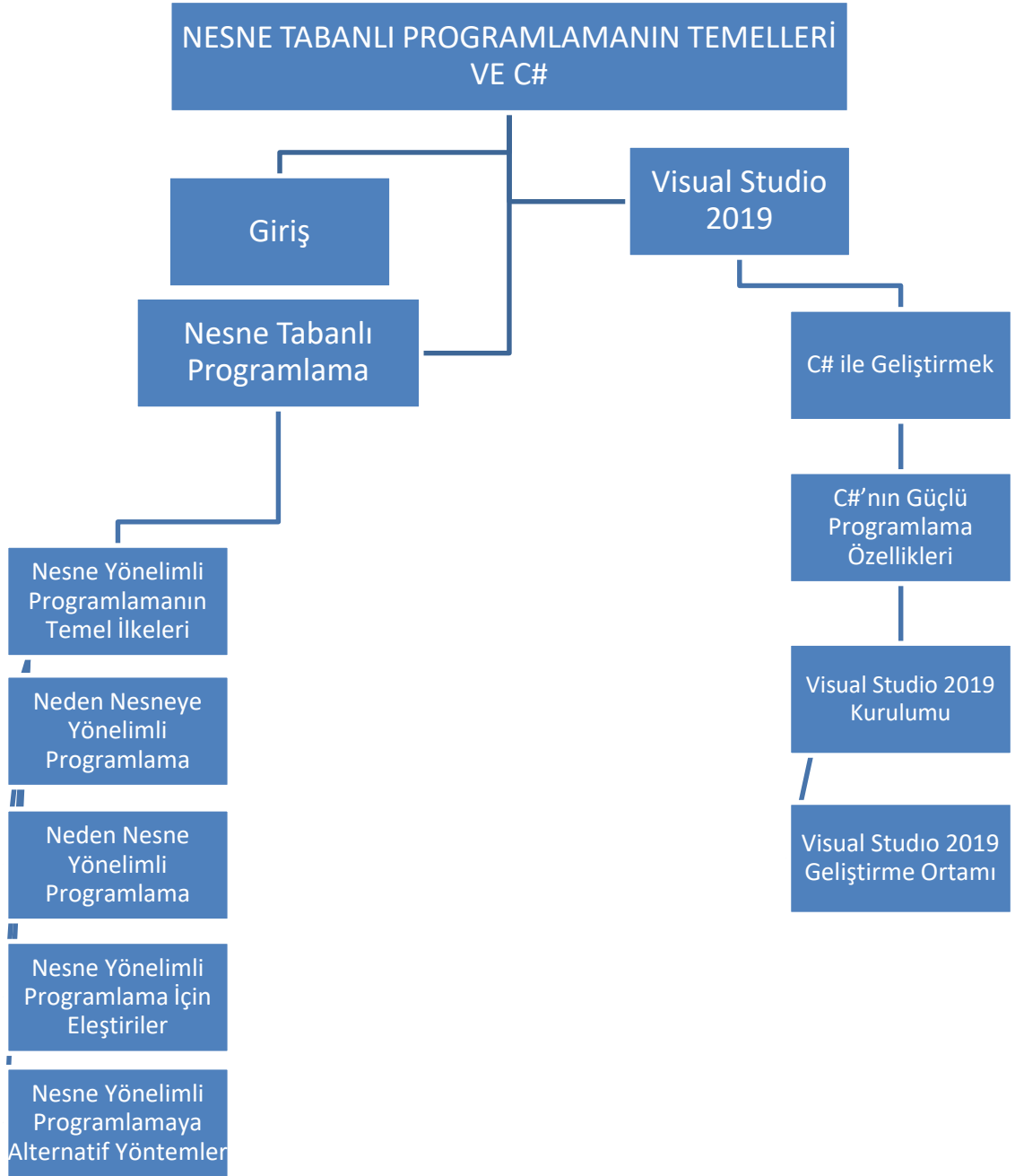
- Bu üniteyi çalıştıktan sonra;
  - Neden Nesne Yönelimli Programlama kullanılmakta, faydaları hakkında bilgi edebilecek,
  - Nesne Yönelimli Programlanın temel yapısı, özellikleri ve ilkeleri konusunda bilgi edebilecek,
  - Visual Studio .Net 2019 ve C# programlama diliyle ilk uygulamanızı geliştirebileceksiniz.



**Atatürk Üniversitesi**  
Açıköğretim Fakültesi

## NESNE TABANLI PROGRAMLAMA I Dr. Muhammet DAMAR

# ÜNİTE 1



## GİRİŞ

Hızla dönüşen ve gelişen bilişim teknolojileri beraberinde yazılım teknolojilerinin ve programlama tekniklerinin de gelişip dönüşmesine sebep olmuştur. Bu dönüşüm sürecinde öne çıkan gelişmelerden birisi de nesne tabanlı programlama tekniğidir. 1960'lı yıllarda başlayan **Nesne Tabanlı Programlama** paradigması, 1990'larda hız kazanarak birçok alanda yaygın hale gelmeye başlamıştır. Aslında 1990'lı yıllar sonrasında teknolojinin hızla gelişmesi ve bilgi sistemlerinin eğitim, bankacılık, finans ve sağlık gibi tüm sektörlerde yaygın olarak kullanılması yazılım geliştirme süreçlerinin gelişmesinde etkili olmuş ve önemli bir dönüşümü başlatmıştır.



Programlamayı öğrenmek uygulama yaparak ve hata ekranları ile boğuşarak gerçekleşmektedir!

Nesne tabanlı programlama, doksanlı yıllar ile başlayan ve günümüzde yoğun kullanılan bir programlama paradigmasıdır. Nesne tabanlı programlamanın temel felsefesi gerçek hayattaki nesnelerin birbirleriyle ilişkilerini temel alarak bu nesneleri özelliklerine göre sınıflandırmak suretiyle programlama dünyasında karşılaşılan karmaşık problemlere çözüm bulmaktır.

Nesne tabanlı programlama yaklaşımına bağlı kalınarak program geliştirilirken, çözülmesi istenen problemler çeşitli parçalara ayrılır; her bir parça problemin küçük bir çözümü olacak şekilde kodlanır. Sonuçta bütün bu parçaların bir araya getirilmesi yazılımı projesi tamamlanır.

Programlamayı öğrenmek uygulama yaparak ve hata ekranları ile boğuşarak gerçekleşmektedir! Aştığınız her hata ve kendi oluşturduğunuz ev ödevleri ve araştırmalarınız sizi sahada bir yere taşıyacaktır. Bu ders kapsamında size nesne tabanlı programlamanın giriş seviyesinde bilgiler sunulmaktadır. Bu bölümde ise .Net Framework ortamı ile tanışacak ve C# programlama dili aracılığı ile ilk programınızı yazabilmeniz için gerekli pratik bilgiyi bulacaksınız.

Bu ders için hazırlanan bölümler nesne tabanlı programlama ile programcılığa başlayan programcı adaylarının giriş seviyesinde kodlama yapabilmesini ve nesne tabanlı programlama konusunda geniş bir bakış açısına sahip olmalarını hedeflemektedir. Bölümler içindeki nesne tabanlı programlama mantığını eksiksiz yansıtmaları için C# programlama dili kullanılmıştır.

Ders kapsamında C#'ı seçmemizin birçok nedeni vardır. Birçok eğitim kurumu tarafından modern programlama dilini öğretmek amacıyla kullanılması, benzerlerine (C++ ve Java) oranla öğrenmesi daha kolay olması ve sektörde birçok büyük kurum tarafından ihtiyaçlarına yönelik yazılımları geliştirmek amacıyla tercih edilen bir dil olması, bu kitapta C#'a yer vermemizin en önemli sebepleridir.

C#, yazılım dünyasındaki dil standartlarını ortaya koyan en önemli kuruluşlardan olan ECMA ve ISO tarafından ortaya konulan standartlar ile Microsoft tarafından geliştirilmiş bir programlama dilidir. Microsoft, ilk zamanlar bu standartlar doğrultusunda C# programlama dilini geliştirse de C# 3.0 sürümü ile birlikte kendi standartlarını ortaya koymuş ve dili tamamen .NET Framework için hazır hale gelmiştir. Bu sayede herhangi bir kuruma bağlı kalınmaksızın

geliştiricilerin ihtiyaçlarının karşılanması için esnek ve sürekli gelişen bir programlama dili ortaya koyulması amaçlanmıştır.

Ders kapsamında nesne tabanlı programlama ile C# ile .Net Framework ortamı ile tanıştıktan sonra nesneye yönelimli programlama alışkanlığını, temel özellikleri kolayca öğrenebileceksiniz. C# bizim için ideal bir programlama aracı olmak ile beraber, bu ders sürecinde nesne tabanlı programlamanın temel unsurlarını, nesne tabanlı programlama yaklaşımını benimseyen tüm programlama dillerinde mevcut temel ilkeleri öğreneceksiniz.



Aştığınız her hata ve kendi oluşturduğunuz ev ödevleri ve araştırmalarınız sizi sahada bir yere taşıyacaktır.

## NESNE TABANLI PROGRAMLAMA

Bir programlama dili, çeşitli çıktıları üreten bir dizi talimattan oluşan resmi bir dildir. Programlama dilleri, bilgisayar programlamada algoritmaları uygulamak için kullanılır. Nesne tabanlı programlama, bir grup değişken ve fonksiyonun nesne adı verilen tek bir birimde birleştirilmesiyle düzenlenen bir tür programlama yaklaşımıdır. Tasarlanan nesneler, her nesnenin gruplandırıldığı sınıflar halinde düzenlenir.

Simülasyon dilinin kısaltması olan Simula, ilk nesne tabanlı programlama dili olarak kabul edilmektedir. Günümüzde ise nesneye yönelimli programlama yaklaşımını benimseyen birçok programlama dili kullanılmaktadır. Ancak bazı programlama dilleri nesneye yönelimli programlama paradigması ile diğerlerine göre daha iyi eşleşmektedir.

1970'lerde ve 1980'lerde, C, Pascal ve Fortran gibi prosedür odaklı programlama dilleri, iş odaklı yazılım sistemleri geliştirmek için yaygın olarak kullanılmaktadır. Prosedürel diller programı doğrusal bir şekilde düzenler; yukarıdan aşağıya doğru çalışırlar. Başka bir deyişle, program birbiri ardına çalışan bir dizi adımdır. Bu tür programlama, birkaç yüz kod satırından oluşan küçük programlar için iyi çalışırken, programlar büyüdükçe, yönetilmesi ve hatalarının ayıklanması zorlaştırmıştır.

Söz konusu bu problemleri aşmak için ise yapısal programlamaya gidilmiştir. Bu programcılık yaklaşımı ile var olan problemler kısmen aşılmasına rağmen programlar karmaşılaştıkça ve diğer sistemler için için girdikçe, yapısal programlamada da problemler ortaya çıkmıştır. Bunlar şu şekilde ifade edilebilir;

- Programların bakımının zorlaşması.
- Mevcut işlevselliği, sistemin tüm işlevselliğini olumsuz etkilemeden değiştirmek zordu.
- Yeni programlar pek çok zaman sıfırdan oluşturuldu ve bu aşamada önceki yatırım ve çabaların pek önemi olmadı.
- Programlama ekip çalışmasına uygun değildi ve programcılar, bir programın nasıl çalıştığını her yönüyle bilmek zorundaydı.
- İş modellerini program modellerine dönüştürmek zordu.
- Yapısal programlama tek başına iyi çalışırken diğer sistemlerle iyi entegre olamadı.



Ezberleme! Kodla,  
Kendine Problem Çıkar,  
Çözüm Üret, Kodla ve  
Çalıştır (!)

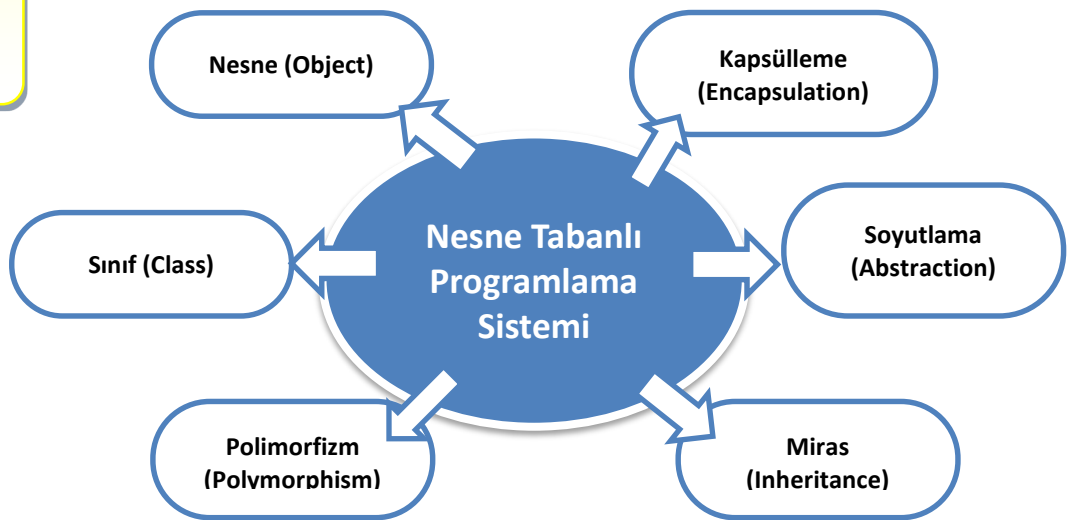
Örneğin saf diye adlandırılan nesneye yönelimli programlama dilleri (Ruby, Scala, JADE, Emerald), her şeyi nesne olarak ele almaktadır. Öte yandan diğer programlama dilleri nesneye yönelimli programlama için tasarlanmış olsalar da prosedürel süreçleri gerçekleştirebilmektedir. Bu tip programlama dillerine C#, F#, Python ve Java örnek gösterilebilir. Günümüzde ise öne çıkan nesneye yönelimli programlama dilleri C#, Python, Ruby, Go, C++ ve Java olarak sıralanabilir.

*Nesne tabanlı programlamanın yapısı veya yapı taşları şunları içerir;*

- Sınıflar (*Classes*), bireysel nesneler, nitelikler ve yöntemler için plan görevi gören kullanıcı tanımlı veri türleridir.
- Nesneler (*Objects*), özel olarak tanımlanmış verilerle oluşturulan bir sınıfın örnekleridir. Nesneler, gerçek dünyadaki nesnelere veya soyut bir varlığa karşılık gelebilir.
- Yöntemler (*Methods*), bir nesnenin davranışlarını tanımlayan bir sınıf içinde tanımlanan işlevlerdir. Sınıf tanımlarında yer alan her yöntem, bir örnek nesneye başvuru ile başlar. Ek olarak, bir nesnede bulunan alt rutinlere örnek yöntemler denir.
- Nitelikler (*Attributes*) sınıf şablonunda tanımlanır ve bir nesnenin durumunu temsil eder. Nesneler, nitelikler alanında depolanan verilere sahip olacaktır. Sınıf nitelikleri sınıfın kendisine aittir.

## Nesne Tabanlı Programlamanın Temel İlkeleri

Nesne Tabanlı Programlama Teorisinde dört temel özelliğin gerçekleştirilmesi zorunlu sayılmaktadır. Bu temel özelliklerden herhangi birini sağlamayan bir dil Nesne Tabanlı Programlama Dili olarak sayılmamaktadır. Bu dört özellik sırasıyla, Kapsülleme (Encapsulation), Soyutlama (Abstraction), Miras (Inheritance), Polimorfizm (Polymorphism), şeklindedir. Aşağıda şekil üzerinde gösterilmektedir.



Şekil 1.1. Nesne Tabanlı Programlama Sistemi Unsurları



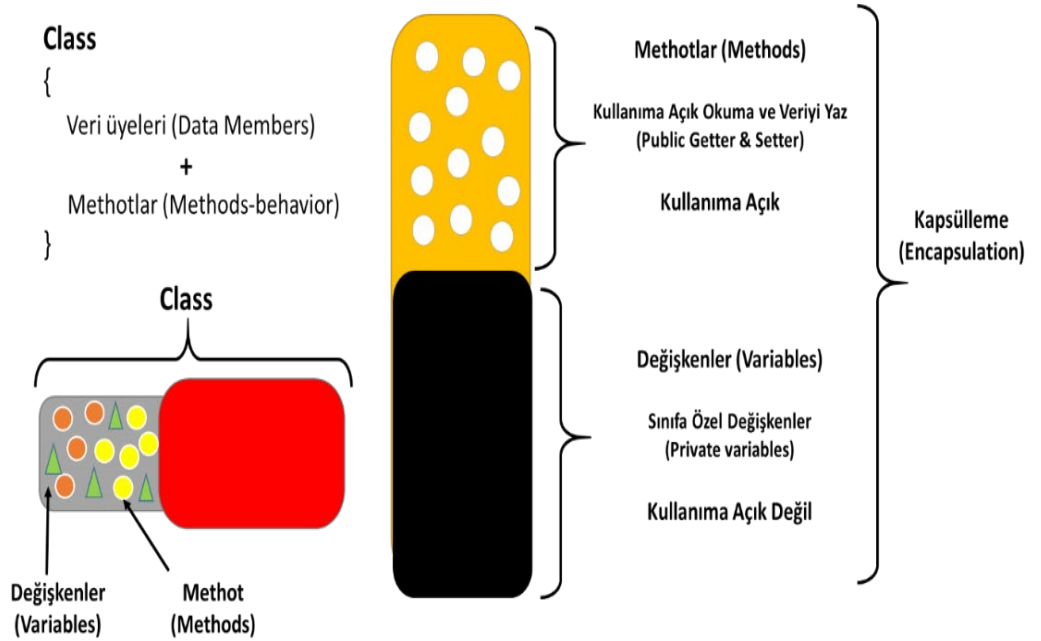
Kapsülleme, sınıfı  
oluşturan metot ve  
özelliklerin  
gerçekleştirme  
biçiminin, bu sınıfı  
kullanacak olan  
kullanıcılardan  
gizlenmiş olmasıdır.

## Kapsülleme veya Paketleme (Encapsulation)

Bu ilke, tüm önemli bilgilerin bir nesnenin içinde bulunduğunu ve yalnızca belirli bilgilerin açığa çıktığını belirtir. Kapsüllemenin anlamı, sınıfı oluşturan metot ve özelliklerin gerçekleştirme biçiminin, bu sınıfı kullanacak olan kullanıcılardan gizlenmiş olmasıdır. Bu özellik, dilin nesne kullanıcısından gereksiz uygulama ayrıntılarını saklar.



**Genel (Public ) Kullanım**  
Nitelendiğinde ilgili unsurlar herkese açık olur. Başka pakette olsa bile, program içindeki, her kod onlara erişebilir.



**Şekil 1.2.** Kapsülleme Sınıf Yapısı İçinde Method ve Değişken Görünümü

Genel tanımla kullanıcı tarafından verilerin, sınıfların ve metotların ne kadarının görüntülenebileceği ve değiştirilebileceğinin sınırlarının konulmasını sağlar. Public (herkese açık), private (özel) ve protected (koruma altında) olmak üzere üç adet access modifier'dan (erişim dönüştürücüsü) bahsedilebilir. Erişim belirteçleri (access modifier) sayesinde kapsülleme çok daha kolay yapılmaktadır.



**Özel (Private ) Kullanım**  
Private erişim belirteci, public belirteci karşıtı gibidir. Bir değer private olarak tanımlanması demek, o değişkene sadece kendi sınıfı içinden ulaşılacağı anlamına gelmektedir.

- **Public (Genel):** Nitelendiğinde ilgili unsurlar herkese açık olur. Başka pakette olsa bile, program içindeki, her kod onlara erişebilir. Bir değer public olarak belirtilmesi; o değer, kod içinde herhangi bir yerden erişilebilir durumda olmasını sağlamaktadır. Public erişim belirleyici tipinde hiçbir kısıtlama yoktur.
- **Private (Özel):** Bazı değişken, metot ya da sınıflara başka sınıftaki kodların erişmesini engellemek isteyebiliriz. Bunun için Private nitelemesini kullanırız. Private erişim belirteci, public belirteci karşıtı gibidir. Bir değer private olarak tanımlanması demek, o değişkene sadece kendi sınıfı içinden ulaşılacağı anlamına gelmektedir. Program içinde kesinlikle değiştirilmemesi gereken, önemli kodlarda kullanılmaktadır.
- **Protected (Korunumlu):** Bir sınıf içindeki değişken ve metotlara alt-sınıfların erişebilmesini, ama paket içindeki ya da program içindeki başka kodların erişmesini engellemek isteyebiliriz. Kod içinde bir değer protected olarak tanımlanması; o değere, değer bulunduğu sınıf ve



Kod içinde bir değerin protected olarak tanımlanması; o değere, değerin bulunduğu sınıf ve ondan türetilen diğer sınıflar içinden erişileceğini göstermektedir.

ondan türetilen diğer sınıflar içinden erişileceğini göstermektedir. Protected; bir anlamda, public ve private erişim belirleyicilerinin birleşimi olarak görülebilmektedir.

- **Internal (İçsel):** Internal olarak tanımlanan bir değer; aynı program içerisinde erişilebilir; fakat farklı bir program içerisinde erişilemez durumdadır. Program içerisinde herhangi bir kısıtlaması yoktur.
- **Protected Internal (İçsel Korunumlu):** Protected internal olarak tanımlanmış değer, tanımlandığı sınıfın içinden ve ondan türetilen sınıfların içinden erişilebilir durumdadır. Türetilen sınıfın aynı program içinde olmaması sorun teşkil etmez.

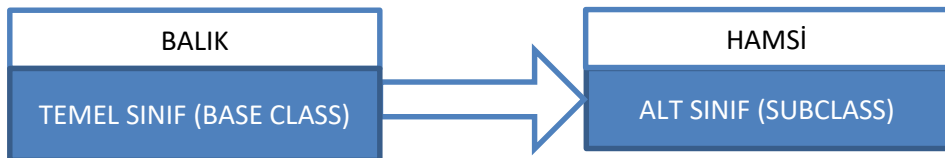
## Soyutlama (Abstraction)

Nesneler, yalnızca diğer nesnelerin kullanımıyla ilgili dahili mekanizmaları ortaya çıkararak gereksiz uygulama kodlarını gizler. Alt sınıfların ortak özelliklerini ve işlevlerini taşıyan ancak henüz bir nesnesi olmayan bir üst sınıf oluşturmak istenirse bir soyut (abstract) üst sınıf oluşturulur. Soyut sınıfın yöntemleri alt sınıfları tarafından üzerine yazılmak üzere şablon olarak tanımlanabilir veya soyut metod olarak oluşturulabilir. Türetilmiş sınıfın işlevselliği genişletilmiş olabilir. Bu konsept, geliştiricilerin zaman içinde ek değişiklikler veya eklemeler yapmasına yardımcı olmaktadır.

## Miras (Inheritance)

Bir sınıftan başka bir sınıf türetirken aralarında bir alt-üst ilişkisi oluşturmayı ve bu sınıflar üzerinde ortak metodlar ve özellikler kullanılmasını sağlayan bir mekanizmadır. Hali hazırda var olan sınıfların üzerine başka sınıfların inşa edilmesini sağlar. Sınıflar, diğer sınıflardan gelen kodları yeniden kullanabilir.

Nesneye yönelik programlamada, bir nesne, genellikle bir nesne sınıfına ait bir örnektir (instance). Örneğin, Balık sınıfının örneği Hamsi veya Levrek olabilir (Şekil 1.3). Bir nesne sınıfından alt sınıflar (subclasses) oluşturulabiliyorsa, türetme özelliği (derivation) var demektir. Kendisinden alt sınıf (subclass veya child class) üretilen sınıfa, temel sınıf (base class) veya süper sınıf (super class) veya ana sınıf (parent class) adı verilir.



Şekil 1.3. Temel Sınıf ve Alt Sınıf

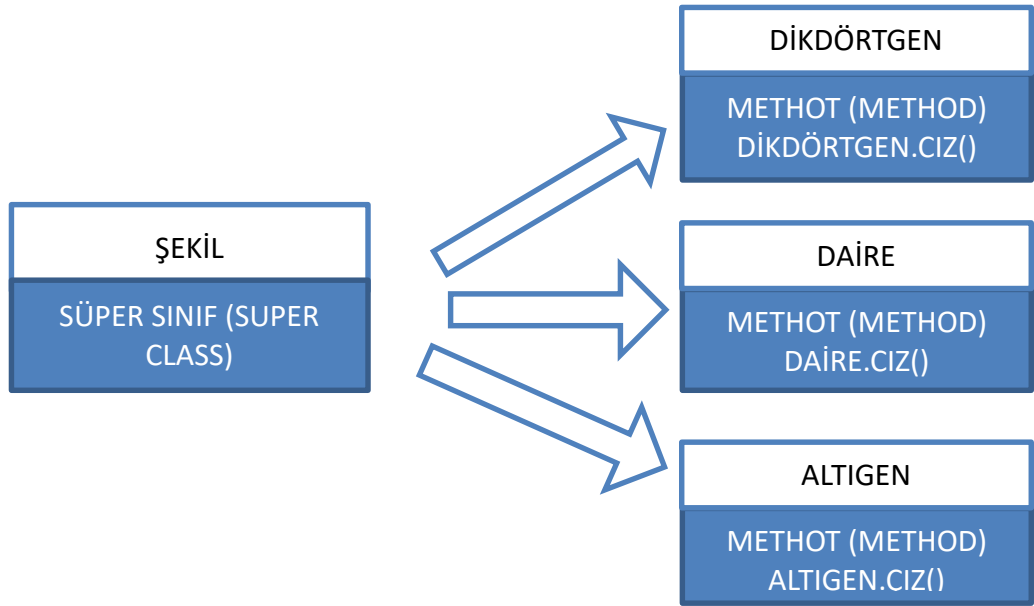
Alt sınıfın nesneleri, türetildikleri temel sınıfa ait özellikleri alıyorsa, burada miras alma (inheritance) özelliği vardır denir. Bu anlamda, miras alma özellikli bir nesne tabanlı programlama dilinde, bir nesne sınıfından türetilen alt nesne sınıfına ait nesneler, üst sınıfın özelliklerini (properties) ve metodlarını (methods) aynen alırlar.

## Polimorfizm (Polymorphism)

Nesneler, davranışları paylaşmak üzere tasarlanmıştır ve birden fazla biçim alabilirler. Polymorphism yani çok biçimlilik Nesneye Yönelik Programlamada programlama dilinin farklı tip verileri ve sınıfları farklı şekilde işleme yeteneğini belirten özelliğidir. Daha açık ifade etmek gerekirse; metotları ve türetilmiş sınıfları yeniden tanımlama yeteneğidir. Farklı nesnelerin, aynı mesaja (olaya ya da uyarıma) farklı şekillerde cevap verebilme yeteneği olarak da ifade edilebilir. Örneğin Şekil 1.4 üzerinde **ŞEKİL** süper sınıfımız (*super class*) üzerinden, dikdörtgen, daire, altıgen adlı üç alt sınıf türetebiliriz. Bu alt sınıfların her biri kendi örneklerini çizmek için **ÇİZ** adlı bir metoda sahip olabilir. Fakat her bir alt sınıf için aynı metot, uygulamada dikdörtgen, daire veya altıgen çizecektir.



Polimorfizm yani çok biçimlilik Nesneye Yönelik Programlamada programlama dilinin farklı tip verileri ve sınıfları farklı şekilde işleme yeteneğini belirten özelliğidir.



Şekil 1.4. Polimorfizm Örneği

## Neden Nesne Tabanlı Programlama

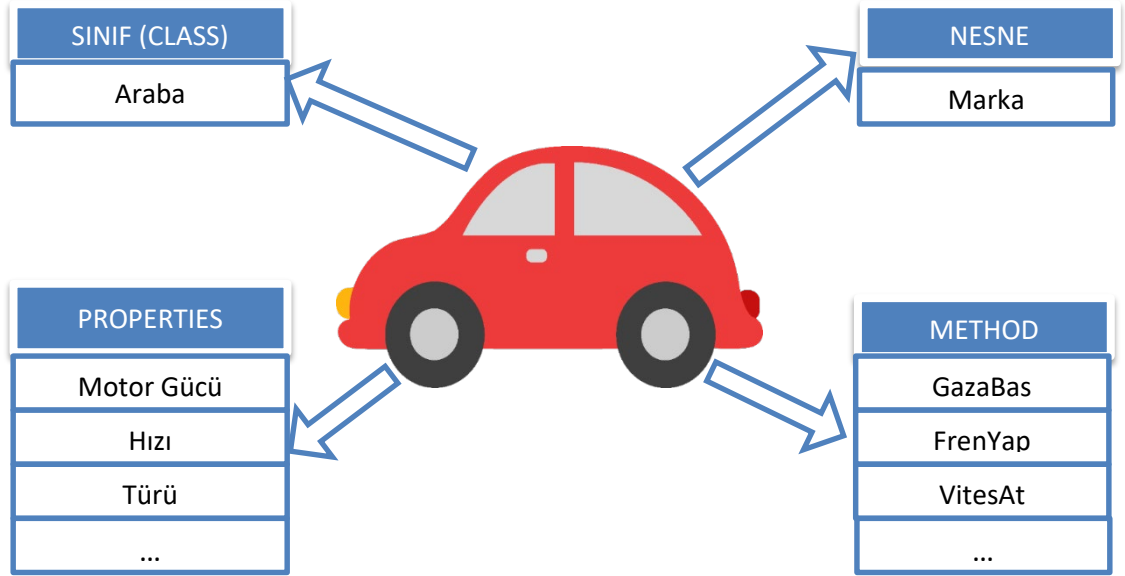
Yazılım geliştirme süreçlerinde nesneleri kullanmak ve tasarlanan sistemi veri modeline ve fonksiyonlarına göre nesneler halinde tasarlamak bize pek çok fayda sağlayacaktır. Nesne tabanlı programlama mantığı, modülerlik, tekrar kullanılabilirlik, üretkenlik, güvenlik, esneklik, bakım gibi pek çok avantaj sağlar. Bu avantajlar kısaca şu şekildedir;



Modülerlik, tekrar kullanılabilirlik, üretkenlik, güvenlik, esneklik, bakım gibi pek çok fayda ve kolaylığa nesne yönelimli programlama sahiptir.

- **Modülerlik (Modularity):** Kapsülleme, nesnelerin bağımsız olmasını sağlayarak sorun gidermeyi ve işbirliğine dayalı geliştirmeyi kolaylaştırır.
- **Tekrar Kullanılabilirlik (Reusability):** Kod, kalıtım yoluyla yeniden kullanılabilir, yani bir ekibin aynı kodu birden çok kez yazması gerekmez. Ayrıca, nesnelerin üretildiği sınıflar, ortam içinde her uygulama geliştiricinin kullanımına açık olduğu için ortak bir kütüphane oluşturulmuştur.





Şekil 1.5. Nesne Tabanlı Programlama ve Gerçek Hayat

- **Üretkenlik (Productivity):** Programcılar, birden çok kitaplık ve yeniden kullanılabilir kod kullanarak yeni programları daha hızlı oluşturabilir.
- **Kolayca yükseltilebilir ve ölçeklenebilir (Easily upgradable and scalable):** Programcılar, sistem işlevlerini bağımsız olarak uygulayabilir.
- **Genişletilebilirlik (Extensibility):** Nesne Tabanlı Programlamada mevcut bir sınıfa yeni özellik ve metotlar ekleyerek artan işlevsellik sağlamak çok kolaydır.
- **Arayüz tanımlama (Interface descriptions):** Nesnelerin birbiriyle iletişimi için kullanılan mesaj aktarma teknikleri sayesinde dış sistemlerin tanımlanması basittir.
- **Güvenlik (Security):** Kapsülleme ve soyutlama kullanılarak karmaşık kodlar gizlenir, yazılım bakımı daha kolaydır ve internet protokolleri korunur.
- **Esneklik (Flexibility):** Polimorfizm, tek bir fonksiyonun yerleştirildiği sınıfa uyum sağlamasına olanak tanır. Aynı arayüzden farklı nesneler de geçebilir.
- **Yazılımın bakımı (Maintenance):** Birbirine benzer özellikler taşıyan veya aynı amaca hizmet eden sınıflar isim alanı (namespace) veya paket (package) içinde aynı konu ile ilişkili sınıflar bir araya getirilmektedir. Bu sayede yazılımın bakımı, sürdürülebilirliği ve yeni eklemeler kolayca eklenebilir.



En büyük eleştiri nedeni, nesneye yönelik programlamanın yazılım geliştirmenin veri bileşenini aşırı odaklanması ve hesaplama veya algoritmalara yeterince odaklanmamasıdır.

## Nesne Tabanlı Programlama İçin Eleştiriler

Nesne tabanlı programlama modeli, geliştiriciler tarafından kimi zaman eleştirilmiştir. En büyük eleştiri nedeni, nesneye yönelik programlamanın yazılım geliştirmenin veri bileşenini aşırı odaklanması ve hesaplama veya algoritmalara yeterince odaklanmamasıdır.

Ek olarak Nesneye Yönelimli Programlama kodunun yazılması daha karmaşık olabilir ve derlenmesi daha uzun sürebilir. Özellikle küçük projeler veya

prosedürel programlamanın ihtiyaç duyulduğu yerlerde dezavantaj yaratabilir. Bu durum proje yetkinliğinizi arttıkça aşılabilecek bir noktadır.

## Nesne Tabanlı Programlamaya Alternatif Yöntemler

- *Fonksiyonel programlama (Functional programming)*: Bu, telekomünikasyon ve hataya dayanıklı sistemler için kullanılan Erlang ve Scala gibi dilleri içerir.
- *Yapılandırılmış veya modüler programlama (Structured or modular programming)*: PHP programlama dili burada örnek verilebilir.
- *Zorunlu programlama (Imperative programming)*: Nesneye yönelimli programlamanın bu alternatifi, modellerden ziyade işleve odaklanır ve C++, C# ve Java programlama dillerini içerir.
- *Bildirimsel programlama (Declarative programming)*: Bu programlama yöntemi, görevin veya istenen sonucun ne olduğuna ilişkin ifadeleri içerir, ancak nasıl elde edileceğini içermez. Diller arasında Prolog ve Lisp bulunur.
- *Mantıksal programlama (Logical programming)*: Çoğunlukla biçimsel mantığa dayanan ve Prolog gibi dilleri kullanan bu yöntem, bir problem alanı hakkında gerçekleri veya kuralları ifade eden bir dizi cümle içerir. Kural tabanlı mantıksal sorgulardan yararlanabilecek görevlere odaklanır.

*Nesneye Yönelimli Programlama paradigması*; büyük projelerin geliştirilmesi için ideal bir paradigmadır. Program sınıfların (*Class*) ve nesnelerin (*Object*), güçlü veri güvenliği sağlar. Ayrıca karmaşık projelerde dahi kodun okunabilirliği ve bakımının kolay olması, programcının üretkenliğini yükseltir, yeni ver türlerinin program içinde kullanılabilmesini sağlar.

Çoğu gelişmiş programlama dili, farklı programlama yöntemleri için kullanılabildikleri için geliştiricilerin modelleri birleştirmesine olanak tanır. Örneğin C# veya JavaScript, Nesne Yönelimli Programlama ve fonksiyonel programlama için kullanılabilir.

Buradaki programlama paradigmalarında aslında en iyi ya da tek doğru yoktur. Her yöntemin programcı tarafından kullanılabilir iyi yönleri vardır. Bu programlama tecrübeniz ve projenin ölçeği ve ihtiyaçlarına bağlı olarak sizin tercihinize kalmıştır. Bununla birlikte kadar çok deneyim kazanırsanız, üzerinde çalıştığınız belirli bir projeye hangi kod stilinin en uygun olabileceğini o kadar iyi belirleyebilir, hatta birden fazla paradigmayı örtüştürüp de kullanabilirsiniz.

## VISUAL STUDIO 2019

### C# ile Geliştirmek

C# ile programlamak için C# programları oluşturmanın bir yoluna ihtiyacımız olacaktır. Bunu bir komut satırı derleyicisi, Visual Studio veya bir programlama düzenleyicisi için bir C# paketi ile yapabiliriz ki biz uygulamalarımız için Visual Studio 2019 geliştirme ortamı tercih edilmiştir.

C#, Microsoft tarafından .NET platformuyla birlikte oluşturulan ve geliştirilen modern, nesne yönelimli, genel amaçlı bir programlama dilidir. C# ve .NET platformunda geliştirilmiş çok çeşitli yazılımlar vardır: ofis uygulamaları, web uygulamaları, web siteleri, masaüstü uygulamaları, mobil uygulamalar, oyunlar ve diğerleri.

C#, Microsoft tarafından geliştirilen ve Avrupa Bilgisayar Üreticileri Birliği (ECMA) ve Uluslararası Standartlar Organizasyonu (ISO) tarafından onaylanan modern, genel amaçlı, nesne yönelimli bir programlama dilidir. C#, Anders Hejlsberg ve ekibi tarafından .Net (Dot Net) Framework'ün geliştirilmesi sırasında geliştirilmiştir. C# Programlama dilinde sınıf tanımlamak class kelimesi kullanılır.

C#, farklı bilgisayar platformları ve mimarilerinde çeşitli üst düzey dillerin kullanımına izin veren yürütülebilir kod ve çalışma zamanı ortamından oluşan Ortak Dil Altyapısı (Common Language Infrastructure) için tasarlanmıştır. Aşağıdaki nedenler C#'ı yaygın olarak kullanılan bir profesyonel dil yapar;

- Modern, genel amaçlı bir programlama dilidir.
- Nesne yönelimlidir.
- Bileşen odaklıdır.
- Öğrenmesi kolaydır.
- Yapılandırılmış bir dildir.
- Verimli programlar üretir.
- Farklı bilgisayar platformlarında derlenebilir.
- .Net Framework'ün bir parçasıdır.

C#, Java ve C++'a ve bir dereceye kadar Delphi, VB.NET ve C gibi dillere benzeyen üst düzey bir dildir. C#, nesne yönelimli paradigmayı tamamiyle benimsemektedir. Yöntemleri içeren sınıflardaki bir dizi tanımdan oluşurlar ve yöntemler, program mantığını, yani bilgisayarın yürüttüğü talimatları içerir.

Visual Studio, C# programları geliştirmek için harika bir ortam sağlamaktadır. Maliyet bir sorunsuzsa, Visual Studio Express ürünü çoğu geliştirme senaryosunu içinde barındırmakta ve bu kitap kapsamında işlenecek konular için yeterlidir. Ayrıca SharpDevelop IDE'yi de içinde barındırmaktadır. Özetle her ikisi de ücretsiz olarak mevcuttur.

Microsoft dışı platformlar hedeflediğinizde de seçenekleriniz elbette ki mevcuttur. Mono projesi bu amacı gerçekleştirmekte, Linux, iOS ve Android'i hedefleyebilen bir C# ortamı sağlamaktadır.

C# dili, üzerinde yürütüldüğü Common Language Runtime (CLR) adı verilen özel bir ortamla birlikte dağıtılır. Bu ortam, temel işlevsellik, derleyiciler, hata ayıklayıcılar ve diğer geliştirme araçları sağlayan bir standart kitaplık paketi olan CLR'yi içeren .NET Framework platformunun bir parçasıdır.

.NET Runtime, bellek ayırma, güvenlik, tür güvenliği, özel durum işleme ve diğer birçok düşük düzeyli sorunu yönetmektedir. .NET Runtime'ın çok işlemcili sunuculardan akıllı telefonlara ve mikrodenetleyicilere kadar her şey üzerinde çalışan birkaç farklı çeşidi bulunmaktadır. Yararlı görevleri gerçekleştirmek için C#

kodunuz .NET Base Class Library'deki (BCL) kodu kullanacaktır. BCL, birçok program için faydalı olabilecek sınıfları içerir ve aşağıdakiler için destek içerir;

- Ağ işlemlerini gerçekleştirmek
- G/Ç işlemlerini gerçekleştirme
- Güvenliği yönetme
- Küreselleştirici programlar
- Metni değiştirme
- Bir veritabanına erişme
- XML'i manipüle etme
- Olay günlüğü, izleme ve diğer tanılama işlemleriyle etkileşim kurma
- Yönetilmeyen kod kullanma
- Dinamik olarak kod oluşturma ve çağırma

Küreselleştirici programlar ifadesinde geçen küreselleşme kavramı, geliştiricilerin dünyanın farklı bölgelerinde kullanılabilecek uygulamalar yazmasına yardımcı olur. Uygulamanın birden çok dili, farklı tarih ve sayı biçimlerini vb. desteklemesine yardımcı olur. BCL'nin üzerinde, aşağıdakiler de dahil olmak üzere belirli uygulama veya hizmet türleri oluşturmayı hedefleyen özel kitaplıklar bulunur;

- Konsol uygulamaları,
- Windows Forms veya Windows Presentation Foundation (WPF) kullanan Windows GUI uygulamaları,
- ASP.NET (web) uygulamaları,
- Windows Hizmetleri,
- Windows Communication Foundation (WCF) kullanan hizmet odaklı uygulamalar,
- İş akışı etkin uygulamalar, Windows Workflow Foundation (WF),
- Windows 8 uygulamaları ve Windows Mobil Telefon uygulamaları.

Günümüzde C# en popüler programlama dillerinden biridir. Dünya çapında milyonlarca geliştirici tarafından kullanılmaktadır. C#, Microsoft tarafından uygulamaların geliştirilmesi ve yürütülmesine yönelik modern platformlarının bir parçası olarak geliştirildiğinden, .NET Framework, dil Microsoft odaklı şirketler, kuruluşlar ve bireysel geliştiriciler arasında geniş çapta yayılmıştır. C# dili ve .NET platformu tamamen Microsoft tarafından korunur ve yönetilir ve üçüncü taraflara açık değildir. Bu nedenle IBM, Oracle ve SAP gibi diğer tüm büyük yazılım şirketleri, çözümlerini Java platformuna dayandırmakta ve kendi yazılım ürünlerini geliştirmek için Java'yı birincil dil olarak kullanmaktadır. Bu kimi zaman C# geliştiricileri için olumsuz durumlar oluşturabilir. Fakat bu durum küresel rekabet ortamında olağan bir durumdur.

## C#'ın Güçlü Programlama Özellikleri

C# yapıları geleneksel üst düzey dilleri yakından takip etse de, C ve C++ dir. Java ile güçlü bir benzerliği vardır, onu dünya çapında birçok programcıya sevdiren

çok sayıda güçlü programlama özelliğine sahiptir. C#'ın birkaç önemli özelliğinin listesi aşağıdadır;

- Boolean Koşulları,
- Otomatik Çöp Toplama,
- Standart Kütüphane,
- Montaj Versiyonu,
- Özellikler ve Olaylar,
- Delegeler ve Etkinlik Yönetimi,
- Kullanımı kolay Jenerikler,
- İndeksleyiciler Koşullu Derleme,
- Basit Çoklu Okuma,
- LINQ ve Lambda İfadeleri,
- Windows ile entegrasyon yapılabilmesi.

## Visual Studio 2019 Kurulumu

Nesne tabanlı programlar yapmak ve bu programlar içerisinde Visual C# dilini daha kolay kullanmak amacıyla Studio.NET yazılım geliştirme ortamı kullanılmaktadır. Sürüm 16.11 versiyonu kod geliştirme ve anlatımda kullanılacaktır.

Bu bölümde şu ana kadar geliştirilmiş en son ve en kapsamlı sürümü Visual Studio 2019 .Net versiyonunun kurulumu anlatılacaktır. Bu sürümde Windows işletim sistemi üzerinde çalışan pek çok uygulamayı geliştirmek mümkündür. Visual Studio ve .NET'i kullanarak masaüstü, web, mobil, oyun ve nesnelerin interneti için uygulamalar geliştirebilirsiniz. .NET uygulamalarını C#, F # veya Visual Basic dilinde yazabilirsiniz. Örneğin; Windows Azure, Office 365 ile Cloud (bulut) teknolojisi ile çalışacak ofis uygulamaları geliştirilebilir.

Visual Studio 2019 kurulumu sırasında öncelikle bilgisayarımızın uygun sistem gereksinimlerine sahip olup olmadığı kontrol edilmelidir. Aşağıda şekil üzerinde bu gereksinimler desteklenen işletim sistemleri, donanım gereksinimleri ve son olarak desteklenen dillerdir. Bunlar aşağıda açıklanmaktadır.



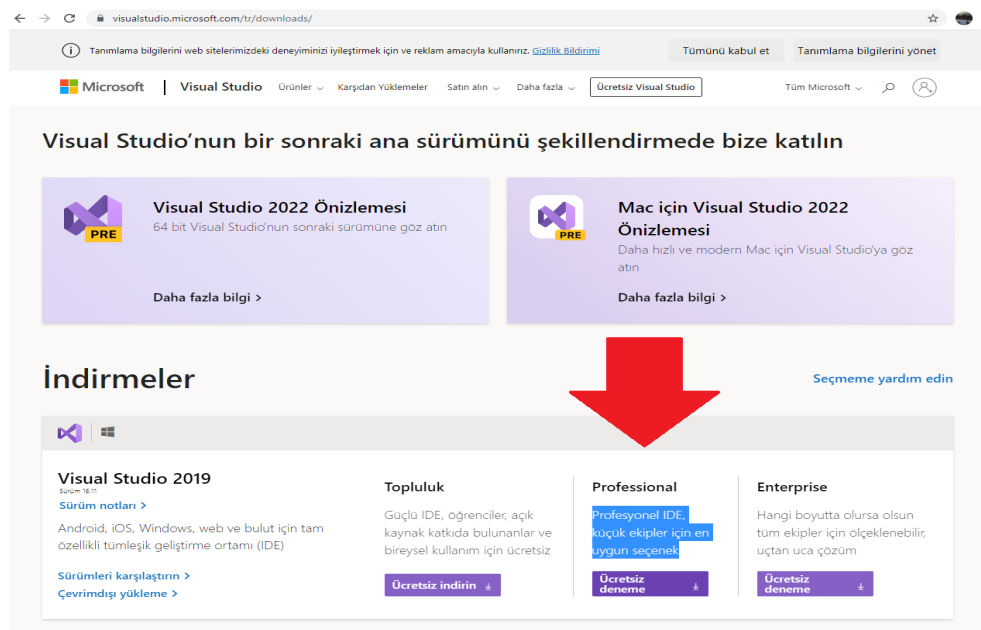
Visual Studio 2019 Programın kurulabilmesi ve verimli çalışabilmesi için sistem gereksinimleri, Visual Studio Enterprise 2019, Visual Studio Professional 2019, Visual Studio Community 2019, Visual Studio Team Foundation Server Office 2019 yazılımlarını destekler.

Tablo 1.1. Visual Studio 2019 Sistem Gereksinimleri

Desteklenen İşletim Sistemleri	Donanım
<p>Visual Studio 2019, aşağıdaki işletim sistemlerinde yüklenir ve çalıştırılır (64 bit önerilir; ARM desteklenmez):</p> <ul style="list-style-type: none"> <li>Windows 10 veya üzeri: Home, Professional, Education ve Enterprise (LTSC ve S desteklenmez)</li> <li>Windows Server 2019: Standard ve Datacenter</li> <li>Windows Server 2016: Standard ve Datacenter</li> <li>Windows 8.1: Core, Professional ve Enterprise</li> <li>Windows Server 2012 R2: Essentials, Standard, Datacenter</li> <li>Windows 7 SP1 (en son Windows güncelleştirmeleri ile): Home Premium, Professional, Enterprise, Ultimate</li> </ul>	<ul style="list-style-type: none"> <li>1,8 GHz veya daha hızlı işlemci. Dört çekirdekli veya daha iyisi önerilir.</li> <li>2 GB RAM; 8 GB RAM önerilir (sanal makinede çalıştırılıyorsa en az 2,5 GB)</li> <li>Sabit disk alanı: Yüklenen özelliklere bağlı olarak minimum 800 MB, (210 GB'a kadar) kullanılabilir alan; normal yüklemeler için 20-50 GB boş alan gerekir.</li> <li>Sabit disk hızı: Performansı artırmak için Windows ve Visual Studio'yu bir katı hal sürücüsüne (SSD) yükleyin.</li> <li>En az 720p (1280x720) ekran çözünürlüğü destekleyen ekran kartı; Visual Studio WXGA (1366x768) ve üzeri çözünürlükte en iyi şekilde çalışır.</li> </ul>
Desteklenen Diller	
<p>Visual Studio İngilizce, Çince (Basitleştirilmiş), Çince (Geleneksel), Çekçe, Fransızca, Almanca, İtalyanca, Japonca, Korece, Lehçe, Portekizce (Brezilya), Rusça, İspanyolca ve Türkçe dillerinde kullanılabilir.</p>	

Visual Studio 2019 programın kurulabilmesi ve verimli çalışabilmesi için sistem gereksinimleri aşağıda verilmiştir. Bu sistem gereksinimleri, Visual Studio Enterprise 2019, Visual Studio Professional 2019, Visual Studio Community 2019, Visual Studio Team Foundation Server Office 2019 yazılımlarını destekler.

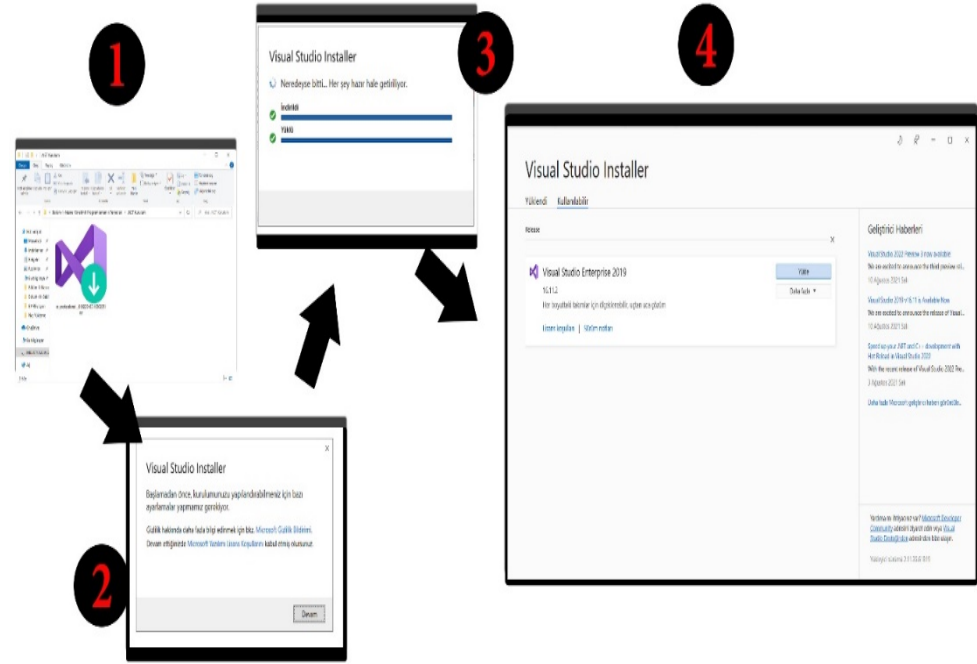
Aşağıda şekil üzerinde de görüleceği gibi Visual Studio 2019 farklı hizmetler için farklı özelliklere sahiptir. Üç farklı versiyon (**Topluluk**, **Professional** ve **Enterprise seçenekleri**) Visual Studio web sitesinden (<https://visualstudio.microsoft.com/tr/downloads/>) indirilebilir.



Şekil 1.6. Visual Studio 2019 Programını İndirilebileceğimiz Web Sitesi Bilgileri

Bunlardan **Topluluk**, güçlü bir IDE'ye sahip öğrenci ve açık kaynak kod geliştiricileri ve bireysel kullanıma uygun sürümdür ve ücretsiz indirilebilir. **Profesyonel** (Professional: daha çok küçük ekipler) ve **Kurumsal** (Enterprise: tüm ekip türleri için ölçeklenebilir, uçtan uca çözüm sağlamaktadır.) ise ücretsiz deneme sürümü ile sunulmaktadır.

Aşağıda sırasıyla Visual Studio 2019 programı Micsosoft internet sitesinden indirilerek yüklenmeye başlanabilir. Bu adımlar, dört basamaklı olarak aşağıda şekiller üzerinde ifade edilmektedir.



**Şekil 1.7.** Visual Studio 2019 Programı İndirme ve Yüklemeye Başlama Adımları

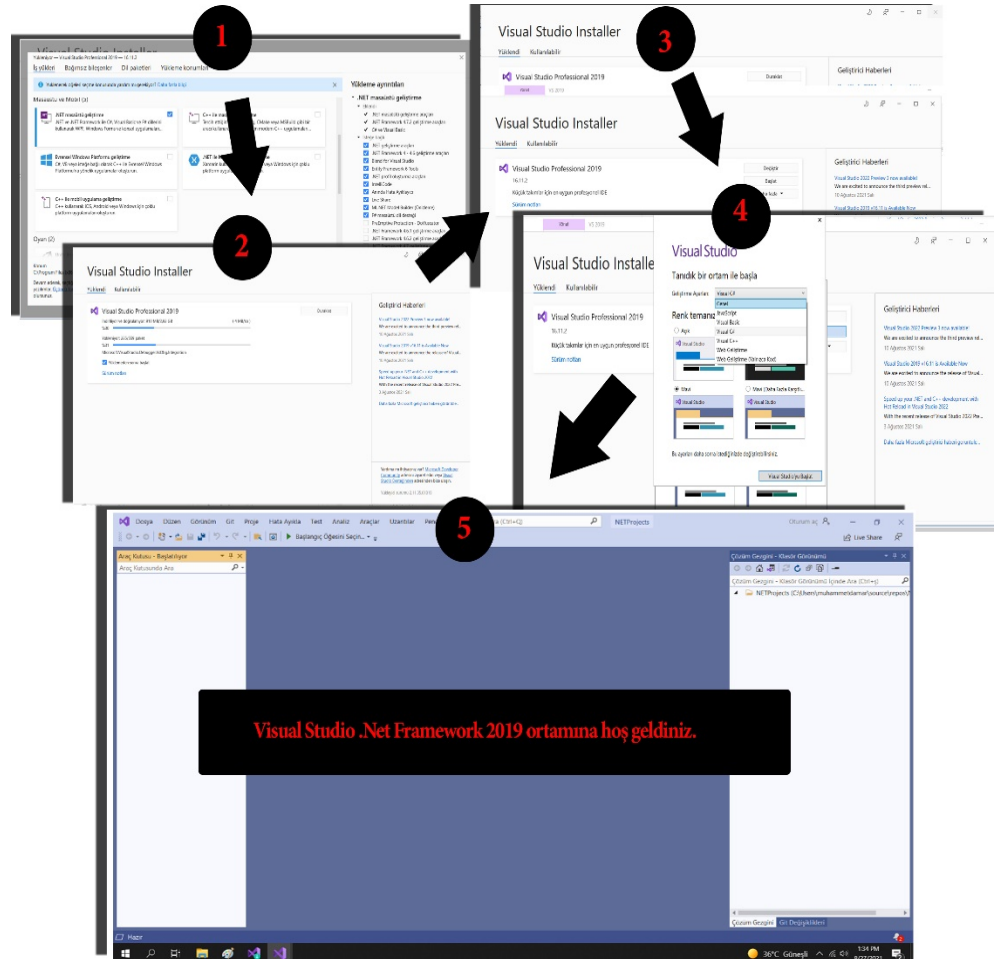
Bu adımda kurulumumuzda hangi yönde program geliştireceksek bu modülleri sırasıyla tercih edip sonraki etapta ekranın en aşağısında yer alan “Yükle” butonuna tıklayarak kurulum işini sürdürüyoruz. Burada sistem gereksinimleri bölümünde de ifade ettiğimiz gibi bilgisayarımız uygun konfigürasyonlara sahip olmalı ve bilgisayarımızın diskinde yeterli yerimizin bulunması gerekmektedir. Örneğin bizim tercihlerimiz .Net Konsol, masaüstü ve web uygulamaları paketlerini içermesine rağmen 11,82 GB bir kapasitenin hard diskten tahsis edildiği görülebilir.



Bizim tercihlerimiz .Net Konsol, masaüstü ve web uygulamaları paketlerini içermesine rağmen 11,82 GB bir kapasitenin hard diskten tahsis edildiği görülebilir.

Sonraki etapta adım adım kurulum işlemimiz aşağıdaki şekilde sürmektedir. Unutulmaması gereken nokta sadece 11,82 GB yer ile hareket edilmemesidir. Bilgisayarımızda .Net Studio 2019 kurulumu için en az 20 GB yeriniz muhakkak olmalıdır. Bu sayede programı geliştirme sürecinizde daha az sıkıntı yaşayacaksınız. Aşağıda tüm bu adımların ekran görüntüleri beş aşamada gösterilmektedir.





Şekil 1.8. Visual Studio .Net Framework 2019 Beş Adımda Kurulum

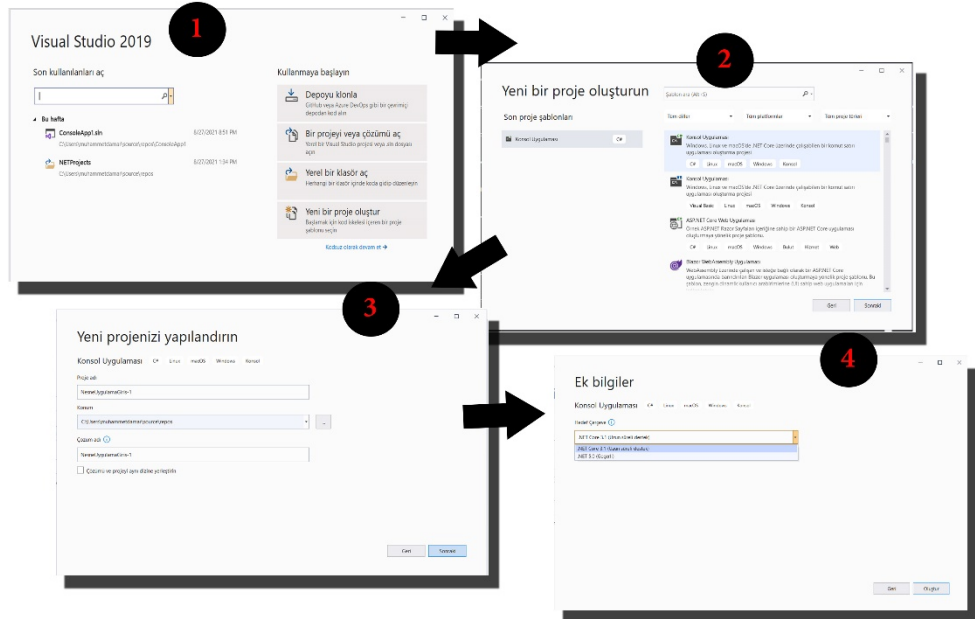
Kurulum tamamlandığında başlat diyerek veya Başlat → Programlar → Visual Studio 2019 seçenekleri ile programınızı çalıştırabilirsiniz. Kurulum tamamlandıktan sonra size hangi programlama dili tercihi ile başlamak istediğiniz sorulacaktır. Siz C# programlama dilini seçerek program yazabilmek için .Net Framework Visual Studio 2019 ortamını artık kullanmaya başlayabilirsiniz.

Visual Studio .Net Framework 2019 ortamına hoş geldiniz. Bundan sonrası bu ders içeriği yanında sizin kendinizce oluşturduğunuz güncel hayata dair problemlerin Visual Studio .Net Framework 2019 ortamında sizin tarafınızdan çözüm bulması ile yazılım geliştirmeye devam edebilirsiniz..

### Visual Studio 2019 Geliştirme Ortamı

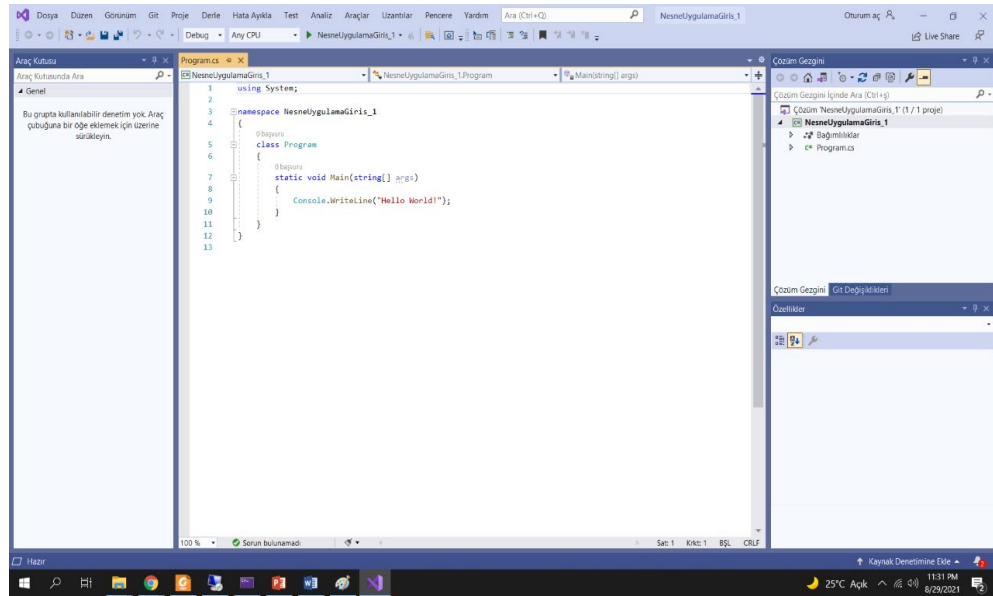
Visual Studio 2019 programının bilgisayara kurulum işlemi tamamlandıktan sonra, Başlat menüsü içinden Programlar bölümü altından Visual Studio 2019 programı seçilerek program çalıştırılır. Program çalıştırıldığında Şekil 1.9'da görüntülenen Visual Studio 2019 ortamında *Yeni Bir Proje Oluşturma* dört adımda gösterilmektedir. Bu ekranda da görüldüğü gibi daha önce açtığımız uygulamaları açabileceğimiz gibi yeni bir projeyi de burada açabiliriz. Bu adım bizim birinci adımımızdır.





Şekil 1.9. Visual Studio 2019 Ortamında Dört Adımda Yeni Bir Proje Oluşturma

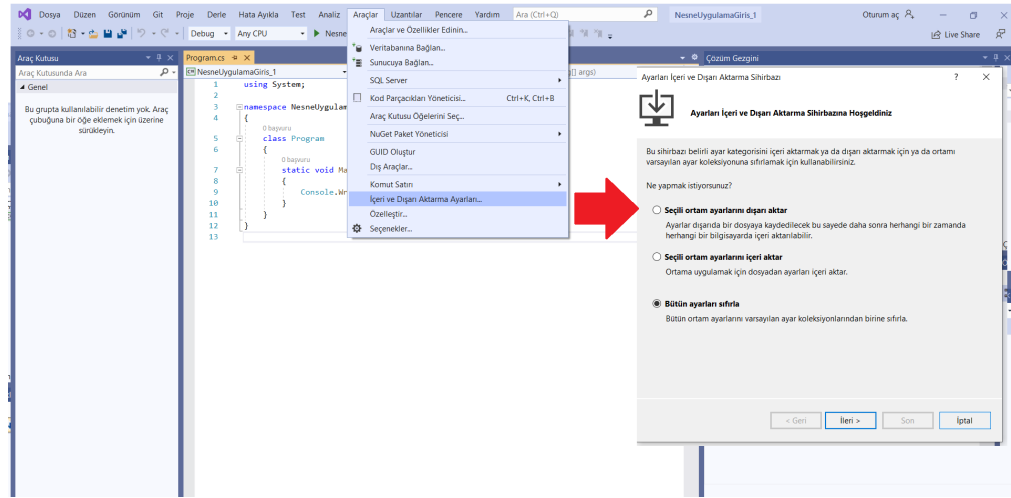
İkinci adımda “*Yeni bir proje oluştur*” menüsü tercihi edilerek ikinci ekranımız açılır. Buradan amacımıza uygun projeyi seçebiliriz. Biz ilk uygulamamızı *Konsol Uygulaması* olarak açmayı tercih ediyoruz. Proje türümüzü Visual Studio 2019 Konsol uygulaması olarak seçildikten sonra Proje Adı, Çözüm Adı ve Dosyanın kaydedileceği konum belirlenir. Konsol uygulamamız Visual Studio 2019.Net Framework .Net 5.0 veya .Net Core 3.1 ortamından birisi tercih edilmektedir. Biz proje uygulamalarımızda .Net Core 3.1 seçeneğini tercih ederek bir sonraki adıma geçebiliriz. Açılan son penceremiz Konsol uygulamamızı istediğimiz gibi şekillendirebileceğimiz ve kodlayabileceğimiz ekrandır. Karşımıza aşağıdaki “*Hello World*” mesajı ile C# programlama dilini kodlamaya başlayabileceğimiz .Net ortamı karşımıza çıkmaktadır (Şekil 1.10).



Şekil 1.10. Visual Studio 2019 Ortamında Yeni Bir Proje Oluşturma - V

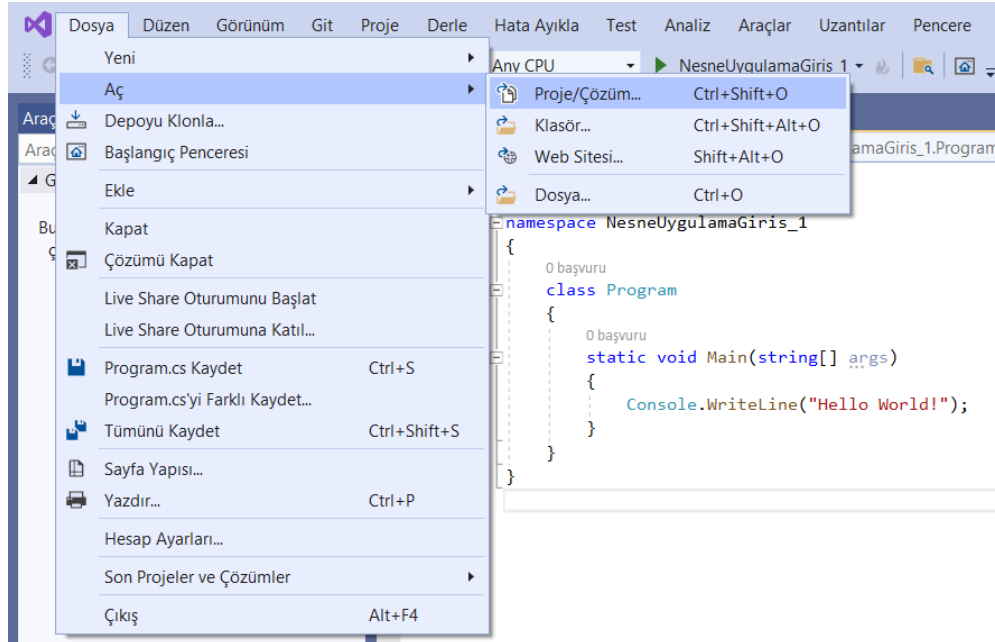
İlerleyen zamanlarda, Visual Studio .Net ortamının Visual C# programlama dili haricinde başka bir programlama diline göre ayarlanması gerektiğinde, Tools

(Araçlar) menüsünden İçeri ve Dışarı Aktarma Ayarları (*Import and Export Settings*) seçeneği tıklanarak gelen diyalog penceresinden Bütün ayarları sıfırla (*Reset all settings*) seçeneği seçilir ve istenilen programlama dili tekrardan aktif edilebilir.



Şekil 1.11. Visual Studio .Net Framework 2019 İlk Programım Ekran Görünümü

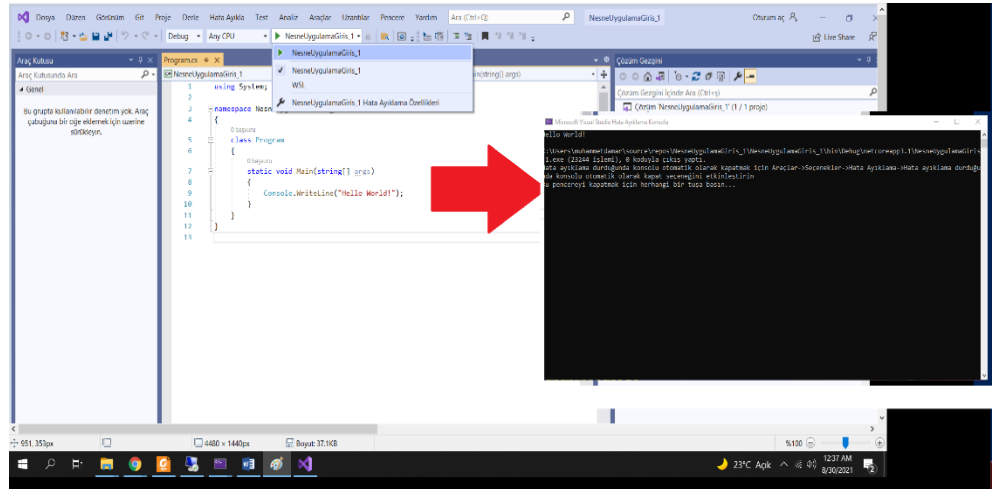
Ayrıca Open Project seçildiğinde *Belgelerim* klasöründe bulunan Visual Studio 2019 klasörü altındaki *Projeler (Projects)* klasöründe önceden hazırlanmış projelere erişmek mümkün olacaktır. Elbette sadece bu klasör içinden değil farklı ortamlardan da projemizi çağırabilir veya açabiliriz. Geliştirilen her projenin kaydedilmesi sonrası; Studio.NET, Projeler (Projects) klasörü altında projenin tüm dosyalarını içine alacak ayrı bir klasör olarak oluşturulur. Visual Studio.Net 2019 editöründe ilk açıldığında görüntülenmemiş olan pencereler view-görünüm menüsü kullanılarak programcıya gösterilir.



Şekil 1.12. Visual Studio .Net Framework 2019 Var Olan Bir Projeyi Açma

.Net Framework 2019 ortamında geliştirdiğimiz kodları aynı zamanda çalıştırmamızı sağlamaktadır. Aşağıdaki şekil üzerinde nesne uygulama giriş

konusundaki konsol için yazılan ilk kod bloğumuzun çalıştırıldığında ekran görüntüsü aşağıda şekil üzerinde verilmektedir.



Şekil 1.13. Visual Studio .Net Framework 2019 İlk Programım Ekran Görünümü



## Özet

- Hızla dönüşen ve gelişen bilişim teknolojileri beraberinde donanımlar üzerinde koşan yazılım teknolojilerinin ve programlama tekniklerinin de gelişip dönüşmesine sebep olmuştur. Bu tekniklerden en önemlisi nesne yönelimli programlama tekniğidir. 1960'lı yıllarda başlayan Nesneye Yönelimli Programlama paradigması, 1990'larda hız kazanarak birçok alanda yaygın hale gelmeye başlamıştır.
- Bu ders için hazırlanan bölümler nesne yönelimli programlama ile programcılığa başalayn programcı adaylarının giriş seviyesinde kodlama yapabilmesini ve Nesne Yönelimli Programlama konusunda bilgilenmesini hedeflemektedir. Bölümler içindeki Nesne Yönelimli Programlama kültürü için ideal bir dil olan C# programlama dili gerçekleştirilmiştir.
- Microsoft, ilk zamanlar bu standartlar doğrultusunda C# programlama dilini geliştirse de C# 3.0 sürümü ile birlikte kendi standartlarını ortaya koymuş ve dili tamamen .NET framework için hazır hale gelmiştir. Bu sayede herhangi bir kuruma bağlı kalımsız yazılım geliştiricilerinin ihtiyaçlarının karşılanması için güçlü bir programlama dili geliştirilmiştir.
- Visual Studio .NET birkaç programlama dilinin aynı anda kullanılabilir yapıya sahiptir. Bu özellik, bütünleşik yazılım geliştirme özelliği olarak tanımlanır ve bu sayede C#, Basic, NET, C++, F# gibi birçok programlama dillerinin hem görsel tasarımı için ortam hem de konsol uygulamaları için destek sağlar. C# Programlama dilinde sınıf tanımlamak class kelimesi kullanılır. Visual Studio.Net 2019 editöründe ilk açıldığında görüntülenmemiş olan pencereler view-görünüm menüsü kullanılarak programcıya gösterilir.
- Visual Studio 2019 .Net versiyonu ile Windows işletim sistemi üzerinde çalışan pek çok uygulamayı geliştirmek mümkündür. Visual Studio ve .NET'i kullanarak masaüstü, web, mobil, oyun ve nesnelerin interneti için uygulamalar geliştirebilirsiniz. .NET uygulamalarını C#, F# veya Visual Basic dilinde yazabilirsiniz. Örneğin; Windows Azure, Office 365 ile Cloud (bulut) teknolojisi ile çalışacak ofis uygulamaları geliştirilebilir.
- Nesneye Yönelimli Programlamaya Alternatif Yöntemler şu şekilde sıralanabilir; fonksiyonel programlama (functional programming), yapılandırılmış veya modüler programlama (structured or modular programming), zorunlu programlama (imperative programming), bildirimsel programlama (declarative programming), mantıksal programlama (logical programming).
- Yazılım geliştirme süreçlerinde nesneleri kullanmak ve tasarlanan sistemi veri modeline ve fonksiyonlarına göre nesneler halinde tasarlamak bize pek çok fayda sağlayacaktır. Modülerlik, tekrar kullanılabilirlik, üretkenlik, güvenlik, esneklik, bakım gibi pek çok fayda ve kolaylığa Nesne yönelimli programlama sahiptir. Detaylı olarak şu şekilde açıklanabilir: modülerlik (modularity), tekrar kullanılabilirlik (reusability), üretkenlik (productivity): kolayca yükseltilebilir ve ölçeklenebilir (easily upgradable and scalable), genişletilebilirlik (extensibility), arayüz tanımlama (interface descriptions), güvenlik (security), esneklik (flexibility), yazılımın bakımı (maintenance) şeklinde ifade edilebilir.

## DEĞERLENDİRME SORULARI

1. Visual Studio.Net 2019 editöründe ilk açıldığında görüntülenmemiş olan pencereler hangi menü kullanılarak görüntülenirler?
  - a) Dosya (File)
  - b) Görünüm (View)
  - c) Düzen (Edit)
  - d) Pencere (Window)
  - e) Çalıştır (Build)
2. Ders kapsamında kullandığımız programlama dili ve kullandığımız platformun adı ve sürümü aşağıdakilerden hangisidir?
  - a) C# - Visual Studio 2019
  - b) Php - Visual Studio 2013
  - c) Javascript - Visual Studio 2013
  - d) Java - Visual Studio 2019
  - e) Java - - Visual Studio 2013
3. C# Programlama dilinde sınıf tanımlamak için aşağıdaki anahtar kelimelerden hangisi kullanılır?
  - a) function
  - b) class
  - c) namespace
  - d) main
  - f) interface
4. Aşağıda verilenden hangisi Nesneye Yönelik Programlamanın sağladığı faydalardan birisi değildir?
  - a) Modülerlik
  - b) Tekrar Kullanılabilirlik
  - c) Bilgi Saklama
  - d) Kolay Hata Ayıklama
  - e) Zor Kod Geliştirme
5. Nesne tabanlı programlamanın yapısı veya yapı taşları arasında aşağıdakilerden hangisi yoktur?
  - a) Sınıflar (Classes)
  - b) Nesneler (Objects)
  - c) Yöntemler (Methods)
  - d) Kalem (Pen)
  - e) Nitelikler (Attributes)

6. Nesne Tabanlı Programlamanın sağladığı avantajlar arasında aşağıdakilerden hangisi yoktur?
  - a) güvenlik (security)
  - b) yazılımın bakımı (maintenance)
  - c) kolayca yükseltilebilir ve ölçeklenebilir (easily upgradable and scalable)
  - d) esneklik (flexibility)
  - e) pahalıdır (expensive)
7. Aşağıdakilerden hangisi nesne tabanlı bir programlama dili değildir?
  - a) C#
  - b) Java
  - c) F#
  - d) C++
  - e) Assembly
8. C# programlama dili için verilen erişim denetleyicilerden hangisinde nitelediği öğeler herkese açık olur?
  - a) private
  - b) protected
  - c) public
  - d) protected global
  - e) private ptotected
9. C# programlama dili için verilen erişim denetleyicilerden hangisinde nitelediği öğeler sadece bulunduğu sınıf içinde herkese açık olur ve başka sınıftaki kodların erişmesini engellemek için kullanılır?
  - a) pivate
  - b) protected
  - c) public
  - d) protected global
  - e) private ptotected
10. Nesneye Yönelimli Programlamaya Alternatif Yöntemler arasında aşağıdakilerden hangisi sayılmaz?
  - a) yapılandırılmış veya modüler programlama (structured or modular programming)
  - b) eğlenceli programlama (funny programming)
  - c) bildirimsel programlama (declarative programming),
  - d) fonksiyonel programlama (functional programming),
  - e) mantıksal programlama (logical programming)

**Cevap Anahtarı**

1.b, 2.a, 3.b, 4.e, 5.d, 6.e, 7.e, 8.c, 9.a, 10.b

## YARARLANILAN KAYNAKLAR

- C# Klavuzu (2021). C# Belgeleri, Visual Studio. Erişim Tarihi: 27/08/2021, <https://docs.microsoft.com/tr-tr/>
- Clark, D. (2013). *Beginning object-oriented programming with VB 2005: From novice to professional*. New York: Apress.
- Dathan, B., & Ramnath, S. (2015). *Object-Oriented Analysis, Design and Implementation*. ABD: Springer.
- Gillis, A.S. & Lewis, S. (2021). Object-Oriented Programming (OOP). Erişim Tarihi: 27/08/2021, <https://searchapparchitecture.techtarget.com/>
- Gunnerson, E., & Wienholt, N. (2012). *A Programmer's Guide to C# 5.0*. New York: Apress.
- İTÜBİDB (2019). Nesne Yönelimli Programlama (Object-Oriented Programming). Seyir Defteri. İTÜ Bilgi İşlem Daire Başkanlığı. Erişim Tarihi: 27/08/2021, <https://bidb.itu.edu.tr/sevir-defteri/blog/2019/02/05/object-oriented-programming>
- İÜAÖF (2021). İstanbul Üniversitesi Açık ve Uzaktan Eğitim Fakültesi, Nesneye Yönelik Programlama. Erişim Tarihi: 27/08/2021, [https://cdn-acikogretim.istanbul.edu.tr/auzefcontent/20\\_21\\_Guz/nesneye\\_yonelik\\_programlama/](https://cdn-acikogretim.istanbul.edu.tr/auzefcontent/20_21_Guz/nesneye_yonelik_programlama/)
- Karaçay T. (2021). Erişim Belirteçleri (Access Modifiers), Başkent Üniversitesi, Bölüm 11. Erişim Tarihi: 27/08/2021, [https://mail.baskent.edu.tr/~tkaracay/etudio/ders/prg/java/java\\_ndx.html](https://mail.baskent.edu.tr/~tkaracay/etudio/ders/prg/java/java_ndx.html)
- Küçükkoç, İ. (2020). Algoritma ve Programlamaya Giriş, Ders Notları. Balıkesir Üniversitesi, Mühendislik Fakültesi Endüstri Mühendisliği Bölümü.
- Nakov, S., & Kolev, V. (2013). *Fundamentals of Computer Programming with C#: The Bulgarian C# Book*. Sofia: Faber Publishing.
- Turkcell (2021). Geleceği Yazarlar. Nesne ve Nesne Yönelimli Programlama Teorisinin 4 Temel Özelliği. Erişim Tarihi: 27/08/2021, <https://gelecegiyazarlar.turkcell.com.tr/konu/egitim/android-101/nesne-ve-nesne-yonelimli-programlama-teorisinin-4-temel-ozelligi>
- TutorialsPoint (2014). C# Programming Object-Oriented Programming. Erişim Tarihi: 27/08/2021, <http://www.tutorialspoint.com>.