



SAKARYA
ÜNİVERSİTESİ

BSM 101

BİLGİSAYAR MÜHENDİSLİĞİNE GİRİŞ

HÜSEYİN ESKİ, İSMAİL ÖZTEL

~ Programlama Dilleri~

İÇERİK



- Yazılım geliştirme süreci
- Programlama dillerinin tarihçesi
- Programlama dillerinin sınıflandırılması
- Nesne yönelimli programlama
- Programlama dillerinin elemanları
 - Kod söz dizimi, veri, atama deyimi, tür kontrolü, kontrol deyimleri, alt programlar, modüller

Program

- Bilgisayar mühendisliğinin bir kolu olan programlama dilleri teorisi tasarım, uygulama, analiz, dillerin sınıflandırılması ve dillerin özelliklerini içinde barındırır.
- Programlama dilleri teorisi , yazılım mühendisliği, dil bilimi, matematik gibi farklı disiplinleri içerir.
- Program: bir problemi çözmek için kullanılan ardışık simgeler dizisi olarak adlandırılabilir.
- Programlama dili: bir makinenin davranışlarını kontrol etmek için, yukarıda sözü edilen ardışık simgelerin oluşturduğu yapay bir dildir.

Yazılım Geliştirme Süreci

- Bir yazılım 5 evre geçirerek ortaya çıkar, bu evreler genellikle birbirini takip eder:
 - Gereksinim analizi
 - Yazılım tasarımı
 - Kodlama
 - Sertifikasyon
 - Bakım

Yazılım Geliştirme Süreci

- Gereksinim analizi
 - Bir yazılım, belirli bir kullanıcı kitlesinin belirli bir ihtiyacını yerine getirmek için geliştirilir.
 - Bu yazılım, beklenen ihtiyaçları tam olarak karşılayabilmelidir.
 - Bu ihtiyaçlar kullanıcılar tarafından net bir şekilde yazılı olarak ortaya konmalıdır. Bu aşamada genellikle kullanıcılar ve yazılım geliştiriciler bir arada olurlar.
 - Sistemin başarısı, sistemin nasıl çalışacağını belirleyen kullanıcı kitlesinin ihtiyaçlarının eksiksiz ve net bir şekilde ortaya koyulması ile doğrudan ilişkilidir.

Yazılım Geliştirme Süreci

- Yazılım tasarımı
 - Bir önceki adıma ortaya konan gereksinim belgeleri kullanılarak tasarım adımı gerçekleşir.
 - Bu aşamanın çıktısında tasarım ile ilgili tanımlamaların yapıldığı bir belge ortaya çıkar.
 - Bu belgelerde sistemi oluşturan ayrıntılar (modüller ve arayüzler) tanımlanır.
 - Burada tercih edilen tasarım yönteminin, kullanılacak programlama dilinin seçimine etkisi büyük olacaktır.

Yazılım Geliştirme Süreci

- Kodlama
 - Tasarım evresinde ortaya çıkan belgelendirmelerde yer alan tanımlamaların, bir programlama dili aracılığı ile gerçekleştirilmesi bu aşamada olur.
 - Bu aşama sonucunda çalışma, tümüyle gerçekleşmiş ve raporlanmış bir yazılımdır.
- Sertifikasyon
 - Bu adıma yazılımın kalite denetimi yapılarak son kullanıcıya sunulur.
 - Sertifikasyon işlemi modüller gerçekleştikçe olabileceği gibi tüm yazılım gerçekleştirimi tamamlanınca da yapılabilir.
 - Bu aşamadaki amaç; gereksinimler belgesindeki isterlerin tam olarak karşılanması ve arayüzlerin problemsiz olarak çalışmasıdır.

Yazılım Geliştirme Süreci

- Bakım
 - Bu aşamada yazılımda meydana gelen hataların giderilmesi ve yazılıma, ihtiyaca yönelik yeni bileşenlerin eklenmesi yer alır.
 - Genel olarak yazılım geliştirme süreçleri incelendiğinde; bir yazılım sisteminin bakım maliyeti, sistemin diğer evreleri için harcanan toplam maliyete eşit, hatta daha fazla olabilmektedir.

Yazılım Geliştirme Sürecinde Programlama Dilinin Önemi

- Yazılım geliştirme sürecinde başarıya ulaşmak için dikkat edilmesi gereken etkenlerden biri de seçilecek programlama dilidir.
- Seçilecek programlama dili ve bu dille yazılmış yazılımın bazı temel kavramları sağlaması gerekir.
 - Yazılım güvenilir olmalıdır.
 - Yazılım bakıma elverişli olmalıdır.
 - Yazılım verimli çalışmalıdır.

Yazılım Geliştirme Sürecinde Programlama Dilinin Önemi

- Yazılım güvenilir olmalıdır.
 - Kullanıcılar, üzerinde işlemlerini yürüttükleri yazılımlara güvenmelidirler.
 - Kullanıcılar yazılım kullanma esnasında ortaya çıkabilecek yazılım ve donanımsal hata mesajlarında, yazılıma olan güvenini kaybedebilirler.
 - Bir yazılımın güvenilir olabilmesi için kullanılan dil aşağıdaki kriterleri sahip olmalıdır:
 - Yazılabilirlik
 - Okunabilirlik
 - Sıra dışı durumları göğüsleyebilme

Yazılım Geliştirme Sürecinde Programlama Dilinin Önemi

- Yazılım güvenilir olmalıdır.
 - Yazılabilirlik:
 - Probleme dair veri giriş çıkışı, akış denetimi gibi unsurların, programlama dilinin içerisinde doğal olarak karşılanabilmesi ile ilgilidir.
 - Programcı, problem ile ilgili detayların dışındaki detaylar ile ilgilenmemelidir.
 - Ölçülmesi zor bir özellik olsa da yüksek seviyeli dillerin yazılabilirlik özelliği, düşük seviyedeki dillere göre daha fazla olduğu söylenebilir.

Yazılım Geliştirme Sürecinde Programlama Dilinin Önemi

- Yazılım güvenilir olmalıdır.
 - Okunabilirlik:
 - Yazılım geliştirirken, program mantığını takip etmek, sınamak ve hata ayıklamak mümkün olabilmelidir.
 - Dil ne kadar basit ise, algoritmalar kullanılan dil ile ne derece doğal bir şekilde ifade edilebiliyorsa; kodun sonradan incelenmesi, işlevinin anlaşılması da o derece kolay olacaktır. Ör: goto
 - Sıra dışı durumları göğüsleyebilme:
 - Kullanılan dil, hatalı girişi gibi durumlara hazırlıklı olmalı ve yeri geldiğinde bu ortaya çıkabilecek durumlar için uygun çözümler içermelidir.

Yazılım Geliştirme Sürecinde Programlama Dilinin Önemi

- Yazılım bakıma elverişli olmalıdır.
 - Yazılım bakımının temel unsurları:
 - Hataların ayıklanması ve giderilmesi
 - Yeni modüllerin eklenmesi ve çıkarılması
 - Yazılımın kullanıcı ihtiyaçlarına göre geliştirilmesi
 - Bir yazılımın bakımının yapılabilmesi için kullanılan dilin okunabilirliği ile değiştirilebilirlik yeteneğinin yüksek olması gerekir.

Yazılım Geliştirme Sürecinde Programlama Dilinin Önemi

- Yazılım verimli çalışmalıdır
 - Yazılımın verimliliği programlama dilinin olanaklarına doğrudan bağlıdır.
 - Programlama dili tasarlanırken dikkat edilen en önemli hususlardan biri verimliliktir.
 - İlk zamanlarda verimlilik, programlama dilinin üzerinde koştığı donanımı etkin ve hızlı bir şekilde kullanabilmesi ile ölçülüyordu.
 - Ör: FORTRAN, IBM 704 için tasarlanmış ve dilin kısıtları bu makineye göre ayarlanmıştı. Böylece FORTRAN, IBM 704 üzerinde yüksek verimlilikle çalışan bir dil olarak ortaya çıktı.
 - Verimlilik için donanımın etkin ve hızlı çalışabilir olmasının yanında sistemin oluşturulabilmesi ve bakımının yapılabilmesinde harcanacak zaman ve emek de önemli bileşenler olmuştur.

Programlama Dillerinin Tarihçesi

- 1822 yılında Charles Babbage'nin fark makinesi icadı ile, makinelerin bir işi yapma emirlerini alma ihtiyacı doğmuştur.
- Programlama dili bu ihtiyaca yönelik doğmuştur.
- İlk programlarda, talimatlar adım adım bilgisayara girilmekteydi.
- Sonraki yıllarda, programlarda mantıksal dallanmalar, nesne yönelimli yaklaşımlar ortaya çıkmıştır.



Programlama Dillerinin Tarihçesi

- Fark makinesi dişli çarkların ayarlanması yoluyla çalışan mekanik bir yapıdır.
- Bu ayarlamalar ilk programlama dili olarak da kabul edilebilir.
- Sonraki dönemlerde mekanik hareketin yerini elektriksel işaretler almıştır.
- 1942 yılında ENIAC üzerinde elektriksel işaretlerden oluşan ilk programlama dili kullanılmıştır.



Programlama Dillerinin Tarihçesi

- 1949'da Short Code geliştirilmiştir.
 - Short Code elektronik cihazları kodlayabilen ilk programlama dilidir.
 - Karmaşık programlama dillerinin ilk örneğidir.
 - 0 ve 1'ler kullanılarak programlama yapılmaktadır.
- 1951 yılında ilk derleyici yazılmıştır.
 - Daha hızlı programlama yapabilmenin önü açılmıştır.
- 1957 yılında ilk gelişmiş programlama dili FORTRAN geliştirilmiştir.
 - Bilimsel hesaplamalar için geliştirildi.
 - Günümüzde tercih edilmemekle birlikte, kullandığımız basit veri türlerinin kaynağı FORTRAN dilidir.

Programlama Dillerinin Tarihçesi

- 1959'da COBOL dili geliştirilmiştir.
 - FORTRAN giriş çıkış işlemlerinin kontrolünde yeterli değildi, giriş çıkış işlemlerinin ticari işlemlerde önemli olmasından dolayı COBOL geliştirildi.
- 1958'de John McCarthy yapay zeka araştırmaları için LISP dilini geliştirmiştir.
 - İçerisinde diğer dillerden farklı olarak sadece liste veri türünü barındırır.
 - Dilin kendi kendini değiştirme ve kendi kendini geliştirme gibi yetenekleri vardır.
- Yine 1958 yılında ALGOL dili geliştirildi.
 - PASCAL, C, C++, JAVA gibi dillerin gelişmesinde ilk aşamadır.
 - Bir sonraki versiyon ALGOL 68 ile birlikte rekürsif fonksiyon özelliği geldi.

Programlama Dillerinin Tarihçesi

- 1964'te BASIC dili geliştirildi.
 - Komutlar ardışık olarak icra edilir.
 - Değişken tanımlama ve giriş çıkış işlemleri oldukça basit ancak sınırlı bir dildir.
 - Görselliği ön plana çıkaran Visual Basic geliştirilmiştir.
 - VB ile Windows uygulamaları kolay bir şekilde hazırlanabildiğinden çok popüler oldu.
- 1968 yılında PASCAL dili; COBOL, FORTRAN ve ALGOL gibi dillerin iyi özellikleri alınarak geliştirildi.
 - Bazı dillerin iyi özellikleri alınarak oluşturulduğundan oldukça başarılı bir dil olmuştur.
 - İşaretçi veri türü de geliştirilmiştir.
 - Case kontrol yapısının içeren ilk dildir.
 - Dinamik değişkenler program çalışırken oluşturulabilirler ve silinebilirler.

Programlama Dillerinin Tarihçesi

- 1972 yılında C dili geliştirilmiştir.
 - İlk gelişmiş dillerden günümüz modern dillerine geçiş PASCAL'dan C diline geçiş ile olmuştur.
 - C dili ile PASCAL dili arasındaki benzerlikler oldukça fazladır.
 - İşaretçi yapısını kapsamlı bir şekilde kullandığı için hızlı ve güçlü bir dildir ama okunabilirliği zordur.
 - Geliştirilmeye devam edilen UNIX sisteminde kullanılmak üzere geliştirilmiştir.
 - UNIX, Windows, MacOS, Linux sürümlerinde kullanılmıştır.
 - Seksenli yıllarda nesne yönelimli programlama yaklaşımının çıkmasıyla sınıflı C olarak isimlendirilen (C with classes) geliştirilmiştir.

Programlama Dillerinin Tarihçesi

- 1983 yılında C++ dili kullanıma sunulmuştur.
 - C dilinin gücünü nesne yönelimli programlamanın gücü ile birleştirmiştir.
- Yazılan bir programın bütün makinelerde çalışabilmesi, yani aslında taşınabilirlik probleminin giderilebilmesi için 1994'te Java geliştirildi.
 - Java, kullanımı günden güne artan web üzerine de yoğunlaşmıştır.
 - Çok kullanılan programlama dillerinden biridir.

Programlama Dillerinin Tarihçesi

- Programlama dilleri yıllardır gelişmekte olup, gelişmeye devam edecekleri de öngörülmektedir.
- Her yeni geliştirilen dil, dillerde var olan eksikleri ve ihtiyaçlara yönelik ortaya çıkan yeni eksiklikleri gidermektedir.
- Bu şekilde programlama dilleri Bilgisayar Mühendisliği'nin dinamik konularından biri olmaya devam edecektir.

Programlama Dillerinin Sınıflandırılması

- Genel sınıflandırma
 - Temel programlama dilleri
 - Fortran, C, Cobol
 - Veriye yönelik programlama dilleri
 - Lisp, Snobol
 - Nesneye yönelik programlama dilleri
 - C++, Java, C#

Programlama Dillerinin Sınıflandırılması

- Uygulama alanlarına göre sınıflandırma
 - Bilimsel ve mühendislik diller
 - Fortran, C, Python, R
 - Sistem programlama dilleri
 - C, Assembly
 - Veri tabanı dilleri
 - Dbase, Clipper
 - Yapay zeka dilleri
 - Prolog, Lisp
 - Genel amaçlı programlama dilleri
 - C, Basic

Programlama Dillerinin Sınıflandırılması

- Seviyelerine göre sınıflandırma
 - Düşük seviye
 - Assembly
 - Orta seviye
 - C, C++, C#
 - Yüksek seviye
 - Pascal, Python
 - Çok yüksek
 - Dbase, VB

Nesne Yönelimli Programlama

- Gerçek dünyada var olan ya da programcı tarafından kurgulanan mantıksal bir varlık olarak programlarda nesne kavramı kullanılmaktadır.
- Bir nesne; kendisini oluşturan veriler ve veriler üzerinde yürütülecek işlemler ile birlikte bir bütündür.
- Benzer nesnelere genel bir isim verilip sınıf isminde bir tür oluşturulur.

Nesne Yönelimli Programlama

- Nesne yönelimli programlama üç temel yapı üzerine oturmaktadır.
 - Veri soyutlama: yeni veri türlerini modelleyen sınıflar oluşturulması
 - Kalıtım: bir sınıfın özelliklerini alıp yeni özelliklerle yeni bir sınıf elde edilmesi
 - Çok biçimlilik: aynı isimli işlemlerin farklı nesneler için farklı algılanması olayı

Nesne Yönelimli Programlama

- C
 - Yapısal programlama dilleri arasındadır.
 - Oldukça kullanışlı ve esnek bir yapıya sahiptir.
 - C ile birlikte
 - Sistem yazımı
 - Oyun programlama
 - Yazıcı kontrolü gibi geniş bir yelpazede uygulamalar yapılabilir. Kullanım alanı çok geniştir.

Nesne Yönelimli Programlama

- C++
 - Nesne yönelimli programlama yapılabilen diller arasındadır.
 - C’de belirtilen tüm güçlü özellikleri barındırır.
 - Dünyada en çok kullanılan dillerden biridir.
- Java
 - Nesneye dayalı bir programlama dilidir.
 - Görsel özellikleri, kütüphane desteği ile kullanımı giderek artmıştır.
 - Hemen hemen her alanda kullanılan güçlü ve esnek bir dildir.
 - İnternet tabanlı yazılımlar tasarlanması için de ciddi desteği vardır.

Nesne Yönelimli Programlama

- C#
 - Nesne yönelimli bir programlama dilidir
 - C++ ve Java'nın iyi yönlerini alarak oluşturulmuştur.
 - Ayrıca Java teknolojisine rakip bir Microsoft dildir.
 - Yerel ve internet uygulamaları kolayca yapılabilmektedir.
- VB.Net
 - Nesneye dayalı bir dildir.
 - VB'deki bir çok özelliği barındırmaktadır.
 - OOP desteklenerek daha verimli kod yazımı sağlanmıştır.

Nesne Yönelimli Programlama

- PASCAL
 - Yapısal bir dildir ve kod yazımı olarak C'ye benzemektedir.
 - Genelde bilimsel araştırmalar için üniversitelerde kullanıldı.
- Delphi
 - PASCAL tabanlı bir dildir.
 - Nesne yönelimli programlama yapılabilir.
 - Görsel programlama yapılabilir.

Nesne Yönelimli Programlama

- Python
 - Nesne yönelimli bir dildir.
 - Yorumlayıcı kullanır.
 - Sistem programlama ve bilimsel hesaplamalarda kullanılan bir dildir.

Programlama Ortamı

- Bir programlama ortamında dil ile birlikte bir çok unsur bulunur.
- Bu unsurların amacı, programlama dili ile birlikte oluşturulan kodların, üzerinde koştugu donanım tarafından anlaşılmasını sağlamaktır.
- Bu unsurlar:
 - Editör: kod yazımı ve yazılan kodların değişimi için gereken araçtır.
 - Derleyici: editörde oluşturulan kaynak kodun makine koduna dönüşmesini sağlar.
 - Yorumlayıcı: bir programın kaynak kodunu satır satır yürüten programdır.
 - Kütüphane: nesne dosyalarından oluşur.

Programlama Ortamı

- Bağlayıcı: programdaki tüm nesne dosyalarını birleştirir ve çalışabilir tek bir program haline getirir.
- Yükleyici: yürütülebilir dosyayı belleğe getirir.
- Hata ayıklayıcı: programdaki hataların anlaşılabilmesi için programın adım adım çalıştırılmasını sağlar.

Programlama Dillerinin Elemanları

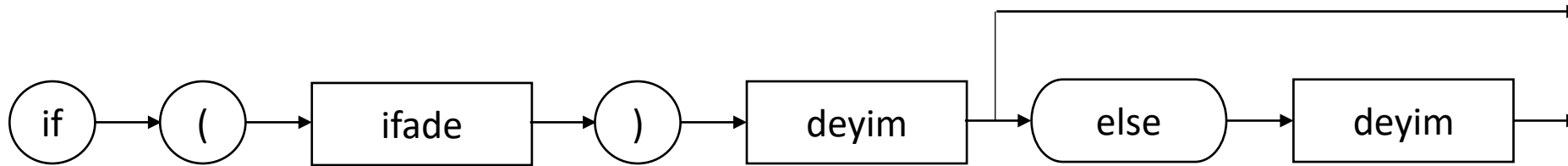
- Sözdizimi:
 - Doğal dillerde de olduğu gibi programlama dillerinde de bir sözdizimi (syntax) vardır.
 - Sözdizimi, simgelerin yazıldığı düzenin geçerli olup olmadığını inceleyen kurallar kümesidir.
 - Sözdizimi olarak en çok bilinen notasyon EBNF notasyonudur.
 - Bir kodun sözdizimi doğru olsa bile, kod anlamsal olarak doğru olmayabilir.
 - C dilindeki if deyiminin sözdizimi:
 - *if-deyimi ::= if (ifade) deyim [else deyim]*

Programlama Dillerinin Elemanları

- C dilindeki if deyiminin sözdizimi:
 - *if-deyimi* ::= **if** (*ifade*) *deyim* [**else** *deyim*]
 - Burada italik yazılı olanlar sözdizimsel ifadelerdir.
 - Koyu yazılanlar programda görülmesi gereken isim ya da simgelerdir.
 - Tüm kurallar ::= simgesini içerir ve anlamı “-den oluşmuştur” anlamı taşır.
 - [] içerdiği ifadenin seçimlik olduğunu gösterir.
 - {} içerdiği ifadenin sıfır ya da daha fazla tekrar edebileceğini belirtir.
 - | veya anlamında kullanılır.
 - *değişken-tanımı* ::= *tür belirleyici* { *belirleyici* }

Programlama Dillerinin Elemanları

- *if-deyimi* ::= **if** (*ifade*) *deyim* [**else** *deyim*] kuralı, sözdizimi diyagramları şeklinde ifade edilebilir.



- Daire ve oval şekiller gerçek karakterleri, dikdörtgenler ise sözdizimsel ifadeleri göstermektedir.

Programlama Dillerinin Elemanları

- Sözdizimsel hataların çoğu derleyici tarafından yakalanır.
- Bazı hatalar ise derleyici tarafından yakalanamayabilir ve çalışma zamanında hataya neden olur. Bu tür hatalardan bazıları:
 - Belirleyicilerin uzunlukları ile ilgili dile ait sınırlamalar
 - Ör: Eğer sınırlamalarda belirleyicilerin ilk on karakteri alınıyorsa aşağıdaki iki değişken program için aynı değişken olarak değerlendirilecektir.
 - Ogrencinin_numarasi - Ogrencinin_notu
 - Küçük büyük harf ayrımı
 - Ör: Aşağıdaki iki değişken küçük büyük harf ayrımının olduğu dillerde farklı algılanırken, bu ayrımın yapılmadığı dillerde farklı değerlendirilir ve hataya neden olabilir.
 - Sayi - SAYI

Programlama Dillerinin Elemanları

- Veri
 - Bir programlama dilini öğrenirken deyimler ve komutlar ön plandadır ve bununla birlikte verinin yapısı ile ilgili konular da bu süreçte öğrenilir.
 - Tür: bir değerler ve bu değerler üzerinde yapılabilecek işlemler kümesidir.
 - Örneğin C dili için int veri tipi içinde belirli bir büyüklüğe kadar değerler saklanabilir ve bu değerler üzerinde bazı aritmetik işlemler yapılabilir.
 - C++ gibi modern dillerde, dilin izin verdiği veri tiplerinin dışında çözülmeye çalışılan probleme uygun veri tipleri de tanımlanabilir.

Programlama Dillerinin Elemanları

- Tür kontrolü
 - Tür kontrolü atanacak veri ile atanacağı bellek hücresi için belirlenmiş olan veri tipinin uyumlu olup olmadığının kontrol edilmesidir.
 - Atama işlemlerinde atama deyiminin sol tarafında bir bellek hücresi adresi hesaplanır. Atama deyiminin sağ tarafında ise belirli bir türde bir değer vardır.
 - Adresle ilişkilendirilmiş veri türü ile atanacak olan verinin türü aynı olmayabilir.
 - Aynı zamanda atanacak olan verinin boyutu, atanacak olan adresle ilişkilendirilmiş veri türünün boyutunu da aşabilir.

Programlama Dillerinin Elemanları

- Tür kontrolü
 - Tür kontrolü ile ilgili olası yaklaşımlar
 - Hiçbir şey yapılmaz, sorumluluk bu durumda programcıya aittir.
 - Atama deyiminin sol tarafının veri türünü sağ taraftaki veri türüyle aynı yap.
 - Güçlü tür kontrolü: Uyumsuzluk durumunda atama yapma
 - Burada esneklik ve güvenilirlik arasında bir ters orantı bulunur.
 - Tür kontrolü arttıkça güvenilirlik artar, bununla birlikte uygun bir tür kümesi oluşturmak için daha fazla çaba sarf etmek gerekir.
 - Tür kontrolü azaldıkça esneklik artar ancak hata bulmak zorlaşır ve güvenilirlik azalır.

Programlama Dillerinin Elemanları

- Kontrol deyimleri
 - Kontrol deyimleri programın yürütme sırasını değiştirmek için kullanılır.
 - İyi yapılandırılmış bir kontrol deyimi
 - Birden fazla seçenekten birinin seçilmesini sağlayabilir: *if, switch*
 - Tekrarlı ifadelerin çevirim durumları için kullanılabilir: *for, while*
 - Yürütme zamanının büyük bölümü döngüler içinde geçer.
 - Genellikle döngülerdeki hatalar döngünün başında yada sonunda gözlemlenebilir.

Programlama Dillerinin Elemanları

- Alt programlar
 - Alt programlar, içinde veri tanımlamalarının ve deyimlerin bulunduğu program parçalarıdır.
 - Programların farklı yerlerinden defalarca kez çağrılabilirler.
 - Bir alt program çağrıldığında, çağırılan yerden verileri parametre olarak alır.
- Modüller
 - Programlarda satır sayısı arttıkça programları kontrol etmek ve anlamak zorlaşır.
 - Büyük çaplı programlar bir proje ekibi tarafından geliştirilirler.
 - Yazılımlar, yönetilebilecek küçük parçalar halinde modüllere ayrılır.